LS-DYNA®

KEYWORD USER'S MANUAL

VOLUME I

LS-DYNA R10.0
10/16/17 (r:9023)

LIVERMORE SOFTWARE TECHNOLOGY CORPORATION (LSTC)

Corporate Address

Livermore Software Technology Corporation
P. O. Box 712
Livermore, California 94551-0712

Support Addresses

Livermore Software Technology Corporation
7374 Las Positas Road
Livermore, California 94551
Tel: 925-449-2500 Fax: 925-449-2507
Email: sales@lstc.com
Website: www.lstc.com

Livermore Software Technology Corporation
1740 West Big Beaver Road
Suite 100
Troy, Michigan 48084
Tel: 248-649-4728 Fax: 248-649-6328

Disclaimer

LS-DYNA®, LS-OPT® and LS-PrePost® are registered trademarks of Livermore Software Technology Corporation in the United States. All other trademarks, product names and brand names belong to their respective owners.

LSTC reserves the right to modify the material contained within this manual without prior notice.

The information and examples included herein are for illustrative purposes only and are not intended to be exhaustive or all-inclusive. LSTC assumes no liability or responsibility whatsoever for any direct or indirect damages or inaccuracies of any type or nature that could be deemed to have resulted from the use of this manual.

Any reproduction, in whole or in part, of this manual is prohibited without the prior written approval of LSTC. All requests to reproduce the contents hereof should be sent to sales@lstc.com.

Patents

LSTC products are protected under the following patents:

US Patents: 7167816, 7286972, 7308387, 7382367, 7386425, 7386428, 7392163, 7395128, 7415400, 7428713, 7472602, 7499050, 7516053, 7533577, 7590514, 7613585, 7640146, 7657394, 7660480, 7664623, 7702490, 7702494, 7945432, 7953578, 7987143, 7996344, 8050897, 8069017, 8126684, 8150668, 8165856, 8180605, 8190408, 8200458, 8200464, 8209157, 8271237, 8296109, 8306793, 8374833, 8423327, 8467997, 8489372, 8494819, 8515714, 8521484, 8577656, 8612186, 8666719, 8744825, 8768660, 8798973, 8898042, 9020784, 9098657, 9117042, 9135377, 9286422, 9292632, 9405868, 9430594, 9507892, 9607115.

Japan Patents: 5090426, 5281057, 5330300, 5373689, 5404516, 5411013, 5411057, 5431133, 5520553, 5530552, 5589198, 5601961, 5775708, 5792995, 5823170, 6043146, 6043198.

China Patents: ZL200910207380.5, ZL200910165817.3, ZL200910221325.1, ZL200910246429.8, ZL201010128222.3, ZL201010132510.6, ZL201010155066.X, ZL201010171603.X, ZL201010174074.9, ZL201010287263.7, ZL201110037461.2, ZL201010533046.1, ZL201110035253.9, ZL201110065789.5, ZL201110132394.2, ZL201110140142.4, ZL201210039535.0, ZL201210286459.3, ZL201210275406.1, ZL201210514668.9, ZL201210422902.5, ZL201210424131.3, ZL201310021716.5, ZL201210475617.X, ZL201310081855.7.

EPO Patents: EP219192881, EP2180416, EP2219124.

and pending patents applications.

AES

DISCLAIMER

This software is provided 'as is' with no explicit or implied warranties in respect of any properties, including, but not limited to, correctness and fitness for purpose.

Issue Date: 21/01/2002

This file contains the code for implementing the key schedule for AES (Rijndael) for block and key sizes of 16, 24, and 32 bytes.

TABLE OF CONTENTS

TABLE OF CONTENTS 0-1

INTRODUCTION 1-1

CHRONOLOGICAL HISTORY....................1-1

1989-1990....................1-2

1991-1992....................1-4

1993-1994....................1-5

1995 1-6

950 1-9

960 1-13

970 1-18

971 1-24

971R6....................1-32

97R6.1 1-38

R7.0....................1-42

R7.1 1-58

R8.0....................1-75

R9.0....................1-98

R10.0....................1-154

MATERIAL MODELS....................1-204

SPATIAL DISCRETIZATION....................1-206

CONTACT-IMPACT INTERFACES 1-209

INTERFACE DEFINITIONS FOR COMPONENT ANALYSIS....................1-211

PRECISION 1-211

GETTING STARTED....................2-1

DESCRIPTION OF KEYWORD INPUT....................2-1

SUMMARY OF COMMONLY USED OPTIONS 2-12

EXECUTION SYNTAX 2-13

SENSE SWITCH CONTROLS....................2-16

Procedure for LS-DYNA/MPP 2-18

FILES....................2-19

RESTART ANALYSIS 2-21

VDA/IGES DATABASES....................2-23

LS-PrePost®....................2-24

TABLE OF CONTENTS

EXECUTION SPEEDS 2-26

UNITS 2-27

GENERAL CARD FORMAT 2-28

*AIRBAG 3-1

*AIRBAG 3-2

Core Cards 3-3

*AIRBAG_SIMPLE_PRESSURE_VOLUME 3-10

*AIRBAG_SIMPLE_AIRBAG_MODEL 3-12

*AIRBAG_ADIABATIC_GAS_MODEL 3-15

*AIRBAG_WANG_NEFSKE 3-17

*AIRBAG_LOAD_CURVE 3-33

*AIRBAG_LINEAR_FLUID 3-34

*AIRBAG_HYBRID 3-36

*AIRBAG_HYBRID_JETTING 3-36

*AIRBAG_HYBRID_CHEMKIN 3-45

*AIRBAG_FLUID_AND_GAS 3-51

*AIRBAG_ALE 3-55

*AIRBAG_INTERACTION 3-69

*AIRBAG_PARTICLE 3-71

*AIRBAG_REFERENCE_GEOMETRY 3-87

*AIRBAG_SHELL_REFERENCE_GEOMETRY 3-90

*ALE 4-1

*ALE_AMBIENT_HYDROSTATIC 4-4

*ALE_COUPLING_NODAL_CONSTRAINT 4-7

*ALE_COUPLING_NODAL_DRAG 4-10

*ALE_COUPLING_NODAL_PENALTY 4-14

*ALE_COUPLING_RIGID_BODY 4-17

*ALE_ESSENTIAL_BOUNDARY 4-19

*ALE_FAIL_SWITCH_MMG 4-21

*ALE_FRAGMENTATION 4-23

*ALE_FSI_PROJECTION 4-25

*ALE_FSI_SWITCH_MMG 4-28

*ALE_FSI_TO_LOAD_NODE 4-33

*ALE_MULTI-MATERIAL_GROUP 4-35

*ALE_REFERENCE_SYSTEM_CURVE 4-39

*ALE_REFERENCE_SYSTEM_GROUP 4-42

*ALE_REFERENCE_SYSTEM_NODE 4-50
*ALE_REFERENCE_SYSTEM_SWITCH....................4-53
*ALE_REFINE 4-55
*ALE_SMOOTHING....................4-56
*ALE_STRUCTURED_MESH....................4-59
*ALE_STRUCTURED_MESH_CONTROL_POINTS 4-62
*ALE_STRUCTURED_MESH_REFINE 4-65
*ALE_SWITCH_MMG....................4-67
*ALE_TANK_TEST....................4-74
*ALE_UP_SWITCH 4-78

*BOUNDARY 5-1
*BOUNDARY_ACOUSTIC_COUPLING 5-3
*BOUNDARY_ACOUSTIC_IMPEDANCE....................5-6
*BOUNDARY_ACOUSTIC_MAPPING 5-7
*BOUNDARY_ALE_MAPPING 5-8
*BOUNDARY_AMBIENT 5-14
*BOUNDARY_AMBIENT_EOS 5-16
*BOUNDARY_CONVECTION....................5-18
*BOUNDARY_COUPLED 5-21
*BOUNDARY_CYCLIC 5-23
*BOUNDARY_DE_NON_REFLECTING....................5-26
*BOUNDARY_FLUX 5-27
*BOUNDARY_MCOL....................5-32
*BOUNDARY_NON_REFLECTING 5-34
*BOUNDARY_NON_REFLECTING_2D 5-36
*BOUNDARY_PAP 5-38
*BOUNDARY_PORE_FLUID 5-40
*BOUNDARY_PRECRACK....................5-42
*BOUNDARY_PRESCRIBED_ACCELEROMETER_RIGID....................5-43
*BOUNDARY_PRESCRIBED_FINAL_GEOMETRY....................5-45
*BOUNDARY_PRESCRIBED_MOTION 5-47
*BOUNDARY_PRESCRIBED_ORIENTATION_RIGID 5-57
*BOUNDARY_PRESSURE_OUTFLOW....................5-65
*BOUNDARY_PWP 5-66
*BOUNDARY_RADIATION 5-70
*BOUNDARY_RADIATION_SEGMENT 5-72

*BOUNDARY_RADIATION_SEGMENT_VF 5-75
*BOUNDARY_RADIATION_SET 5-77
*BOUNDARY_RADIATION_SET_VF 5-80
*BOUNDARY_SLIDING_PLANE 5-83
*BOUNDARY_SPC 5-84
BOUNDARY_SPC_SYMMETRY_PLANE_*{OPTION} 5-87
*BOUNDARY_SPH_FLOW 5-90
*BOUNDARY_SPH_NON_REFLECTING 5-93
*BOUNDARY_SPH_SYMMETRY_PLANE 5-94
*BOUNDARY_SYMMETRY_FAILURE 5-95
*BOUNDARY_TEMPERATURE 5-96
*BOUNDARY_THERMAL_BULKFLOW 5-98
*BOUNDARY_THERMAL_BULKNODE 5-99
*BOUNDARY_THERMAL_WELD 5-101
*BOUNDARY_THERMAL_WELD_TRAJECTORY 5-105
*BOUNDARY_USA_SURFACE 5-111

*BOUNDARY_ELEMENT_METHOD 5-1
*BOUNDARY_ELEMENT_METHOD_CONTROL 6-2
*BOUNDARY_ELEMENT_METHOD_FLOW 6-4
*BOUNDARY_ELEMENT_METHOD_NEIGHBOR 6-6
*BOUNDARY_ELEMENT_METHOD_SYMMETRY 6-10
*BOUNDARY_ELEMENT_METHOD_WAKE 6-11

*CASE 7-1

*COMMENT 8-1

*COMPONENT 9-1
*COMPONENT_GEBOD 9-2
*COMPONENT_GEBOD_JOINT 9-4
*COMPONENT_HYBRIDIII 9-8
*COMPONENT_HYBRIDIII_JOINT 9-11

*CONSTRAINED 10-1
*CONSTRAINED_ADAPTIVITY 10-3
*CONSTRAINED_BEAM_IN_SOLID 10-4
*CONSTRAINED_BUTT_WELD 10-9
*CONSTRAINED_COORDINATE 10-12
*CONSTRAINED_EULER_IN_EULER 10-16

*CONSTRAINED_EXTRA_NODES 10-18
*CONSTRAINED_GENERALIZED_WELD 10-20
*CONSTRAINED_GLOBAL 10-32
*CONSTRAINED_INTERPOLATION....................10-34
*CONSTRAINED_INTERPOLATION_SPOTWELD 10-39
*CONSTRAINED_JOINT 10-46
*CONSTRAINED_JOINT_COOR 10-64
*CONSTRAINED_JOINT_STIFFNESS....................10-72
*CONSTRAINED_JOINT_USER_FORCE 10-86
*CONSTRAINED_LAGRANGE_IN_SOLID 10-87
*CONSTRAINED_LINEAR_GLOBAL 10-109
*CONSTRAINED_LINEAR_LOCAL....................10-112
*CONSTRAINED_LOCAL 10-115
*CONSTRAINED_MULTIPLE_GLOBAL 10-117
*CONSTRAINED_NODAL_RIGID_BODY 10-120
*CONSTRAINED_NODE_INTERPOLATION....................10-129
*CONSTRAINED_NODE_SET 10-131
*CONSTRAINED_NODE_TO_NURBS_PATCH....................10-134
*CONSTRAINED_POINTS....................10-135
*CONSTRAINED_RIGID_BODIES 10-137
*CONSTRAINED_RIGID_BODY_INSERT....................10-139
*CONSTRAINED_RIGID_BODY_STOPPERS 10-141
*CONSTRAINED_RIVET....................10-145
*CONSTRAINED_SHELL_TO_SOLID 10-147
*CONSTRAINED_SPLINE 10-149
*CONSTRAINED_SPOTWELD 10-151
*CONSTRAINED_SPR2....................10-155
*CONSTRAINED_TIE-BREAK 10-163
*CONSTRAINED_TIED_NODES_FAILURE 10-165

*CONTACT 11-1
*CONTACT 11-2
Options for *Contact keyword 11-2
Remarks 11-6
Additional cards for *CONTACT keyword 11-10
ID Card. 11-12

MPP Cards. Variables set with these cards are only active when using MPP LS-DYNA. 11-13
Card 1. 11-17
Card 2. 11-21
Card 3. 11-27
Card 4: AUTOMATIC_SURFACE_TIEBREAK....11-29
Card 4: AUTOMATIC_SURFACE_TO_SURFACE_COMPOSITE....11-34
Card 4: SINGLE_SURFACE_TIED 11-37
Card 4: AUTOMATIC_SURFACE_TO_SURFACE_TIED_WELD 11-38
Card 4: CONSTRAINT_…_TO_SURFACE....11-39
Card 4: DRAWBEAD 11-40
Card 4: ERODING_..._SURFACE....11-51
Card 4: SURFACE_INTERFERENCE 11-53
Card 4: RIGID_TO_RIGID....11-55
Card 4: TIEBREAK_NODES 11-57
Card 4: TIEBREAK_SURFACE....11-58
Card 4: CONTRACTION_JOINT 11-60
THERMAL: 11-62
THERMAL FRICTION:....11-65
ORTHO FRICTION: 11-68
Optional Card A: 11-71
Optional Card B: 11-79
Optional Card C: 11-82
Optional Card D: 11-87
Optional Card E:....11-91
Optional Card F:....11-94
General Remarks: 11-95
Contact Examples:....11-106
*CONTACT_ADD_WEAR....11-109
*CONTACT_AUTO_MOVE 11-113
*CONTACT_COUPLING....11-120
*CONTACT_ENTITY 11-122
*CONTACT_GEBOD 11-131
*CONTACT_GUIDED_CABLE 11-135
*CONTACT_INTERIOR....11-137
*CONTACT_RIGID_SURFACE 11-139
*CONTACT_1D....11-143

*CONTACT_2D 11-144
*CONTACT_2D_[SLIDING, TIED, & PENALTY] 11-147
*CONTACT_2D_[AUTOMATIC, & FORCE_TRANSDUCER] 11-153
*CONTACT_2D_NODE_TO_SOLID 11-161

*CONTROL 12-1
*CONTROL_ACCURACY 12-8
*CONTROL_ACOUSTIC 12-11
*CONTROL_ADAPSTEP 12-12
*CONTROL_ADAPTIVE 12-13
*CONTROL_ADAPTIVE_CURVE 12-24
*CONTROL_ALE 12-32
*CONTROL_BULK_VISCOSITY 12-40
*CONTROL_CHECK_SHELL 12-42
*CONTROL_COARSEN 12-44
*CONTROL_CONTACT 12-46
*CONTROL_COUPLING 12-59
*CONTROL_CPM 12-61
*CONTROL_CPU 12-63
*CONTROL_DEBUG 12-64
*CONTROL_DISCRETE_ELEMENT 12-65
*CONTROL_DYNAMIC_RELAXATION 12-70
*CONTROL_EFG 12-75
*CONTROL_ENERGY 12-77
*CONTROL_EXPLICIT_THERMAL 12-79
*CONTROL_EXPLICIT_THERMAL_ALE_COUPLING 12-80
*CONTROL_EXPLICIT_THERMAL_BOUNDARY 12-81
*CONTROL_EXPLICIT_THERMAL_CONTACT 12-82
*CONTROL_EXPLICIT_THERMAL_INITIAL 12-83
*CONTROL_EXPLICIT_THERMAL_OUTPUT 12-84
*CONTROL_EXPLICIT_THERMAL_PROPERTIES 12-86
*CONTROL_EXPLICIT_THERMAL_SOLVER 12-89
CONTROL_EXPLOSIVE_SHADOW_*{OPTION} 12-90
*CONTROL_FORMING 12-92
*CONTROL_FORMING_AUTOCHECK 12-94
*CONTROL_FORMING_AUTO_NET 12-104
*CONTROL_FORMING_AUTOPOSITION_PARAMETER 12-108

*CONTROL_FORMING_BESTFIT 12-116
*CONTROL_FORMING_INITIAL_THICKNESS 12-123
*CONTROL_FORMING_MAXID 12-126
*CONTROL_FORMING_ONESTEP 12-128
*CONTROL_FORMING_OUTPUT 12-146
*CONTROL_FORMING_PARAMETER_READ 12-156
*CONTROL_FORMING_POSITION 12-159
*CONTROL_FORMING_PRE_BENDING 12-161
*CONTROL_FORMING_PROJECTION 12-165
*CONTROL_FORMING_REMOVE_ADAPTIVE_CONSTRAINTS 12-167
*CONTROL_FORMING_SCRAP_FALL 12-170
*CONTROL_FORMING_SHELL_TO_TSHELL 12-183
*CONTROL_FORMING_STONING 12-188
*CONTROL_FORMING_TEMPLATE 12-195
*CONTROL_FORMING_TIPPING 12-202
*CONTROL_FORMING_TOLERANC 12-205
*CONTROL_FORMING_TRAVEL 12-210
*CONTROL_FORMING_TRIM_MERGE 12-212
*CONTROL_FORMING_TRIMMING 12-215
CONTROL_FORMING_UNFLANGING_*{OPTION} 12-217
*CONTROL_FORMING_USER 12-228
*CONTROL_FREQUENCY_DOMAIN 12-232
*CONTROL_HOURGLASS 12-234
*CONTROL_IMPLICIT 12-238
*CONTROL_IMPLICIT_AUTO 12-239
*CONTROL_IMPLICIT_BUCKLE 12-244
*CONTROL_IMPLICIT_CONSISTENT_MASS 12-246
*CONTROL_IMPLICIT_DYNAMICS 12-247
*CONTROL_IMPLICIT_EIGENVALUE 12-252
*CONTROL_IMPLICIT_FORMING 12-256
*CONTROL_IMPLICIT_GENERAL 12-273
*CONTROL_IMPLICIT_INERTIA_RELIEF 12-278
*CONTROL_IMPLICIT_JOINTS 12-279
*CONTROL_IMPLICIT_MODAL_DYNAMIC 12-280
*CONTROL_IMPLICIT_MODAL_DYNAMIC_DAMPING 12-282
*CONTROL_IMPLICIT_MODAL_DYNAMIC_MODE 12-284
CONTROL_IMPLICIT_MODES_*{OPTION} 12-285

*CONTROL_IMPLICIT_ROTATIONAL_DYNAMICS 12-289
*CONTROL_IMPLICIT_SOLUTION 12-293
*CONTROL_IMPLICIT_SOLVER 12-305
CONTROL_IMPLICIT_STABILIZATION_*{OPTION} 12-312
CONTROL_IMPLICIT_STATIC_CONDENSATION_*{OPTION} 12-314
*CONTROL_IMPLICIT_TERMINATION 12-317
*CONTROL_MAT 12-319
*CONTROL_MPP 12-320
*CONTROL_MPP_CONTACT_GROUPABLE 12-322
CONTROL_MPP_DECOMPOSITION_ARRANGE_PARTS_*OPTION 12-323
*CONTROL_MPP_DECOMPOSITION_AUTOMATIC 12-324
*CONTROL_MPP_DECOMPOSITION_BAGREF 12-325
*CONTROL_MPP_DECOMPOSITION_CHECK_SPEED 12-326
CONTROL_MPP_DECOMPOSITION_CONTACT_DISTRIBUTE_*OPTION 12-327
*CONTROL_MPP_DECOMPOSITION_CONTACT_ISOLATE 12-328
*CONTROL_MPP_DECOMPOSITION_DISABLE_UNREF_CURVES 12-329
*CONTROL_MPP_DECOMPOSITION_DISTRIBUTE_ALE_ELEMENTS 12-330
*CONTROL_MPP_DECOMPOSITION_DISTRIBUTE_SPH_ELEMENTS 12-331
*CONTROL_MPP_DECOMPOSITION_ELCOST 12-332
*CONTROL_MPP_DECOMPOSITION_FILE 12-333
*CONTROL_MPP_DECOMPOSITION_METHOD 12-334
*CONTROL_MPP_DECOMPOSITION_NUMPROC 12-335
*CONTROL_MPP_DECOMPOSITION_OUTDECOMP 12-336
CONTROL_MPP_DECOMPOSITION_PARTS_DISTRIBUTE_*OPTION 12-337
CONTROL_MPP_DECOMPOSITION_PARTSET_DISTRIBUTE_*OPTION 12-338
*CONTROL_MPP_DECOMPOSITION_RCBLOG 12-339
*CONTROL_MPP_DECOMPOSITION_SCALE_CONTACT_COST 12-340
*CONTROL_MPP_DECOMPOSITION_SCALE_FACTOR_SPH 12-341
*CONTROL_MPP_DECOMPOSITION_SHOW 12-342
*CONTROL_MPP_DECOMPOSITION_TRANSFORMATION 12-343
*CONTROL_MPP_IO_LSTC_REDUCE 12-345
*CONTROL_MPP_IO_NOBEAMOUT 12-346
*CONTROL_MPP_IO_NOD3DUMP 12-347
*CONTROL_MPP_IO_NODUMP 12-348
*CONTROL_MPP_IO_NOFAIL 12-349
*CONTROL_MPP_IO_NOFULL 12-350
*CONTROL_MPP_IO_SWAPBYTES 12-351

*CONTROL_MPP_MATERIAL_MODEL_DRIVER 12-352
*CONTROL_MPP_PFILE..12-353
*CONTROL_NONLOCAL 12-354
*CONTROL_OUTPUT 12-355
*CONTROL_PARALLEL 12-362
*CONTROL_PORE_FLUID 12-365
*CONTROL_PORE_AIR...12-371
*CONTROL_REFINE_ALE 12-372
*CONTROL_REFINE_ALE2D 12-378
*CONTROL_REFINE_MPP_DISTRIBUTION 12-383
*CONTROL_REFINE_SHELL 12-385
*CONTROL_REFINE_SOLID 12-390
*CONTROL_REMESHING..12-395
*CONTROL_REQUIRE_REVISION 12-400
*CONTROL_RIGID 12-402
*CONTROL_SHELL..12-406
*CONTROL_SOLID 12-419
*CONTROL_SOLUTION 12-422
*CONTROL_SPH..12-424
*CONTROL_SPOTWELD_BEAM 12-429
*CONTROL_START..12-433
*CONTROL_STAGED_CONSTRUCTION 12-434
*CONTROL_STEADY_STATE_ROLLING...12-437
*CONTROL_STRUCTURED 12-439
*CONTROL_SUBCYCLE 12-440
*CONTROL_TERMINATION...12-442
*CONTROL_THERMAL_EIGENVALUE 12-444
*CONTROL_THERMAL_NONLINEAR...12-445
*CONTROL_THERMAL_SOLVER 12-447
*CONTROL_THERMAL_TIMESTEP...12-453
*CONTROL_TIMESTEP...12-455
*CONTROL_UNITS 12-463

*DAMPING 13-1
DAMPING_FREQUENCY_RANGE_*{OPTION}..13-2
*DAMPING_GLOBAL 13-5
*DAMPING_PART_MASS 13-7

*DAMPING_PART_STIFFNESS 13-10
*DAMPING_RELATIVE 13-12
*DATABASE 14-1
*DATABASE 14-3
*DATABASE_ALE 14-15
*DATABASE_ALE_MAT 14-17
*DATABASE_BINARY 14-18
*DATABASE_BINARY_D3PROP 14-26
*DATABASE_CPM_SENSOR 14-27
*DATABASE_CROSS_SECTION 14-30
*DATABASE_EXTENT 14-35
*DATABASE_EXTENT_AVS 14-36
DATABASE_EXTENT_BINARY_*{OPTION} 14-41
*DATABASE_EXTENT_D3PART 14-55
*DATABASE_EXTENT_INTFOR 14-58
*DATABASE_EXTENT_MOVIE 14-61
*DATABASE_EXTENT_MPGS 14-62
DATABASE_EXTENT_SSSTAT_*OPTION 14-63
*DATABASE_FATXML 14-65
*DATABASE_FORMAT 14-66
*DATABASE_FREQUENCY_ASCII 14-67
*DATABASE_FREQUENCY_BINARY 14-69
*DATABASE_FSI 14-73
*DATABASE_FSI_SENSOR 14-78
*DATABASE_HISTORY 14-81
*DATABASE_MASSOUT 14-86
*DATABASE_NODAL_FORCE_GROUP 14-87
*DATABASE_PAP_OUTPUT 14-88
*DATABASE_PROFILE 14-89
*DATABASE_PWP_FLOW 14-92
*DATABASE_PWP_OUTPUT 14-93
*DATABASE_RCFORC_MOMENT 14-94
*DATABASE_RECOVER_NODE 14-95
*DATABASE_SPRING_FORWARD 14-97
*DATABASE_SUPERPLASTIC_FORMING 14-98
*DATABASE_TRACER 14-99

*DATABASE_TRACER_GENERATE....14-102

*DEFINE....15-1

*DEFINE_ADAPTIVE_SOLID_TO_DES....15-5

*DEFINE_ADAPTIVE_SOLID_TO_SPH....15-8

*DEFINE_BOX....15-12

*DEFINE_BOX_ADAPTIVE....15-15

*DEFINE_BOX_COARSEN....15-22

*DEFINE_BOX_DRAWBEAD....15-25

*DEFINE_BOX_SPH....15-27

*DEFINE_CONNECTION_PROPERTIES....15-30

*DEFINE_CONSTRUCTION_STAGES....15-38

*DEFINE_CONTACT_EXCLUSION....15-39

*DEFINE_CONTACT_VOLUME....15-41

*DEFINE_COORDINATE_NODES....15-44

*DEFINE_COORDINATE_SYSTEM....15-46

*DEFINE_COORDINATE_VECTOR....15-52

*DEFINE_CPM_BAG_INTERACTION....15-54

*DEFINE_CPM_CHAMBER....15-55

*DEFINE_CPM_GAS_PROPERTIES....15-58

*DEFINE_CPM_VENT....15-60

*DEFINE_CURVE....15-63

*DEFINE_CURVE_BOX_ADAPTIVITY....15-67

*DEFINE_CURVE_COMPENSATION_CONSTRAINT....15-72

*DEFINE_CURVE_DRAWBEAD....15-78

*DEFINE_CURVE_DUPLICATE....15-81

*DEFINE_CURVE_ENTITY....15-82

*DEFINE_CURVE_FEEDBACK....15-84

*DEFINE_CURVE_FLC....15-87

*DEFINE_CURVE_FUNCTION....15-90

*DEFINE_CURVE_SMOOTH....15-107

*DEFINE_CURVE_TRIM....15-109

*DEFINE_DEATH_TIMES....15-128

*DEFINE_DE_ACTIVE_REGION....15-131

*DEFINE_DE_BOND....15-132

*DEFINE_DE_BY_PART....15-134

*DEFINE_DE_HBOND....15-136

DEFINE_DE_INJECTION_*{OPTION} 15-140
*DEFINE_DE_MASSFLOW_PLANE 15-143
*DEFINE_DE_TO_BEAM_COUPLING 15-144
*DEFINE_DE_TO_SURFACE_COUPLING 15-146
*DEFINE_DE_TO_SURFACE_TIED 15-150
*DEFINE_ELEMENT_DEATH 15-152
*DEFINE_ELEMENT_GENERALIZED_SHELL 15-154
*DEFINE_ELEMENT_GENERALIZED_SOLID 15-159
*DEFINE_FABRIC_ASSEMBLIES 15-162
*DEFINE_FILTER 15-164
*DEFINE_FORMING_BLANKMESH 15-166
*DEFINE_FORMING_CLAMP 15-172
*DEFINE_FORMING_CONTACT 15-177
*DEFINE_FRICTION 15-179
*DEFINE_FRICTION_ORIENTATION 15-182
*DEFINE_FUNCTION 15-189
*DEFINE_FUNCTION_TABULATED 15-192
*DEFINE_GROUND_MOTION 15-194
*DEFINE_HAZ_PROPERTIES 15-195
*DEFINE_HAZ_TAILOR_WELDED_BLANK 15-198
*DEFINE_HEX_SPOTWELD_ASSEMBLY 15-199
*DEFINE_LANCE_SEED_POINT_COORDINATES 15-201
*DEFINE_MATERIAL_HISTORIES 15-203
*DEFINE_MULTI_DRAWBEADS_IGES 15-207
*DEFINE_PBLAST_AIRGEO 15-209
*DEFINE_PBLAST_GEOMETRY 15-212
*DEFINE_PLANE 15-214
*DEFINE_POROUS 15-216
*DEFINE_PRESSURE_TUBE 15-219
*DEFINE_REGION 15-222
*DEFINE_SD_ORIENTATION 15-226
*DEFINE_SET_ADAPTIVE 15-228
*DEFINE_SPH_ACTIVE_REGION 15-229
*DEFINE_SPH_DE_COUPLING 15-234
*DEFINE_SPH_INJECTION 15-236
*DEFINE_SPH_TO_SPH_COUPLING 15-237
DEFINE_SPOTWELD_FAILURE_*{OPTION} 15-240

*DEFINE_SPOTWELD_FAILURE_RESULTANTS....15-244
*DEFINE_SPOTWELD_MULTISCALE....15-246
*DEFINE_SPOTWELD_RUPTURE_PARAMETER....15-247
*DEFINE_SPOTWELD_RUPTURE_STRESS....15-250
*DEFINE_STAGED_CONSTRUCTION_PART....15-252
DEFINE_STOCHASTIC_ELEMENT_*OPTION....15-254
*DEFINE_STOCHASTIC_VARIATION....15-255
*DEFINE_TABLE....15-261
*DEFINE_TABLE_2D....15-264
*DEFINE_TABLE_3D....15-266
*DEFINE_TABLE_MATRIX....15-268
*DEFINE_TARGET_BOUNDARY....15-272
*DEFINE_TRACER_PARTICLES_2D....15-274
*DEFINE_TRANSFORMATION....15-275
*DEFINE_TRIM_SEED_POINT_COORDINATES....15-280
*DEFINE_VECTOR....15-282
*DEFINE_VECTOR_NODES....15-283
EXAMPLES....15-284

*DEFORMABLE_TO_RIGID....16-1
*DEFORMABLE_TO_RIGID....16-2
*DEFORMABLE_TO_RIGID_AUTOMATIC....16-3
*DEFORMABLE_TO_RIGID_INERTIA....16-9

*ELEMENT....17-1
*ELEMENT_BLANKING....17-3
*ELEMENT_BEAM....17-4
*ELEMENT_BEAM_PULLEY....17-18
*ELEMENT_BEAM_SOURCE....17-20
*ELEMENT_BEARING....17-22
*ELEMENT_DIRECT_MATRIX_INPUT....17-26
*ELEMENT_DISCRETE....17-29
*ELEMENT_DISCRETE_SPHERE....17-32
*ELEMENT_GENERALIZED_SHELL....17-34
*ELEMENT_GENERALIZED_SOLID....17-36
*ELEMENT_INERTIA....17-38
*ELEMENT_INTERPOLATION_SHELL....17-41
*ELEMENT_INTERPOLATION_SOLID....17-44

*ELEMENT_LANCING 17-47
*ELEMENT_MASS 17-59
*ELEMENT_MASS_MATRIX 17-60
*ELEMENT_MASS_PART 17-62
*ELEMENT_PLOTEL 17-64
*ELEMENT_SEATBELT 17-65
*ELEMENT_SEATBELT_ACCELEROMETER 17-68
*ELEMENT_SEATBELT_PRETENSIONER 17-70
*ELEMENT_SEATBELT_RETRACTOR 17-76
*ELEMENT_SEATBELT_SENSOR 17-81
*ELEMENT_SEATBELT_SLIPRING 17-85
*ELEMENT_SHELL 17-91
*ELEMENT_SHELL_NURBS_PATCH 17-100
*ELEMENT_SHELL_SOURCE_SINK 17-108
*ELEMENT_SOLID 17-109
*ELEMENT_SOLID_NURBS_PATCH 17-117
*ELEMENT_SPH 17-122
*ELEMENT_TRIM 17-124
ELEMENT_TSHELL_*{OPTION} 17-125

*END 18-129

*EOS 19-1

*FREQUENCY_DOMAIN 20-1
*FREQUENCY_DOMAIN_ACCELERATION_UNIT 20-2
*FREQUENCY_DOMAIN_ACOUSTIC_BEM 20-4
*FREQUENCY_DOMAIN_ACOUSTIC_FEM 20-17
*FREQUENCY_DOMAIN_ACOUSTIC_FRINGE_PLOT 20-23
*FREQUENCY_DOMAIN_ACOUSTIC_INCIDENT_WAVE 20-26
*FREQUENCY_DOMAIN_ACOUSTIC_SOUND_SPEED 20-28
*FREQUENCY_DOMAIN_FRF 20-29
*FREQUENCY_DOMAIN_MODE 20-37
*FREQUENCY_DOMAIN_PATH 20-39
*FREQUENCY_DOMAIN_RANDOM_VIBRATION 20-41
*FREQUENCY_DOMAIN_RESPONSE_SPECTRUM 20-55
*FREQUENCY_DOMAIN_SSD 20-60

*HOURGLASS 21-1

*INCLUDE....22-1

*INCLUDE....22-2

*INCLUDE_AUTO_OFFSET....22-15

*INCLUDE_COMPENSATION....22-17

*INCLUDE_MULTISCALE_SPOTWELD....22-27

*INCLUDE_TRIM....22-30

*INCLUDE_UNITCELL....22-32

*INITIAL....23-1

*INITIAL_AIRBAG_PARTICLE_POSITION....23-3

*INITIAL_ALE_MAPPING....23-4

*INITIAL_AXIAL_FORCE_BEAM....23-8

*INITIAL_CONTACT_WEAR....23-10

*INITIAL_CRASHFRONT....23-12

*INITIAL_DETONATION....23-13

*INITIAL_EOS_ALE....23-17

INITIAL_FATIGUE_DAMAGE_RATIO_*{OPTION}....23-19

*INITIAL_FIELD_SOLID....23-21

INITIAL_FOAM_REFERENCE_GEOMETRY_*OPTION....23-23

*INITIAL_GAS_MIXTURE....23-25

*INITIAL_HYDROSTATIC_ALE....23-27

*INITIAL_IMPULSE_MINE....23-30

*INITIAL_INTERNAL_DOF_SOLID....23-34

*INITIAL_LAG_MAPPING....23-36

*INITIAL_MOMENTUM....23-39

*INITIAL_PWP_DEPTH....23-40

*INITIAL_STRAIN_SHELL....23-41

*INITIAL_STRAIN_SOLID....23-43

*INITIAL_STRAIN_TSHELL....23-44

*INITIAL_STRESS_BEAM....23-45

*INITIAL_STRESS_DEPTH....23-49

*INITIAL_STRESS_SECTION....23-51

*INITIAL_STRESS_SHELL....23-53

INITIAL_STRESS_SOLID_*{OPTION}....23-58

*INITIAL_STRESS_SPH....23-62

*INITIAL_STRESS_TSHELL....23-63

*INITIAL_TEMPERATURE....23-66

*INITIAL_VEHICLE_KINEMATICS 23-68
*INITIAL_VELOCITY 23-71
*INITIAL_VELOCITY_NODE 23-74
*INITIAL_VELOCITY_RIGID_BODY 23-75
*INITIAL_VELOCITY_GENERATION 23-76
*INITIAL_VELOCITY_GENERATION_START_TIME 23-80
*INITIAL_VOID 23-81
*INITIAL_VOLUME_FRACTION_GEOMETRY 23-85

*INTEGRATION 24-1
*INTEGRATION_BEAM 24-2
*INTEGRATION_SHELL 24-17

*INTERFACE....25-1
*INTERFACE_BLANKSIZE 25-3
*INTERFACE_COMPENSATION_NEW 25-36
Compensation Methods Overview: 25-41
Preventing Undercut 25-43
Iterative spring back compensation 25-43
Iterative compensation applied during die construction 25-47
Compensation of localized regions 25-47
Accelerated spring back compensation (ASC)....25-47
Line die compensation....25-48
Compensation of trim dies (trim die nesting)....25-49
Local smoothing of tooling mesh: 25-49
Compensation with symmetric boundary condition....25-50
Global compensation using the original tool mesh 25-51
Updating compensated tool with small amount of part shape change 25-52
Option REFINE_RIGID: rigid tool mesh refinement for a better convergence 25-52
*INTERFACE_COMPONENT_FILE....25-61
*INTERFACE_COMPONENT....25-63
*INTERFACE_DE_HBOND 25-65
*INTERFACE_LINKING_DISCRETE_NODE 25-68
*INTERFACE_LINKING_EDGE....25-69
*INTERFACE_LINKING_FILE 25-70
*INTERFACE_LINKING_NODE 25-71
*INTERFACE_LINKING_SEGMENT 25-74
*INTERFACE_SPRINGBACK 25-75

*INTERFACE_SSI 25-82
*INTERFACE_SSI_AUX 25-86
*INTERFACE_SSI_AUX_EMBEDDED 25-87
*INTERFACE_SSI_STATIC 25-89
*INTERFACE_WELDLINE_DEVELOPMENT 25-91

*KEYWORD 26-1

*LOAD 27-1
*LOAD_ALE_CONVECTION 27-4
*LOAD_BEAM 27-6
*LOAD_BLAST 27-8
*LOAD_BLAST_ENHANCED 27-11
*LOAD_BLAST_SEGMENT 27-17
*LOAD_BLAST_SEGMENT_SET 27-18
*LOAD_BODY 27-19
*LOAD_BODY_GENERALIZED 27-24
*LOAD_BODY_POROUS 27-28
*LOAD_BRODE 27-31
*LOAD_DENSITY_DEPTH 27-33
*LOAD_ERODING_PART_SET 27-35
*LOAD_GRAVITY_PART 27-37
*LOAD_HEAT_CONTROLLER 27-39
*LOAD_HEAT_GENERATION 27-40
*LOAD_MASK 27-42
*LOAD_MOTION_NODE 27-45
*LOAD_MOVING_PRESSURE 27-47
*LOAD_NODE 27-49
*LOAD_REMOVE_PART 27-52
*LOAD_RIGID_BODY 27-54
*LOAD_SEGMENT 27-57
*LOAD_SEGMENT_CONTACT_MASK 27-62
*LOAD_SEGMENT_FILE 27-64
*LOAD_SEGMENT_FSILNK 27-67
*LOAD_SEGMENT_NONUNIFORM 27-70
*LOAD_SEGMENT_SET 27-73
*LOAD_SEGMENT_SET_ANGLE 27-75
*LOAD_SEGMENT_SET_NONUNIFORM 27-77

*LOAD_SEISMIC_SSI....27-80
*LOAD_SHELL 27-84
*LOAD_SPCFORC 27-87
*LOAD_SSA....27-88
*LOAD_STEADY_STATE_ROLLING 27-92
*LOAD_STIFFEN_PART 27-96
*LOAD_SUPERPLASTIC_FORMING 27-98
*LOAD_SURFACE_STRESS 27-104
*LOAD_THERMAL....27-106
*LOAD_THERMAL_CONSTANT 27-107
*LOAD_THERMAL_CONSTANT_ELEMENT 27-109
*LOAD_THERMAL_CONSTANT_NODE 27-110
*LOAD_THERMAL_D3PLOT 27-111
*LOAD_THERMAL_LOAD_CURVE 27-112
*LOAD_THERMAL_TOPAZ 27-113
*LOAD_THERMAL_VARIABLE....27-114
*LOAD_THERMAL_VARIABLE_BEAM....27-116
*LOAD_THERMAL_VARIABLE_ELEMENT 27-118
*LOAD_THERMAL_VARIABLE_NODE....27-119
*LOAD_THERMAL_VARIABLE_SHELL....27-120
*LOAD_VOLUME_LOSS 27-122

*MODULE....28-124
*MODULE_LOAD 28-125
File name of the library to be loaded, 80 characters maximum. 28-125
*MODULE_PATH....28-127
*MODULE_USE....28-128

*NODE....29-1
*NODE....29-2
*NODE_MERGE_SET....29-4
*NODE_MERGE_TOLERANCE 29-5
*NODE_RIGID_SURFACE....29-6
*NODE_SCALAR....29-7
*NODE_THICKNESS....29-9
*NODE_TO_TARGET_VECTOR 29-11
*NODE_TRANSFORM....29-12

*PARAMETER 30-1

*PARAMETER 30-2
*PARAMETER_DUPLICATION 30-5
*PARAMETER_EXPRESSION 30-6
*PARAMETER_TYPE 30-9

*PART 31-1
*PART 31-2
*PART_ADAPTIVE_FAILURE 31-14
*PART_ANNEAL 31-15
*PART_COMPOSITE 31-16
*PART_DUPLICATE 31-24
*PART_MODES 31-26
*PART_MOVE 31-31
*PART_SENSOR 31-34
*PART_STACKED_ELEMENTS 31-35

*PARTICLE_BLAST 32-1

*PERTURBATION 33-1
*PERTURBATION 33-2

*RAIL 34-1
*RAIL_TRACK 34-2
*RAIL_TRAIN 34-8

*RIGIDWALL 35-1
*RIGIDWALL_FORCE_TRANSDUCER 35-2
*RIGIDWALL_GEOMETRIC 35-4
*RIGIDWALL_PLANAR 35-13

*SECTION 36-1
*SECTION_ALE1D 36-2
*SECTION_ALE2D 36-4
*SECTION_BEAM 36-6
*SECTION_BEAM_AISC 36-24
*SECTION_DISCRETE 36-27
*SECTION_POINT_SOURCE 36-30
*SECTION_POINT_SOURCE_MIXTURE 36-33
*SECTION_SEATBELT 36-39
*SECTION_SHELL 36-41
*SECTION_SOLID 36-60

*SECTION_SPH 36-76
*SECTION_TSHELL 36-79

*SENSOR 37-1
*SENSOR_CONTROL 37-3
*SENSOR_CPM_AIRBAG 37-7
*SENSOR_DEFINE_CALC-MATH 37-8
*SENSOR_DEFINE_ELEMENT 37-10
*SENSOR_DEFINE_FORCE 37-13
*SENSOR_DEFINE_FUNCTION 37-15
*SENSOR_DEFINE_MISC 37-16
*SENSOR_DEFINE_NODE 37-18
*SENSOR_SWITCH 37-21
*SENSOR_SWITCH_CALC-LOGIC 37-22
*SENSOR_SWITCH_SHELL_TO_VENT 37-24

*SET 38-1
*SET_BEAM 38-3
*SET_BEAM_ADD 38-6
*SET_BEAM_INTERSECT 38-7
*SET_BOX 38-8
*SET_DISCRETE 38-9
*SET_DISCRETE_ADD 38-12
*SET_MODE 38-13
*SET_MULTI 38-15
*SET_MULTI-MATERIAL_GROUP_LIST 38-15
*SET_NODE 38-17
*SET_NODE_ADD 38-23
*SET_NODE_INTERSECT 38-26
*SET_PART 38-27
*SET_PART_ADD 38-31
*SET_SEGMENT 38-33
*SET_SEGMENT_ADD 38-40
*SET_SEGMENT_INTERSECT 38-41
*SET_2D_SEGMENT 38-42
*SET_SHELL 38-44
*SET_SHELL_ADD 38-49
*SET_SHELL_INTERSECT 38-50

*SET_SOLID 38-51
*SET_SOLID_ADD 38-56
*SET_SOLID_INTERSECT 38-57
*SET_TSHELL 38-58

*TERMINATION 39-1
*TERMINATION_BODY 39-2
*TERMINATION_CONTACT 39-3
*TERMINATION_CURVE 39-4
*TERMINATION_DELETED_SHELLS 39-5
*TERMINATION_DELETED_SOLIDS 39-6
*TERMINATION_NODE 39-7
*TERMINATION_SENSOR 39-8

*TITLE 40-1

*USER 41-1
*USER_INTERFACE 41-1
*USER_LOADING 41-4
*USER_LOADING_SET 41-5

Restart Input Data 42-1
*CHANGE 42-4
BOUNDARY_CONDITION 42-4
CONTACT_SMALL_PENETRATION 42-5
CURVE_DEFINITION 42-5
OUTPUT 42-6
RIGIDWALL_GEOMETRIC 42-6
RIGIDWALL_PLANAR 42-6
RIGID_BODY_CONSTRAINT 42-6
RIGID_BODY_STOPPERS 42-8
STATUS_REPORT_FREQUENCY 42-10
THERMAL_PARAMETERS 42-11
VELOCITY 42-12
VELOCITY_GENERATION 42-13
VELOCITY_NODE 42-14
VELOCITY_RIGID_BODY 42-15
VELOCITY_ZERO 42-15
*CONTROL_DYNAMIC_RELAXATION 42-16

*CONTROL_SHELL 42-18
*CONTROL_TERMINATION 42-20
*CONTROL_TIMESTEP 42-21
*DAMPING_GLOBAL 42-23
*DATABASE 42-24
*DATABASE_BINARY 42-26
DELETE_*OPTION 42-27
*INTERFACE_SPRINGBACK_LSDYNA 42-29
*RIGID_DEFORMABLE 42-31
*RIGID_DEFORMABLE_CONTROL 42-32
*RIGID_DEFORMABLE_D2R 42-33
*RIGID_DEFORMABLE_R2D 42-34
*STRESS_INITIALIZATION 42-35
*TERMINATION 42-37
*TITLE 42-40

REFERENCES 43-1

APPENDIX A: User Defined Materials 44-1

APPENDIX B: User Defined Equation of State 45-1

APPENDIX C: User Defined Element Interface for Solids and Shells 46-1
General overview 46-2
Resultant/discrete elements 46-5
Extra degrees of freedom 46-6
Related keywords: 46-7
The *SECTION_SHELL card 46-7
The *SECTION_SOLID card 46-9
Sample User Shell Element 101 (Belytschko-Tsay shell) 46-10
Sample User Solid Element 101 (constant stress solid) 46-13
Examples 46-17

APPENDIX D: User Defined Airbag Sensor 47-1

APPENDIX E: User Defined Solution Control 48-1

APPENDIX F: User Defined Interface Control 49-1

APPENDIX G: User Defined Interface Friction and Conductivity 50-1

APPENDIX H: User Defined Thermal Material Model 51-1

APPENDIX I: Occupant Simulation Including the Coupling to the CAL3D and MADYMO programs....52-1

INTRODUCTION 52-1

THE LS-DYNA/OCCUPANT SIMULATION PROGRAM LINK..52-1

AIRBAG MODELING 52-2

COMMON ERRORS..52-3

APPENDIX J: Interactive Graphics Commands..53-1

APPENDIX K: Interactive Material Model Driver 54-1

INTRODUCTION 54-1

INPUT DEFINITION 54-1

INTERACTIVE DRIVER COMMANDS 54-2

APPENDIX L: VDA Database 55-1

APPENDIX M: Commands for Two-Dimensional Rezoning 56-1

APPENDIX N: Rigid-Body Dummies..57-1

APPENDIX O: LS-DYNA MPP User Guide 58-1

Supported Features 58-1

Contact Interfaces..58-1

Output Files and Post-Processing..58-2

Parallel Specific Options..58-3

PFILE..58-4

Execution of MPP/LS-DYNA 58-14

APPENDIX P: Implicit Solver..59-1

INTRODUCTION 59-1

NONLINEAR IMPLICIT ANALYSIS 59-2

LINEAR EQUATION SOLVER 59-12

ELEMENTS AND MATERIALS 59-16

CONTACTS..59-17

TROUBLESHOOTING CONVERGENCE PROBLEMS..59-22

CHECKLIST..59-25

APPENDIX Q: User Defined Weld Failure..60-1

APPENDIX R: User Defined Cohesive Model 61-1

APPENDIX S: User Defined Boundary Flux..62-1

APPENDIX T: Metal Forming Glossary 63-3

A Typical DRAW DIE engineering PROCESS 63-3

Types of draw dies....63-3
Types of flanging dies....63-4
Types of hemming dies....63-4

INTRODUCTION

CHRONOLOGICAL HISTORY

DYNA3D originated at the Lawrence Livermore National Laboratory [Hallquist 1976]. The early applications were primarily for the stress analysis of structures subjected to a variety of impact loading. These applications required what was then significant computer resources, and the need for a much faster version was immediately obvious. Part of the speed problem was related to the inefficient implementation of the element technology which was further aggravated by the fact that supercomputers in 1976 were much slower than today's PC. Furthermore, the primitive sliding interface treatment could only treat logically regular interfaces that are uncommon in most finite element discretizations of complicated three-dimensional geometries; consequently, defining a suitable mesh for handling contact was often very difficult. The first version contained trusses, membranes, and a choice of solid elements. The solid elements ranged from a one-point quadrature eight-noded element with hourglass control to a twenty-noded element with eight integration points. Due to the high cost of the twenty node solid, the zero energy modes related to the reduced 8-point integration, and the high frequency content which drove the time step size down, higher order elements were all but abandoned in later versions of DYNA3D. A two-dimensional version, DYNA2D, was developed concurrently.

A new version of DYNA3D was released in 1979 that was programmed to provide near optimal speed on the CRAY-1 supercomputers, contained an improved sliding interface treatment that permitted triangular segments and was an order of magnitude faster than the previous contact treatment. The 1979 version eliminated structural and higher order solid elements and some of the material models of the first version. This version also included an optional element-wise implementation of the integral difference method developed by Wilkins et al. [1974].

The 1981 version [Hallquist 1981a] evolved from the 1979 version. Nine additional material models were added to allow a much broader range of problems to be modeled including explosive-structure and soil-structure interactions. Body force loads were implemented for angular velocities and base accelerations. A link was also established from the 3D Eulerian code, JOY [Couch, et. al., 1983] for studying the structural response to impacts by penetrating projectiles. An option was provided for storing element data on disk thereby doubling the capacity of DYNA3D.

The 1982 version of DYNA3D [Hallquist 1982] accepted DYNA2D [Hallquist 1980] material input directly. The new organization was such that equations of state and constitutive models of any complexity could be easily added. Complete vectorization of the material models had been nearly achieved with about a 10 percent increase in execution speed over the 1981 version.

In the 1986 version of DYNA3D [Hallquist and Benson 1986], many new features were added, including beams, shells, rigid bodies, single surface contact, interface friction, discrete springs and dampers, optional hourglass treatments, optional exact volume integration, and VAX/ VMS, IBM, UNIX, COS operating systems compatibility, that greatly expanded its range of applications. DYNA3D thus became the first code to have a general single surface contact algorithm.

In the 1987 version of DYNA3D [Hallquist and Benson 1987] metal forming simulations and composite analysis became a reality. This version included shell thickness changes, the Belytschko-Tsay shell element [Belytschko and Tsay, 1981], and dynamic relaxation. Also included were non-reflecting boundaries, user specified integration rules for shell and beam elements, a layered composite damage model, and single point constraints.

New capabilities added in the 1988 DYNA3D [Hallquist 1988] version included a cost effective resultant beam element, a truss element, a C^0 triangular shell, the BCIZ triangular shell [Bazeley et al. 1965], mixing of element formulations in calculations, composite failure modeling for solids, noniterative plane stress plasticity, contact surfaces with spot welds, tie break sliding surfaces, beam surface contact, finite stonewalls, stonewall reaction forces, energy calculations for all elements, a crushable foam constitutive model, comment cards in the input, and one-dimensional slidelines.

By the end of 1988 it was obvious that a much more concentrated effort would be required in the development of this software if problems in crashworthiness were to be properly solved; therefore, Livermore Software Technology Corporation was founded to continue the development of DYNA3D as a commercial version called LS-DYNA3D which was later shortened to LS-DYNA. The 1989 release introduced many enhanced capabilities including a one-way treatment of slide surfaces with voids and friction; cross-sectional forces for structural elements; an optional user specified minimum time step size for shell elements using elastic and elastoplastic material models; nodal accelerations in the time history database; a compressible Mooney-Rivlin material model; a closed-form update shell plasticity model; a general rubber material model; unique penalty specifications for each slide surface; external work tracking; optional time step criterion for 4-node shell elements; and internal element sorting to allow full vectorization of right-hand-side force assembly.

During the last ten years, considerable progress has been made as may be seen in the chronology of the developments which follows.

Capabilities added in 1989-1990:

- arbitrary node and element numbers,

- fabric model for seat belts and airbags,
- composite glass model,
- vectorized type 3 contact and single surface contact,
- many more I/O options,
- all shell materials available for 8 node thick shell,
- strain rate dependent plasticity for beams,
- fully vectorized iterative plasticity,
- interactive graphics on some computers,
- nodal damping,
- shell thickness taken into account in shell type 3 contact,
- shell thinning accounted for in type 3 and type 4 contact,
- soft stonewalls,
- print suppression option for node and element data,
- massless truss elements, rivets – based on equations of rigid body dynamics,
- massless beam elements, spot welds – based on equations of rigid body dynamics,
- expanded databases with more history variables and integration points,
- force limited resultant beam,
- rotational spring and dampers, local coordinate systems for discrete elements,
- resultant plasticity for C^0 triangular element,
- energy dissipation calculations for stonewalls,
- hourglass energy calculations for solid and shell elements,
- viscous and Coulomb friction with arbitrary variation over surface,
- distributed loads on beam elements,
- Cowper and Symonds strain rate model,
- segmented stonewalls,
- stonewall Coulomb friction,
- stonewall energy dissipation,
- airbags (1990),
- nodal rigid bodies,
- automatic sorting of triangular shells into C^0 groups,
- mass scaling for quasi static analyses,

- user defined subroutines,
- warpage checks on shell elements,
- thickness consideration in all contact types,
- automatic orientation of contact segments,
- sliding interface energy dissipation calculations,
- nodal force and energy database for applied boundary conditions,
- defined stonewall velocity with input energy calculations,

Capabilities added in 1991-1992:

- rigid/deformable material switching,
- rigid bodies impacting rigid walls,
- strain-rate effects in metallic honeycomb model 26,
- shells and beams interfaces included for subsequent component analyses,
- external work computed for prescribed displacement/velocity/accelerations,
- linear constraint equations,
- MPGS database,
- MOVIE database,
- Slideline interface file,
- automated contact input for all input types,
- automatic single surface contact without element orientation,
- constraint technique for contact,
- cut planes for resultant forces,
- crushable cellular foams,
- urethane foam model with hysteresis,
- subcycling,
- friction in the contact entities,
- strains computed and written for the 8 node thick shells,
- "good" 4 node tetrahedron solid element with nodal rotations,
- 8 node solid element with nodal rotations,
- 2 × 2 integration for the membrane element,
- Belytschko-Schwer integrated beam,
- thin-walled Belytschko-Schwer integrated beam,

- improved TAURUS database control,
- null material for beams to display springs and seatbelts in TAURUS,
- parallel implementation on Crays and SGI computers,
- coupling to rigid body codes,
- seat belt capability.

Capabilities added in 1993-1994:

- Arbitrary Lagrangian Eulerian brick elements,
- Belytschko-Wong-Chiang quadrilateral shell element,
- Warping stiffness in the Belytschko-Tsay shell element,
- Fast Hughes-Liu shell element,
- Fully integrated thick shell element,
- Discrete 3D beam element,
- Generalized dampers,
- Cable modeling,
- Airbag reference geometry,
- Multiple jet model,
- Generalized joint stiffnesses,
- Enhanced rigid body to rigid body contact,
- Orthotropic rigid walls,
- Time zero mass scaling,
- Coupling with USA (Underwater Shock Analysis),
- Layered spot welds with failure based on resultants or plastic strain,
- Fillet welds with failure,
- Butt welds with failure,
- Automatic eroding contact,
- Edge-to-edge contact,
- Automatic mesh generation with contact entities,
- Drawbead modeling,
- Shells constrained inside brick elements,
- NIKE3D coupling for springback,
- Barlat’s anisotropic plasticity,

- Superplastic forming option,
- Rigid body stoppers,
- Keyword input,
- Adaptivity,
- First MPP (Massively Parallel) version with limited capabilities.
- Built in least squares fit for rubber model constitutive constants,
- Large hysteresis in hyperelastic foam,
- Bilhku/Dubois foam model,
- Generalized rubber model,

Capabilities added in 1995:

- Belytschko - Leviathan Shell
- Automatic switching between rigid and deformable bodies.
- Accuracy on SMP machines to give identical answers on one, two or more processors.
- Local coordinate systems for cross-section output can be specified.
- Null material for shell elements.
- Global body force loads now may be applied to a subset of materials.
- User defined loading subroutine.
- Improved interactive graphics.
- New initial velocity options for specifying rotational velocities.
- Geometry changes after dynamic relaxation can be considered for initial velocities..
- Velocities may also be specified by using material or part ID's.
- Improved speed of brick element hourglass force and energy calculations.
- Pressure outflow boundary conditions have been added for the ALE options.
- More user control for hourglass control constants for shell elements.
- Full vectorization in constitutive models for foam, models 57 and 63.
- Damage mechanics plasticity model, material 81,
- General linear viscoelasticity with 6 term prony series.
- Least squares fit for viscoelastic material constants.
- Table definitions for strain rate effects in material type 24.

- Improved treatment of free flying nodes after element failure.
- Automatic projection of nodes in CONTACT_TIED to eliminate gaps in the surface.
- More user control over contact defaults.
- Improved interpenetration warnings printed in automatic contact.
- Flag for using actual shell thickness in single surface contact logic rather than the default.
- Definition by exempted part ID's.
- Airbag to Airbag venting/segmented airbags are now supported.
- Airbag reference geometry speed improvements by using the reference geometry for the time step size calculation.
- Isotropic airbag material may now be directly for cost efficiency.
- Airbag fabric material damping is specified as the ratio of critical damping.
- Ability to attach jets to the structure so the airbag, jets, and structure to move together.
- PVM 5.1 Madymo coupling is available.
- Meshes are generated within LS-DYNA3D for all standard contact entities.
- Joint damping for translational motion.
- Angular displacements, rates of displacements, damping forces, etc. in JNT-FORC file.
- Link between LS-NIKE3D to LS-DYNA3D via *INITIAL_STRESS keywords.
- Trim curves for metal forming springback.
- Sparse equation solver for springback.
- Improved mesh generation for IGES and VDA provides a mesh that can directly be used to model tooling in metal stamping analyses.
- Capabilities added in 1996-1997 in Version 940:
- Part/Material ID's may be specified with 8 digits.
- Rigid body motion can be prescribed in a local system fixed to the rigid body.
- Nonlinear least squares fit available for the Ogden rubber model.
- Least squares fit to the relaxation curves for the viscoelasticity in rubber.
- Fu-Chang rate sensitive foam.
- 6 term Prony series expansion for rate effects in model 57-now 73
- Viscoelastic material model 76 implemented for shell elements.
- Mechanical threshold stress (MTS) plasticity model for rate effects.

- Thermoelastic-plastic material model for Hughes-Liu beam element.
- Ramberg-Osgood soil model
- Invariant local coordinate systems for shell elements are optional.
- Second order accurate stress updates.
- Four noded, linear, tetrahedron element.
- Co-rotational solid element for foam that can invert without stability problems.
- Improved speed in rigid body to rigid body contacts.
- Improved searching for the a_3, a_5 and a10 contact types.
- Invariant results on shared memory parallel machines with the *a_n* contact types.
- Thickness offsets in type 8 and 9 tie break contact algorithms.
- Bucket sort frequency can be controlled by a load curve for airbag applications.
- In automatic contact each part ID in the definition may have unique:
    - Static coefficient of friction
    - Dynamic coefficient of friction
    - Exponential decay coefficient
    - Viscous friction coefficient
    - Optional contact thickness
    - Optional thickness scale factor
    - Local penalty scale factor
- Automatic beam-to-beam, shell edge-to-beam, shell edge-to-shell edge and single surface contact algorithm.
- Release criteria may be a multiple of the shell thickness in types a_3, a_5, a10, 13, and 26 contact.
- Force transducers to obtain reaction forces in automatic contact definitions. Defined manually via segments, or automatically via part ID's.
- Searching depth can be defined as a function of time.
- Bucket sort frequency can be defined as a function of time.
- Interior contact for solid (foam) elements to prevent "negative volumes."
- Locking joint
- Temperature dependent heat capacity added to Wang-Nefske inflator models.
- Wang Hybrid inflator model [Wang, 1996] with jetting options and bag-to-bag venting.
- Aspiration included in Wang's hybrid model [Nusholtz, Wang, Wylie, 1996].

- Extended Wang's hybrid inflator with a quadratic temperature variation for heat capacities [Nusholtz, 1996].
- Fabric porosity added as part of the airbag constitutive model.
- Blockage of vent holes and fabric in contact with structure or itself considered in venting with leakage of gas.
- Option to delay airbag liner with using the reference geometry until the reference area is reached.
- Birth time for the reference geometry.
- Multi-material Euler/ALE fluids,
    - 2nd order accurate formulations.
    - Automatic coupling to shell, brick, or beam elements
    - Coupling using LS-DYNA contact options.
    - Element with fluid + void and void material
    - Element with multi-materials and pressure equilibrium
- Nodal inertia tensors.
- 2D plane stress, plane strain, rigid, and axisymmetric elements
- 2D plane strain shell element
- 2D axisymmetric shell element.
- Full contact support in 2D, tied, sliding only, penalty and constraint techniques.
- Most material types supported for 2D elements.
- Interactive remeshing and graphics options available for 2D.
- Subsystem definitions for energy and momentum output.
- Boundary element method for incompressible fluid dynamics and fluid-structure interaction problems.

Capabilities added during 1997-1998 in Version 950:

- Adaptive refinement can be based on tooling curvature with FORMING contact.
- The display of drawbeads is now possible since the drawbead data is output into the D3PLOT database.
- An adaptive box option, *DEFINE_BOX_ADAPTIVE, allows control over the refinement level and location of elements to be adapted.
- A root identification file, ADAPT.RID, gives the parent element ID for adapted elements.
- Draw bead box option, *DEFINE_BOX_DRAWBEAD, simplifies drawbead input.

- The new control option, CONTROL_IMPLICIT, activates an implicit solution scheme.
- 2D Arbitrary-Lagrangian-Eulerian elements are available.
- 2D automatic contact is defined by listing part ID's.
- 2D r-adaptivity for plane strain and axisymmetric forging simulations is available.
- 2D automatic non-interactive rezoning as in LS-DYNA2D.
- 2D plane strain and axisymmetric element with 2x2 selective-reduced integration are implemented.
- Implicit 2D solid and plane strain elements are available.
- Implicit 2D contact is available.
- The new keyword, *DELETE_CONTACT_2DAUTO, allows the deletion of 2D automatic contact definitions.
- The keyword, *LOAD_BEAM is added for pressure boundary conditions on 2D elements.
- A viscoplastic strain rate option is available for materials:
  - *MAT_PLASTIC_KINEMATIC
  - *MAT_JOHNSON_COOK
  - *MAT_POWER_LAW_PLASTICITY
  - *MAT_STRAIN_RATE_DEPENDENT_PLASTICITY
  - *MAT_PIECEWISE_LINEAR_PLASTICITY
  - *MAT_RATE_SENSITIVE_POWERLAW_PLASTICITY
  - *MAT_ZERILLI-ARMSTRONG
  - *MAT_PLASTICITY_WITH_DAMAGE
  - *MAT_PLASTICITY_COMPRESSION_TENSION
- Material model, *MAT_Plasticity_with_DAMAGE, has a piecewise linear damage curve given by a load curve ID.
- The Arruda-Boyce hyper-viscoelastic rubber model is available, see *MAT_ARRUDA_BOYCE.
- Transverse-anisotropic-viscoelastic material for heart tissue, see *MAT_HEART_TISSUE.
- Lung hyper-viscoelastic material, see *MAT_LUNG_TISSUE.
- Compression/tension plasticity model, see *MAT_Plasticity_COMPRESSION_TENSION.
- The Lund strain rate model, *MAT_STEINBERG_LUND, is added to Steinberg-Guinan plasticity model.

- Rate sensitive foam model, *MAT_FU_CHANG_FOAM, has been extended to include engineering strain rates, etc.
- Model, *MAT_MODIFIED_Piecewise_Linear_Plasticity, is added for modeling the failure of aluminum.
- Material model, *MAT_SPECIAL_ORTHOTROPIC, added for television shadow mask problems.
- Erosion strain is implemented for material type, *MAT_bamman_damage.
- The equation of state, *EOS_JWLB, is available for modeling the expansion of explosive gases.
- The reference geometry option is extended for foam and rubber materials and can be used for stress initialization, see *INITIAL_FOAM_REFERENCE_GEOMETRY.
- A vehicle positioning option is available for setting the initial orientation and velocities, see *INITIAL_VEHICLE_KINEMATICS.
- A boundary element method is available for incompressible fluid dynamics problems.
- The thermal materials work with instantaneous coefficients of thermal expansion:
    - *MAT_ELASTIC_PLASTIC_THERMAL
    - *MAT_ORTHOTROPIC_THERMAL
    - *MAT_TEMPERATURE_DEPENDENT_ORTHOTROPIC
    - *MAT_ELASTIC_WITH_VISCOSITY
- Airbag interaction flow rate versus pressure differences.
- Contact segment search option, [bricks first optional]
- A through thickness Gauss integration rule with 1-10 points is available for shell elements. Previously, 5 were available.
- Shell element formulations can be changed in a full deck restart.
- The tied interface which is based on constraint equations, TIED_SURFACE_TO_-SURFACE, can now fail if_FAILURE, is appended.
- A general failure criteria for solid elements is independent of the material type, see *MAT_ADD_EROSION
- Load curve control can be based on thinning and a flow limit diagram, see *DEFINE_CURVE_FEEDBACK.
- An option to filter the spotweld resultant forces prior to checking for failure has been added the the option, *CONSTRAINED_SPOTWELD, by appending FILTERED_FORCE, to the keyword.

- Bulk viscosity is available for shell types 1, 2, 10, and 16.
- When defining the local coordinate system for the rigid body inertia tensor a local coordinate system ID can be used. This simplifies dummy positioning.
- Prescribing displacements, velocities, and accelerations is now possible for rigid body nodes.
- One way flow is optional for segmented airbag interactions.
- Pressure time history input for airbag type, LINEAR_FLUID, can be used.
- An option is available to independently scale system damping by part ID in each of the global directions.
- An option is available to independently scale global system damping in each of the global directions.
- Added option to constrain global DOF along lines parallel with the global axes. The keyword is *CONSTRAINED_GLOBAL. This option is useful for adaptive remeshing.
- Beam end code releases are available, see *ELEMENT_BEAM.
- An initial force can be directly defined for the cable material, *MAT_CABLE_DISCRETE_BEAM. The specification of slack is not required if this option is used.
- Airbag pop pressure can be activated by accelerometers.
- Termination may now be controlled by contact, via *TERMINATION_CONTACT.
- Modified shell elements types 8, 10 and the warping stiffness option in the Belytschko-Tsay shell to ensure orthogonality with rigid body motions in the event that the shell is badly warped. This is optional in the Belytschko-Tsay shell and the type 10 shell.
- A one point quadrature brick element with an exact hourglass stiffness matrix has been implemented for implicit and explicit calculations.
- Automatic file length determination for D3PLOT binary database is now implemented. This insures that at least a single state is contained in each D3PLOT file and eliminates the problem with the states being split between files.
- The dump files, which can be very large, can be placed in another directory by specifying

  ```
  d=/home/user /test/d3dump
  ```

  on the execution line.
- A print flag controls the output of data into the MATSUM and RBDOUT files by part ID's. The option, PRINT, has been added as an option to the *PART keyword.

- Flag has been added to delete material data from the D3THDT file. See *DATABASE_EXTENT_BINARY and column 25 of the 19th control card in the structured input.
- After dynamic relaxation completes, a file is written giving the displaced state which can be used for stress initialization in later runs.

Capabilities added during 1998-2000 in Version 960:

Most new capabilities work on both the MPP and SMP versions; however, the capabilities that are implemented for the SMP version only, which were not considered critical for this release, are flagged below. These SMP unique capabilities are being extended for MPP calculations and will be available in the near future. The implicit capabilities for MPP require the development of a scalable eigenvalue solver, which is under development for a later release of LS-DYNA.

- Incompressible flow solver is available. Structural coupling is not yet implemented.
- Adaptive mesh coarsening can be done before the implicit springback calculation in metal forming applications.
- Two-dimensional adaptivity can be activated in both implicit and explicit calculations. (SMP version only)
- An internally generated smooth load curve for metal forming tool motion can be activated with the keyword: *DEFINE_CURVE_SMOOTH.
- Torsional forces can be carried through the deformable spot welds by using the contact type: *CONTACT_SPOTWELD_WITH_TORSION (SMP version only with a high priority for the MPP version if this option proves to be stable.)
- Tie break automatic contact is now available via the *CONTACT_AUTOMATIC_…_TIEBREAK options. This option can be used for glued panels. (SMP only)
- *CONTACT_RIGID_SURFACE option is now available for modeling road surfaces (SMP version only).
- Fixed rigid walls PLANAR and PLANAR_FINITE are represented in the binary output file by a single shell element.
- Interference fits can be modeled with the INTERFERENCE option in contact.
- A layered shell theory is implemented for several constitutive models including the composite models to more accurately represent the shear stiffness of laminated shells.
- Damage mechanics is available to smooth the post-failure reduction of the resultant forces in the constitutive model *MAT_SPOTWELD_DAMAGE.

- Finite elastic strain isotropic plasticity model is available for solid elements. *MAT_FINITE_ELASTIC_STRAIN_PLASTICITY.
- A shape memory alloy material is available: *MAT_SHAPE_MEMORY.
- Reference geometry for material, *MAT_MODIFIED_HONEYCOMB, can be set at arbitrary relative volumes or when the time step size reaches a limiting value. This option is now available for all element types including the fully integrated solid element.
- Non orthogonal material axes are available in the airbag fabric model. See *MAT_FABRIC.
- Other new constitutive models include for the beam elements:
    - *MAT_MODIFIED_FORCE_LIMITED
    - *MAT_SEISMIC_BEAM
    - *MAT_CONCRETE_BEAM
- for shell and solid elements:
    - *MAT_ELASTIC_VISCOPLASTIC_THERMAL
- for the shell elements:
    - *MAT_GURSON
    - *MAT_GEPLASTIC_SRATE2000
    - *MAT_ELASTIC_VISCOPLASTIC_THERMAL
    - *MAT_COMPOSITE_LAYUP
    - *MAT_COMPOSITE_LAYUP
    - *MAT_COMPOSITE_DIRECT
- for the solid elements:
    - *MAT_JOHNSON_HOLMQUIST_CERAMICS
    - *MAT_JOHNSON_HOLMQUIST_CONCRETE
    - *MAT_INV_HYPERBOLIC_SIN
    - *MAT_UNIFIED_CREEP
    - *MAT_SOIL_BRICK
    - *MAT_DRUCKER_PRAGER
    - *MAT_RC_SHEAR_WALL
- and for all element options a very fast and efficient version of the Johnson-Cook plasticity model is available:
- *MAT_SIMPLIFIED_JOHNSON_COOK
- A fully integrated version of the type 16 shell element is available for the resultant constitutive models.

- A nonlocal failure theory is implemented for predicting failure in metallic materials. The keyword *MAT_NONLOCAL activates this option for a subset of elastoplastic constitutive models.
- A discrete Kirchhoff triangular shell element (DKT) for explicit analysis with three in plane integration points is flagged as a type 17 shell element. This element has much better bending behavior than the C0 triangular element.
- A discrete Kirchhoff linear triangular and quadrilateral shell element is available as a type 18 shell. This shell is for extracting normal modes and static analysis.
- A C0 linear 4-node quadrilateral shell element is implemented as element type 20 with drilling stiffness for normal modes and static analysis.
- An assumed strain linear brick element is available for normal modes and statics.
- The fully integrated thick shell element has been extended for use in implicit calculations.
- A fully integrated thick shell element based on an assumed strain formulation is now available. This element uses a full 3D constitutive model which includes the normal stress component and, therefore, does not use the plane stress assumption.
- The 4-node constant strain tetrahedron element has been extended for use in implicit calculations.
- Relative damping between parts is available, see *DAMPING_RELATIVE (SMP only).
- Preload forces are can be input for the discrete beam elements.
- Objective stress updates are implemented for the fully integrated brick shell element.
- Acceleration time histories can be prescribed for rigid bodies.
- Prescribed motion for nodal rigid bodies is now possible.
- Generalized set definitions, i.e., SET_SHELL_GENERAL etc. provide much flexibility in the set definitions.
- The command “sw4.” will write a state into the dynamic relaxation file, D3DRLF, during the dynamic relaxation phase if the D3DRLF file is requested in the input.
- Added mass by PART ID is written into the MATSUM file when mass scaling is used to maintain the time step size, (SMP version only).
- Upon termination due to a large mass increase during a mass scaled calculation a print summary of 20 nodes with the maximum added mass is printed.
- Eigenvalue analysis of models containing rigid bodies is now available using BCSLIB-EXT solvers from Boeing. (SMP version only).

- Second order stress updates can be activated by part ID instead of globally on the *CONTROL_ACCURACY input.
- Interface frictional energy is optionally computed for heat generation and is output into the interface force file (SMP version only).
- The interface force binary database now includes the distance from the contact surface for the FORMING contact options. This distance is given after the nodes are detected as possible contact candidates. (SMP version only).
- Type 14 acoustic brick element is implemented. This element is a fully integrated version of type 8, the acoustic element (SMP version only).
- A flooded surface option for acoustic applications is available (SMP version only).
- Attachment nodes can be defined for rigid bodies. This option is useful for NVH applications.
- CONSTRAINED_POINTS tie any two points together. These points must lie on a shell elements.
- Soft constraint is available for edge to edge contact in type 26 contact.
- CONSTAINED_INTERPOLATION option for beam to solid interfaces and for spreading the mass and loads. (SMP version only).
- A database option has been added that allows the output of added mass for shell elements instead of the time step size.
- A new contact option allows the inclusion of all internal shell edges in contact type *CONTACT_GENERAL, type 26. This option is activated by adding "_INTERIOR" after the GENERAL keyword.
- A new option allows the use deviatoric strain rates rather than total rates in material model 24 for the Cowper-Symonds rate model.
- The CADFEM option for ASCII databases is now the default. Their option includes more significant figures in the output files.
- When using deformable spot welds, the added mass for spot welds is now printed for the case where global mass scaling is activated. This output is in the log file, d3hsp file, and the messag file.
- Initial penetration warnings for edge-to-edge contact are now written into the MESSAG file and the d3hsp file.
- Each compilation of LS-DYNA is given a unique version number.
- Finite length discrete beams with various local axes options are now available for material types 66, 67, 68, 93, and 95. In this implementation the absolute value of SCOOR must be set to 2 or 3 in the *SECTION_BEAM input.
- New discrete element constitutive models are available:

  - *MAT_ELASTIC_SPRING_DISCRETE_BEAM
  - *MAT_INELASTIC_SPRING_DISCRETE_BEAM
  - *MAT_ELASTIC_6DOF_SPRING_DISCRETE_BEAM
  - *MAT_INELASTIC_6DOF_SPRING_DISCRETE_BEAM

- The latter two can be used as finite length beams with local coordinate systems.
- Moving SPC's are optional in that the constraints are applied in a local system that rotates with the 3 defining nodes.
- A moving local coordinate system, CID, can be used to determine orientation of discrete beam elements.
- Modal superposition analysis can be performed after an eigenvalue analysis. Stress recovery is based on type 18 shell and brick (SMP only).
- Rayleigh damping input factor is now input as a fraction of critical damping, i.e. 0.10. The old method required the frequency of interest and could be highly unstable for large input values.
- Airbag option "SIMPLE_PRESSURE_VOLUME" allows for the constant CN to be replaced by a load curve for initialization. Also, another load curve can be defined which allows CN to vary as a function of time during dynamic relaxation. After dynamic relaxation CN can be used as a fixed constant or load curve.
- Hybrid inflator model utilizing CHEMKIN and NIST databases is now available. Up to ten gases can be mixed.
- Option to track initial penetrations has been added in the automatic SMP contact types rather than moving the nodes back to the surface. This option has been available in the MPP contact for some time. This input can be defined on the fourth card of the *CONTROL_CONTACT input and on each contact definition on the third optional card in the *CONTACT definitions.
- If the average acceleration flag is active, the average acceleration for rigid body nodes is now written into the D3THDT and NODOUT files. In previous versions of LS-DYNA, the accelerations on rigid nodes were not averaged.
- A capability to initialize the thickness and plastic strain in the crash model is available through the option *INCLUDE_STAMPED_PART, which takes the results from the LS-DYNA stamping simulation and maps the thickness and strain distribution onto the same part with a different mesh pattern.
- A capability to include finite element data from other models is available through the option, *INCLUDE_TRANSFORM. This option will take the model defined in an INCLUDE file: offset all ID's; translate, rotate, and scale the coordinates; and transform the constitutive constants to another set of units.

Features added during 2001-2002 for the 970 release of LS-DYNA:

Some of the new features, which are also listed below, were also added to later releases of version 960. Most new explicit capabilities work for both the MPP and SMP versions; however, the implicit capabilities for MPP require the development of a scalable eigenvalue solver and a parallel implementation of the constraint equations into the global matrices. This work is underway. A later release of version 970 is planned in 2003 that will be scalable for implicit solutions.

Below is list of new capabilities and features:

- MPP decomposition can be controlled using *CONTROL_MPP_DECOMPOSITION commands in the input deck.
- The MPP arbitrary Lagrangian-Eulerian fluid capability now works for airbag deployment in both SMP and MPP calculations.
- Euler-to-Euler coupling is now available through the keyword *CONSTRAINED_EULER_TO_EULER.
- Up to ten ALE multi-material groups may now be defined. The previous limit was three groups.
- Volume fractions can be automatically assigned during initialization of multi-material cells. See the GEOMETRY option of *INITIAL_VOLUME_FRACTION.
- A new ALE smoothing option is available to accurately predict shock fronts.
- DATABASE_FSI activates output of fluid-structure interaction data to ASCII file DBFSI.
- Point sources for airbag inflators are available. The origin and mass flow vector of these inflators are permitted to vary with time.
- A majority of the material models for solid materials are available for calculations using the SPH (Smooth Particle Hydrodynamics) option.
- The Element Free Galerkin method (EFG or meshfree) is available for two-dimensional and three-dimensional solids. This new capability is not yet implemented for MPP applications.
- A binary option for the ASCII files is now available. This option applies to all ASCII files and results in one binary file that contains all the information normally spread between a large number of separate ASCII files.
- Material models can now be defined by numbers rather than long names in the keyword input. For example the keyword *MAT_PIECEWISE_LINEAR_PLASTICITY can be replaced by the keyword: *MAT_024.
- An embedded NASTRAN reader for direct reading of NASTRAN input files is available. This option allows a typical input file for NASTRAN to be read directly and used without additional input. See the *INCLUDE_NASTRAN keyword.

- Names in the keyword input can represent numbers if the *PARAMETER option is used to relate the names and the corresponding numbers.
- Model documentation for the major ASCII output files is now optional. This option allows descriptors to be included within the ASCII files that document the contents of the file.
- ID's have been added to the following keywords:
    - *BOUNDARY_PRESCRIBED_MOTION
    - *BOUNDARY_PRESCRIBED_SPC
    - *CONSTRAINED_GENERALIZED_WELD
    - *CONSTRAINED_JOINT
    - *CONSTRAINED_NODE_SET
    - *CONSTRAINED_RIVET
    - *CONSTRAINED_SPOTWELD
    - *DATABASE_CROSS_SECTION
    - *ELEMENT_MASS
- Penetration warnings for the contact option, ignore initial penetration, î are added as an option. Previously, no penetration warnings were written when this contact option was activated.
- Penetration warnings for nodes in-plane with shell mid-surface are printed for the AUTOMATIC contact options. Previously, these nodes were ignored since it was assumed that they belonged to a tied interface where an offset was not used; consequently, they should not be treated in contact.
- For the arbitrary spot weld option, the spot welded nodes and their contact segments are optionally written into the d3hsp file. See *CONTROL_CONTACT.
- For the arbitrary spot weld option, if a segment cannot be found for the spot welded node, an option now exists to error terminate. See *CONTROL_CONTACT.
- Spot weld resultant forces are written into the SWFORC file for solid elements used as spot welds.
- Solid materials have now been added to the failed element report.
- A new option for terminating a calculation is available, *TERMINATION_-CURVE.
- A 10-noded tetrahedron solid element is available with either a 4 or 5 point integration rule. This element can also be used for implicit solutions.
- A new 4 node linear shell element is available that is based on Wilson's plate element combined with a Pian-Sumihara membrane element. This is shell type 21.

- A shear panel element has been added for linear applications. This is shell type 22. This element can also be used for implicit solutions.
- A null beam element for visualization is available. The keyword to define this null beam is *ELEMENT_PLOTEL. This element is necessary for compatibility with NASTRAN.
- A scalar node can be defined for spring-mass systems. The keyword to define this node is *NODE_SCALAR. This node can have from 1 to 6 scalar degrees-of-freedom.
- A thermal shell has been added for through-thickness heat conduction. Internally, 8 additional nodes are created, four above and four below the mid-surface of the shell element. A quadratic temperature field is modeled through the shell thickness. Internally, the thermal shell is a 12 node solid element.
- A beam OFFSET option is available for the *ELEMENT_BEAM definition to permit the beam to be offset from its defining nodal points. This has the advantage that all beam formulations can now be used as shell stiffeners.
- A beam ORIENTATION option for orienting the beams by a vector instead of the third node is available in the *ELEMENT_BEAM definition for NASTRAN compatibility.
- Non-structural mass has been added to beam elements for modeling trim mass and for NASTRAN compatibility.
- An optional checking of shell elements to avoid abnormal terminations is available. See *CONTROL_SHELL. If this option is active, every shell is checked each time step to see if the distortion is so large that the element will invert, which will result in an abnormal termination. If a bad shell is detected, either the shell will be deleted or the calculation will terminate. The latter is controlled by the input.
- An offset option is added to the inertia definition. See *ELEMENT_INERTIA_-OFFSET keyword. This allows the inertia tensor to be offset from the nodal point.
- Plastic strain and thickness initialization is added to the draw bead contact option. See *CONTACT_DRAWBEAD_INITIALIZE.
- Tied contact with offsets based on both constraint equations and beam elements for solid elements and shell elements that have 3 and 6 degrees-of-freedom per node, respectively. See BEAM_OFFSET and CONSTRAINED_OFFSET contact options. These options will not cause problems for rigid body motions.
- The segment-based (SOFT = 2) contact is implemented for MPP calculations. This enables airbags to be easily deployed on the MPP version.
- Improvements are made to segment-based contact for edge-to-edge and sliding conditions, and for contact conditions involving warped segments.

- An improved interior contact has been implemented to handle large shear deformations in the solid elements. A special interior contact algorithm is available for tetrahedron elements.
- Coupling with MADYMO 6.0 uses an extended coupling that allows users to link most MADYMO geometric entities with LS-DYNA FEM simulations. In this coupling MADYMO contact algorithms are used to calculate interface forces between the two models.
- Release flags for degrees-of-freedom for nodal points within nodal rigid bodies are available. This makes the nodal rigid body option nearly compatible with the RBE2 option in NASTRAN.
- Fast updates of rigid bodies for metalforming applications can now be accomplished by ignoring the rotational degrees-of-freedom in the rigid bodies that are typically inactive during sheet metal stamping simulations. See the keyword: *CONTROL_RIGID.
- Center of mass constraints can be imposed on nodal rigid bodies with the SPC option in either a local or a global coordinate system.
- Joint failure based on resultant forces and moments can now be used to simulate the failure of joints.
- CONSTRAINED_JOINT_STIFFNESS now has a TRANSLATIONAL option for the translational and cylindrical joints.
- Joint friction has been added using table look-up so that the frictional moment can now be a function of the resultant translational force.
- The nodal constraint options *CONSTRAINED_INTERPOLATION and *CONSTRAINED_LINEAR now have a local option to allow these constraints to be applied in a local coordinate system.
- Mesh coarsening can now be applied to automotive crash models at the beginning of an analysis to reduce computation times. See the new keyword: *CONTROL_COARSEN.
- Force versus time seatbelt pretensioner option has been added.
- Both static and dynamic coefficients of friction are available for seat belt slip rings. Previously, only one friction constant could be defined.
- *MAT_SPOTWELD now includes a new failure model with rate effects as well as additional failure options.
- Constitutive models added for the discrete beam elements:
    - *MAT_1DOF_GENERALIZED_SPRING
    - *MAT_GENERAL_NONLINEAR_6dof_DISCRETE_BEAM
    - *MAT_GENERAL_NONLINEAR_1dof_DISCRETE_BEAM
    - *MAT_GENERAL_SPRING_DISCRETE_BEAM

  - *MAT_GENERAL_JOINT_DISCRETE_BEAM
  - *MAT_SEISMIC_ISOLATOR

- for shell and solid elements:

  - *MAT_plasticity_with_damage_ortho
  - *MAT_simplified_johnson_cook_orthotropic_damage
  - *MAT_HILL_3R
  - *MAT_GURSON_RCDC

- for the solid elements:

  - *MAT_SPOTWELD
  - *MAT_HILL_FOAM
  - *MAT_WOOD
  - *MAT_VISCOELASTIC_HILL_FOAM
  - *MAT_LOW_DENSITY_SYNTHETIC_FOAM
  - *MAT_RATE_SENSITIVE_POLYMER
  - *MAT_QUASILINEAR VISCOELASTIC
  - *MAT_TRANSVERSELY_ANISOTROPIC_CRUSHABLE_FOAM
  - *MAT_VACUUM
  - *MAT_MODIFIED_CRUSHABLE_FOAM
  - *MAT_PITZER_CRUSHABLE FOAM
  - *MAT_JOINTED_ROCK
  - *MAT_SIMPLIFIED_RUBBER
  - *MAT_FHWA_SOIL
  - *MAT_SCHWER_MURRAY_CAP_MODEL

- Failure time added to MAT_EROSION for solid elements.
- Damping in the material models *MAT_LOW_DENSITY_FOAM and *MAT_LOW_DENSITY_VISCOUS_FOAM can now be a tabulated function of the smallest stretch ratio.
- The material model *MAT_PLASTICITY_WITH_DAMAGE allows the table definitions for strain rate.
- Improvements in the option *INCLUDE_STAMPED_PART now allow all history data to be mapped to the crash part from the stamped part. Also, symmetry planes can be used to allow the use of a single stamping to initialize symmetric parts.
- Extensive improvements in trimming result in much better elements after the trimming is completed. Also, trimming can be defined in either a local or global coordinate system. This is a new option in *DEFINE_CURVE_TRIM.
- An option to move parts close before solving the contact problem is available, see *CONTACT_AUTO_MOVE.

- An option to add or remove discrete beams during a calculation is available with the new keyword: *PART_SENSOR.
- Multiple jetting is now available for the Hybrid and Chemkin airbag inflator models.
- Nearly all constraint types are now handled for implicit solutions.
- Calculation of constraint and attachment modes can be easily done by using the option: *CONTROL_IMPLICIT_MODES.
- Penalty option, see *CONTROL_CONTACT, now applies to all *RIGIDWALL options and is always used when solving implicit problems.
- Solid elements types 3 and 4, the 4 and 8 node elements with 6 degrees-of-freedom per node are available for implicit solutions.
- The warping stiffness option for the Belytschko-Tsay shell is implemented for implicit solutions. The Belytschko-Wong-Chang shell element is now available for implicit applications. The full projection method is implemented due to it accuracy over the drill projection.
- Rigid to deformable switching is implemented for implicit solutions.
- Automatic switching can be used to switch between implicit and explicit calculations. See the keyword: *CONTROL_IMPLICIT_GENERAL.
- Implicit dynamics rigid bodies are now implemented. See the keyword *CONTROL_IMPLICIT_DYNAMIC.
- Eigenvalue solutions can be intermittently calculated during a transient analysis.
- A linear buckling option is implemented. See the new control input: *CONTROL_IMPLICIT_BUCKLE
- Implicit initialization can be used instead of dynamic relaxation. See the keyword *CONTROL_DYNAMIC_RELAXATION where the parameter, IDFLG, is set to 5.
- Superelements, i.e., *ELEMENT_DIRECT_MATRIX_INPUT, are now available for implicit applications.
- There is an extension of the option, *BOUNDARY_CYCLIC, to symmetry planes in the global Cartesian system. Also, automatic sorting of nodes on symmetry planes is now done by LS-DYNA.
- Modeling of wheel-rail contact for railway applications is now available, see *RAIL_TRACK and *RAIL_TRAIN.
- A new, reduced CPU, element formulation is available for vibration studies when elements are aligned with the global coordinate system. See *SECTION_SOLID and *SECTION_SHELL formulation 98.
- An option to provide approximately constant damping over a range of frequencies is implemented, see *DAMPING_FREQUENCY_RANGE.

Features added during 2003-2005 for the 971 release of LS-DYNA:

Initially, the intent was to quickly release version 971 after 970 with the implicit capabilities fully functional for distributed memory processing using MPI. Unfortunately, the effort required for parallel implicit was grossly underestimated, and, as a result, the release has been delayed. Because of the delay, version 971 has turned into a major release. Some of the new features, listed below, were also added to later releases of version 970. The new explicit capabilities are implemented in the MPP version and except for one case, in the SMP version as well.

Below is list of new capabilities and features:

- A simplified method for using the ALE capability with airbags is now available with the keyword *AIRBAG_ALE.
- Case control using the *CASE keyword, which provides a way of running multiple load cases sequentially within a single run
- New option to forming contact: *CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_SMOOTH, which use fitted surface in contact calculation.
- Butt weld definition by using the *CONSTRAINED_BUTT_WELD option which makes the definition of butt welds simple relative to the option: *CONSTRAINED_GENERALIZED_WELD_BUTT.
- H-adaptive fusion is now possible as an option with the control input, *CONTROL_ADAPTIVE.
- Added a parameter on, *CONTROL_ADAPTIVE, to specify the number of elements generated around a 90 degree radius. A new option to better calculate the curvature was also implemented.
- Added a new keyword: *CONTROL_ADAPTIVE_CURVE, to refine the element along trimming curves
- Birth and death times for implicit dynamics on the keyword *CONTROL_IMPLICIT_DYNAMICS.
- Added an option to scale the spot weld failure resultants to account for the location of the weld on the segment surface, see *CONTROL_SPOTWELD_BEAM.
- Added an option which automatically replaces a single beam spot weld by an assembly of solid elements using the same ID as the beam that was replaced, see *CONTROL_SPOTWELD_BEAM.
- Boundary constraint in a local coordinate system using *CONSTRAINED_LOCAL keyword.
- A cubic spline interpolation element is now available, *CONSTRAINED_SPLINE.

- Static implicit analyses in of a structure with rigid body modes is possible using the option, *CONTROL_IMPLICIT_INERTIA_RELIEF.
- Shell element thickness updates can now be limited to part ID's within a specified set ID, see the *CONTROL_SHELL keyword. The thickness update for shells can now be optionally limited to the plastic part of the strain tensor for better stability in crash analysis.
- Solid element stresses in spot welds are optionally output in the local system using the SWLOCL parameter on the *CONTROL_SOLID keyword.
- SPOTHIN option on the *CONTROL_CONTACT keyword cards locally thins the spot welded parts to prevent premature breakage of the weld by the contact treatments.
- New function: *CONTROL_FORMING_PROJECT, which can initial move the penetrating slave nodes to the master surface
- New function *CONTROL_FORMING_TEMPLATE, which allows user to easily set up input deck. Its function includes auto-position, define travel curve, termination time, and most of the forming parameters for most of the typical forming process.
- New function *CONTROL_FORMING_USER, *CONTROL_FORMING_POSITION, and *CONTROL_FORMING_TRAVEL, when used together, can allow the user to define atypical forming process.
- Added new contact type *CONTACT_GUIDED_CABLE.
- Circular cut planes are available for *DATABASE_CROSS_SECTION definitions.
- New binary database FSIFOR for fluid structure coupling.
- Added *DATABASE_BINARY_D3PROP for writing the material and property data to the first D3PLOT file or to a new database D3PROP.
- DATABASE_EXTENT_BINARY has new flags to output peak pressure, surface energy density, nodal mass increase from mass scaling, thermal fluxes, and temperatures at the outer surfaces of the thermal shell.
- Eight-character alphanumeric labels can now be used for the parameters SECID, MID, EOSID, HGID, and TMID on the *PART keyword.
- Two NODOUT files are now written: one for high frequency output and a second for low frequency output.
- Nodal mass scaling information can now be optionally written to the D3PLOT file.
- Added option, MASS_PROPERTIES, to include the mass and inertial properties in the GLSTAT and SSSTAT files.
- Added option in *CONTROL_CPU to output the cpu and elapsed time into the GLSTAT file.

- Added an option, IERODE, on the *CONTROL_OUTPUT keyword to include eroded energies by part ID into the MATSUM file. Lumped mass kinetic energy is also in the MATSUM file as part ID 0.
- Added an option, TET10, on the *CONTROL_OUTPUT keyword to output ten connectivity nodes into D3PLOT database rather than 4.
- New keyword, *ELEMENT_SOLID_T4TOT10 to convert 4 node tetrahedron elements to 10 node tetrahedron elements.
- New keyword, *ELEMENT_MASS_PART defines the total additional non-structural mass to be distributed by an area weighted distribution to all nodes of a given part ID.
- New keyword option, SET, for *INTIAL_STRESS_SHELL_SET allows a set of shells to be initialized with the state of stress.
- New option allows the number of cpu's to be specified on the *KEYWORD input.
- Tubular drawbead box option for defining the elements that are included in the drawbead contact, see *DEFINE_BOX_DRAWBEAD.
- New function: *DEFINE_CURVE_DRAWBEAD, allow user to conveniently define drawbead by using curves (in x, y format or iges format)
- New function: *DEFINE_DRAWBEAD_BEAM, which allows user to conveniently define drawbead by using beam part ID, and specify the drawbead force.
- Analytic function can be used in place of load curves with the option *DEFINE_-CURVE_FUNCTION.
- Friction can now be defined between part pair using the *DEFINE_FRICTION input.
- New keyword: *DEFINE_CURVE_TRIM_3D, to allow trimming happens based on blank element normal, rather than use pre-defined direction
- A new trimming algorithm was added: *DEFINE_CURVE_TRIM_NEW, which allow seed node to be input and is much faster then the original algorithm.
- A new keyword, *DEFINE_HEX_SPOTWELD_ASSEMBLY, is available to define a cluster of solid elements that comprise a single spot weld.
- The definition of a vector, see *DEFINE_VECTOR, can be done by defining coordinates in a local coordinate system.
- The definition of a failure criteria between part pairs is possible with a table defined using the keyword, *DEFINE_SPOTWELD_FAILURE_RESULTANTS.
- A new keyword, *DEFINE_CONNECTION_PROPERTIES is available for defining failure properties of spot welds.
- Added *DEFINE_SET_ADAPTIVE to allow the adaptive level and element size to be specified by part ID or element set ID.

- Static rupture stresses for beam type spot welds can be defined in the keyword input, *DEFINE_SPOTWELD_RUPTURE_STRESS.
- Section properties can be define in the *ELEMENT_BEAM definitions for resultant beam elements using the SECTION option.
- Physical offsets of the shell reference surface can be specified on the shell element cards, see the OFFSET option on *ELEMENT_SHELL.
- File names can be located in remote directories and accessed through the *INCLUDE_PART keyword.
- New features to *INCLUDE_STAMPED_PART: two different mirror options, user-defined searching radius.
- *INTIAL_STRESS_SECTION allows for stress initialization across a cross-section, which consists of solid elements.
- An option, IVATN, is available for setting the velocities of slaved nodes and parts for keyword, *INITIAL_VELOCITY_GENERATION.
- Twenty-two built-in cross-section are now available in the definition of beam integration rules, see *INTEGRATION_BEAM.
- The possibility of changing material types is now available for shells using the user defined integration rule, see *INTEGRATION_SHELL.
- The interface springback file created by using the keyword, *INTERFACE_-SPRINGBACK is now optionally written as a binary file.
- An optional input line for *KEYWORD allows the definition of a prefix for all file names created during a simulation. This allows multiple jobs to be executed in the same directory.
- Body force loads can now be applied in a local coordinate system for *LOAD_-BODY.
- A pressure loading feature allows moving pressures to be applied to a surface to simulate spraying a surface with stream of fluid through a nozzle. See keyword *LOAD_MOVING_PRESSURE.
- Thermal expansion can be added to any material by the keyword, *MAT_ADD_-THERMAL_EXPANSION.
- Curves can now be used instead of eight digitized data points in the material model *MAT_ELASTIC_WITH_VISCOSITY_CURVE
- New options for spot weld failure in *MAT_SPOTWELD, which apply to beam and solid elements.
- Failure criteria based on plastic strain to failure is added to material *MAT_-ANISOTROPIC_VISCOPLASTIC.
- Strain rate failure criterion is added to material *MAT_MODIFIED_PIECEWISE_LINEAR_PLASTICITY.

- Strain rate scaling of the yield stress can now be done differently in tension and compression in material with separate pressure cut-offs in tension and compression in material model *MAT_PLASTICITY_TENSION_COMPRESSION.
- The RCDC model is now available to predict failure in material *MAT_PLASTICITY_WITH_DAMAGE.
- Two additional yield surfaces have been added to material *MAT_MODIFIED_HONEYCOMB to provide more accurate predictions of the behavior of honeycomb barrier models.
- Unique coordinate systems can be assigned to the two nodal points of material *MAT_1DOF_GENERALIZED_SPRING.
- Poisson's ratio effects are available in foam defined by load curves in the material *MAT_SIMPLIFIED_RUBBER/FOAM
- Failure effects are available in the rubber/foam material defined by load curves in the *MAT_SIMPLIFIED_RUBBER/FOAM_WITH_FAILURE.
- The material option *MAT_ADD_EROSION now allows the maximum pressure at failure and the minimum principal strain at failure to be specified.
- Strains rather than displacements can now be used with the material model for discrete beams, *MAT_GENERAL_NONLINEAR_6DOF_DISCRETE_BEAM.
- New option for *MAT_TRANSVERSELY_ANISOTROPIC_ELASTIC_PLASTIC_(ECHANGE), which allow two ways to change the Young's modulus during forming simulation.
- New Material model: *MAT_HILL_3R: includes the shear term in the yield surface calculation by using Hill's 1948 an-isotropic material model.
- New Material model: *MAT_KINEMATIC_HARDENING_TRANSVERSELY_ANISOTROPIC: which integrates Mat #37 with Yoshida's two-surface kinematic hardening model.
- Improved formulation for the fabric material, *MAT_FABRIC for formulations 2, 3, and 4. The improved formulations are types 12, 13, and 14.
- Constitutive models added for truss elements:
  - *MAT_MUSCLE
- For beam elements
  - *MAT_MOMENT_CURVATURE
- For shell elements
  - *MAT_RESULTANT_ANISOTROPIC
  - *MAT_RATE_SENSITIVE_COMPOSITE_FABRIC.
  - *MAT_SAMP-1

  - *MAT_SHAPE_MEMORY is now implemented for shells.

- for shell and solid elements:

  - *MAT_BARLAT_YLD2000 for anisotropic aluminum alloys.
  - *MAT_SIMPLIFIED_RUBBER_WITH_DAMAGE
  - *MAT_VISCOELASTIC_THERMAL
  - *MAT_THERMO_ELASTO_VISCOPLASTIC_CREEP

- for the solid elements:

  - *MAT_ARUP_ADHESIVE
  - *MAT_BRAIN_LINEAR_VISCOELASTIC.
  - *MAT_CSCM for modeling concrete.
  - *MAT_PLASTICITY_COMPRESSION_TENSION_EOS for modeling ice.
  - *MAT_COHESIVE_ELASTIC
  - *MAT_COHESIVE_TH
  - *MAT_COHESIVE_GENERAL
  - *MAT_EOS_GASKET
  - *MAT_SIMPLIFIED_JOHNSON_COOK is now implemented for solids.
  - *MAT_PLASTICITY_WITH_DAMAGE is now implemented for solids.
  - *MAT_SPOTWELD_DAIMLERCHRYSLER

- User defined equations-of-state are now available.

- There is now an interface with the MOLDFLOW code.

- Damping defined in *DAMPING_PART_STIFFNESS now works for the Belytschko –Schwer beam element.

- The option *NODE_TRANSFORMATION allows a node set to be transformed based on a transformation defined in *DEFINE_TRANSFORMATION.

- Parameters can be defined in FORTRAN like expressions using *PARAMETER_EXPRESSION.

- A part can be moved in a local coordinate system in *PART_MOVE.

- A simplified method for defining composite layups is available with *PART_COMPOSITE

- The rigid body inertia can be changed in restart via *CHANGE_RIGID_BODY_INERTIA.

- A part set can now be defined by combining other part sets in *SET_PART_ADD.

- Termination of the calculation is now possible if a specified number of shell elements are deleted in a give part ID. See *TERMINATION_DELETED_SHELLS.

- Added hourglass control type 7 for solid elements for use when modeling hyperelastic materials.
- Shell formulations 4, 11, 16, and 17 can now model rubber materials.
- Added a new seatbelt pretensioner type 7 in which the pretensioner and retractor forces are calculated independently and added.
- A new composite tetrahedron element made up from 12 tetrahedron is now available as solid element type 17.
- Shell thickness offsets for *SECTION_SHELL now works for most shell elements, not just the Hughes-Liu shell.
- The Hughes-Liu beam has been extended to include warpage for open cross-sections.
- A resultant beam formulation with warpage is available as beam type 12.
- Two nonlinear shell elements are available with 8 degrees-of-freedom per node to include thickness stretch.
- Tetrahedron type 13, which uses nodal pressures, is now implemented for implicit applications.
- Cohesive solid elements are now available for treating failure.
- Seatbelt shell elements are available for use with the all seatbelt capabilities.
- Superelements can now share degrees-of-freedom and are implemented for implicit applications under MPI.
- A user defined element interface is available for solid and shell elements.
- Thermal shells are available for treating heat flow through shell elements.
- EFG shell formulations 41 and 42 are implemented for explicit analysis.
- EFGPACK is implemented in addition to BCSLIB-EXT solver on the keyword *CONTROL_EFG.
- EFG MPP version is available for explicit analysis.
- EFG fast transformation method is implemented in the EFG solid formulation.
- EFG Semi-Lagrangian kernel and Eulerian kernel options are added for the foam materials.
- EFG 3D adaptivity is implemented for the metal materials.
- EFG E.O.S. and *MAT_ELASTIC_FLUID materials are included in the 4-noded background element formulation.
- Airbag simulations by using ALE method can be switched to control volume method by *ALE_CV_SWITCH.
- *MAT_ALE_VISCOUS now supports Non-Newtonian viscosity by power law or load curve.

- *DATABASE_BINARY_FSIFOR outputs fluid-structure interaction data to binary file.
- *DATABASE_FSI_SENSOR outputs ALE element pressure to ASCII file dbsor.
- *MAT_GAS_MIXTURE supports nonlinear heat capacities.
- *INITIAL_VOLUME_FRACTION_GEOMETRY uses an enhanced algorithm to handle both concave and convex geometries and substantially reduce run time.
- A new keyword *DELETE_FSI allows the deletion of coupling definitions.
- Convection heat transfer activates by *LOAD_ALE_CONVECTION in ALE FSI analysis.
- *ALE_FSI_SWITCH_MMG is implemented to switch between ALE multi-material groups to treat immersed FSI problems.
- Type 9 option is added in *ALE_REFERENCE_SYSTEM_GROUP to deal complex ALE mesh motions including translation, rotation, expansion and contraction, etc.

    - New options in *CONSTRAINED_LAGRANGE_IN_SOLID
    - Shell thickness option for coupling type 4.
    - Bulk modulus based coupling stiffness.
    - Shell erosion treatment.
    - Enable/disable interface force file.

- New coupling method for fluid flowing through porous media are implemented as type 11 (shell) and type 12 (solid) in *CONSTRAINED_LAGRANGE_IN_SOLID.
- *ALE_MODIFIED_STRAIN allows multiple strain fields in certain ALE elements to solve sticking behavior in FSI. (MPP underdevelopment)
- *ALE_FSI_PROJECTION is added as a new constraint coupling method to solve small pressure variation problem. (MPP underdevelopment)
- *BOUNDARY_PRESCRIBED_ORIENTATION_RIGID is added as a means to prescribe as a function of time the general orientation of a rigid body using a variety of methods. This feature is available in release R3 and higher of Version 971.
- *BOUNDARY_PRESCRIBED_ACCELEROMETER_RIGID is added as a means to prescribe the motion of a rigid body based un experimental data gathered from accelerometers affixed to the rigid body. This feature is available in release R3 and higher of Version 971.

Capabilities added during 2008-2011 for Version 971R6 of LS-DYNA:

During the last four years the implicit capabilities are now scalable to a large number of cores; therefore, LS-DYNA has achieved a major goal over 15 years of embedding a scalable implicit solver. Also, in addition to the progress made for implicit solutions many other new and useful capabilities are now available.

- The keyword *ALE_AMBIENT_HYDROSTATIC initializes the hydrostatic pressure field in the ambient ALE domain due to an acceleration like gravity.
- The keyword *ALE_FAIL_SWITCH_MMG allows switching an ALE multi-material-group ID (AMMGID) if the material failure criteria occurs.
- The keyword *ALE_FRAGMENTATION allow switching from the ALE multi-material-group ID, AMMGID, (FR_MMG) of this failed material to another AMMGID (TO_MMG). This feature may typically be used in simulating fragmentation of materials.
- The keyword *ALE_REFINE refines ALE hexahedral solid elements automatically.
- The keyword *BOUNDARY_ALE_MAPPING maps ALE data histories from a previous run to a region of elements. Data are read from or written to a mapping file with a file name given by the prompt “map=” on the command line starting the execution.
- The keyword *BOUNDARY_PORE_FLUID is used to define parts that contain pore fluid where defaults are given on *CONTROL_PORE_FLUID input.
- With the keyword, *BOUNDARY_PRESCRIBED_FINAL_GEOMETRY, the final displaced geometry for a subset of nodal points is defined. The nodes of this subset are displaced from their initial positions specified in the *NODE input to the final geometry along a straight line trajectory. A load curve defines a scale factor as a function of time that is bounded between zero and unity corresponding to the initial and final geometry, respectively. A unique load curve can be specified for each node, or a default load curve can apply to all nodes.
- The keyword, *BOUNDARY_PWP, defines pressure boundary conditions for pore water at the surface of the software.
- The keyword, *CONSTRAINED_JOINT_COOR, defines a joint between two rigid bodies. The connection coordinates are given instead of the nodal point IDs used in *CONSTRAINED_JOINT.
- The keyword, *CONSTRAINED_SPR2, defines a self-piercing rivet with failure. This model for a self-piercing rivet (SPR2) includes a plastic-like damage model that reduces the force and moment resultants to zero as the rivet fails. The domain of influence is specified by a diameter, which should be approximately equal to the rivet’s diameter. The location of the rivet is defined by a single node at the center of two riveted sheets.

- Through the keyword, *CONTROL_BULK_VISCOSITY, bulk viscosity is optional for the Hughes-Liu beam and beam type 11 with warpage. This option often provides better stability, especially in elastic response problems.
- The display of nodal rigid bodies is activated by the parameter, PLOTEL, on the *CONTROL_RIGID keyword.
- The mortar contact, invoked by appending the suffix MORTAR to either FORMING_SURFACE_TO_SURFACE, AUTOMATIC_SURFACE_TO_SURFACE or AUTOMATIC_SINGLE_SURFACE, is a segment to segment penalty based contact. For two segments on each side of the contact interface that are overlapping and penetrating, a consistent nodal force assembly taking into account the individual shape functions of the segments is performed. In this respect the results with this contact may be more accurate, especially when considering contact with elements of higher order. By appending the suffix TIED to the CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_MORTAR keyword or the suffix MORTAR to the CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_TIEBREAK keyword, this is treated as a tied contact interface with tiebreak failure in the latter case. Only OPTION = 9 is supported for the mortar tiebreak contact. The mortar contact is intended for implicit analysis in particular but is nevertheless supported for explicit analysis as well.
- In the database, ELOUT, the number of history variables can be specified for output each integration point in the solid, shell, thick shell, and beam elements. The number of variables is given on the *DATABASE_ELOUT keyword definition.
- A new option is available in *DATABASE_EXTENT_BINARY. Until now only one set of integration points were output through the shell thickness. The lamina stresses and history variables were averaged for fully integrated shell elements, which results in less disk space for the D3PLOT family of files, but makes it difficult to verify the accuracy of the stress calculation after averaging. An option is now available to output all integration point stresses in fully integrated shell elements: 4 x # of through thickness integration points in shell types 6, 7, 16, 18-21, and 3 x # of through thickness integration points in triangular shell types 3, and 17.
- The keyword *DATABASE_PROFILE allows plotting the distribution or profile of data along x, y, or z-direction.
- The purpose of the keyword, *DEFINE_ADAPTIVE_SOLID_TO_SPH, is to adaptively transform a Lagrangian solid Part or Part Set to SPH particles when the Lagrange solid elements comprising those parts fail. One or more SPH particles (elements) will be generated for each failed element to. The SPH particles replacing the failed element inherit all of the properties of failed solid element, e.g. mass, kinematic variables, and constitutive properties.

- With the keywords beginning with, *DEFINE_BOX, a LOCAL option is now available. With this option the diagonal corner coordinates are given in a local coordinate system defined by an origin and vector pair.
- The keyword, *DEFINE_CURVE_DUPLICATE, defines a curve by optionally scaling and offsetting the abscissa and ordinates of another curve defined by the *DEFINE_CURVE keyword.
- The keyword, *DEFINE_ELEMENT_DEATH, is available to delete a single element or an element set at a specified time during the calculation.
- The purpose of the keyword, *DEFINE_FRICTION_ORIENTATION, is to allow for the definition of different coefficients of friction (COF) in specific directions, specified using a vector and angles in degrees. In addition, COF can be scaled according to the amount of pressure generated in the contact interface.
- With the new keyword, *DEFINE_FUNCTION, an arithmetic expression involving a combination of independent variables and other functions, i.e.,

  ```
  f(a,b,c) = a*2 + b*c + sqrt(a*c)
  ```

  is defined where a, b, and c are the independent variables. This option is implemented for a subset of keywords.

  - *ELEMENT_SEATBELT_SLIPRING
  - *LOAD_BEAM
  - *LOAD_MOTION_NODE
  - *LOAD_MOVING_PRESSURE
  - *LOAD_NODE
  - *LOAD_SEGMENT
  - *LOAD_SEGMENT_NONUNIFORM
  - *LOAD_SETMENT_SET_NONUNIFORM
  - *BOUNDARY_PRESCRIBED_MOTION

- If a curve ID is not found, then the function ID's are checked.
- The keyword, *DEFINE_SPH_TO_SPH_COUPLING, defines a penalty based contact to be used for the node to node contacts between SPH parts.
- The keyword, *DEFINE_TABLE_2D, permits the same curve ID to be referenced by multiple tables, and the curves may be defined anywhere in the input.
- The keyword, *DEFINE_TABLE_3D, provides a way of defining a three-dimensional table. A 2D table ID is specified for each abscissa value defined for the 3D table.
- The keyword, *ELEMENT_BEAM_PULLEY, allows the definition of a pulley for truss beam elements (see *SECTION_BEAM, ELFORM = 3). Currently, the beam pulley is implemented for *MAT_001 and *MAT_156. Pulleys allow continuous sliding of a string of truss beam element through a sharp change of angle.

- The purpose of the keyword, *ELEMENT_MASS_MATRIX, is to define a 6x6 symmetric nodal mass matrix assigned to a nodal point or each node within a node set.
- The keyword, *ELEMENT_DISCRETE_SPHERE, allows the definition of a discrete spherical element for discrete element calculations. Each particle consists of a single node with its mass, mass moment of inertia, and radius. Initial coordinates and velocities are specified via the nodal data.
- The two keywords, *ELEMENT_SHELL_COMPOSITE and *ELEMENT_-TSHELL_COMPOSITE, are used to define elements for a general composite shell part where the shells within the part can have an arbitrary number of layers. The material ID, thickness, and material angle are specified for the thickness integration points for each shell in the part
- The keyword, *EOS_USER_DEFINED, allows a user to supply their own equation-of-state subroutine.
- The new keyword *FREQUENCY_DOMAIN provides a way of defining and solving frequency domain vibration and acoustic problems. The related keyword cards given in alphabetical order are:
    - *FREQUENCY_DOMAIN_ACOUSTIC_BEM_{*OPTION*}
    - *FREQUENCY_DOMAIN_ACOUSTIC_FEM
    - *FREQUENCY_DOMAIN_FRF
    - *FREQUENCY_DOMAIN_RANDOM_VIBRATION
    - *FREQUENCY_DOMAIN_RESPONSE_SPECTRUM
    - *FREQUENCY_DOMAIN_SSD
- The keyword, *INITIAL_AIRBAG_PARTICLE, initializes pressure in a closed airbag volume, door cavities for pressure sensing studies, and tires.
- The keyword *INITIAL_ALE_HYDROSTATIC initializes the hydrostatic pressure field in an ALE domain due to an acceleration like gravity.
- The keyword *INITIAL_ALE_MAPPING maps ALE data histories from a previous run. Data are read from a mapping file with a file name given by the prompt “map=” on the command line starting the execution.
- The keyword, *INITIAL_AXIAL_FORCE_BEAM, provides a simplified method to model initial tensile forces in bolts.
- The keyword, *INITIAL_FIELD_SOLID, is a simplified version of the *INITIAL_-STRESS_SOLID keyword which can be used with hyperelastic materials. This keyword is used for history variable input. Data is usually in the form of the eigenvalues of diffusion tensor data. These are expressed in the global coordinate system.
- The equation-of-state, *EOS_MIE_GRUNEISEN, type 16, is a Mie-Gruneisen form with a p-α compaction model.

- The keyword, *LOAD_BLAST_ENHANCED, defines an air blast function for the application of pressure loads due the explosion of conventional charge. While similar to *LOAD_BLAST this feature includes enhancements for treating reflected waves, moving warheads and multiple blast sources. The loads are applied to facets defined with the keyword *LOAD_BLAST_SEGMENT. A database containing blast pressure history is also available (see *DATABASE_BINARY_BLSTFOR).
- The keyword, *LOAD_ERODING_PART_SET, creates pressure loads on the exposed surface composed of solid elements that erode, i.e., pressure loads are added to newly exposed surface segments as solid elements erode.
- The keyword, *LOAD_SEGMENT_SET_ANGLE, applies traction loads over a segment set that is dependent on the orientation of a vector. An example application is applying a pressure to a cylinder as a function of the crank angle in an automobile engine
- The keyword, *LOAD_STEADY_STATE_ROLLING, is a generalization of *LOAD_BODY, allowing the user to apply body loads to part sets due to translational and rotational accelerations in a manner that is more general than the *LOAD_BODY capability. The *LOAD_STEADY_STATE_ROLLING keyword may be invoked an arbitrary number of times in the problem as long as no part has the option applied more than once and they can be applied to arbitrary meshes. This option is frequently used to initialize stresses in tire.
- The keywords INTERFACE_SSI, INTERFACE_SSI_AUX, INTERFACE_SSI_AUX_EMBEDDED and INTERFACE_SSI_STATIC are used to define the soil-structure interface appropriately in various stages of soil-structure interaction analysis under earthquake ground motion.
- The keyword, *LOAD_SEISMIC_SSI, is used to apply earthquake loads due to free-field earthquake ground motion at certain locations — defined by either nodes or coordinates — on a soil-structure interface. This loading is used in earthquake soil-structure interaction analysis. The specified motions are used to compute a set of effective forces in the soil elements adjacent to the soil-structure interface, according to the effective seismic input–domain reduction method.
- The keyword *DEFINE_GROUND_MOTION is used to specify a ground motion to be used in conjunction with *LOAD_SEISMIC_SSI.
- Material types *MAT_005 and *MAT_057 now accept table input to allow the stress quantity versus the strain measure to be defined as a function of temperature.
- The material option *MAT_ADD_EROSION, can now be applied to all nonlinear shell, thick shell, fully integrated solids, and 2D solids. New failure criteria are available.
- The GISSMO damage model, now available as an option in *MAT_ADD_EROSION, is a phenomenological formulation that allows for an incremental descrip-

tion of damage accumulation, including softening and failure. It is intended to provide a maximum in variability for the description of damage for a variety of metallic materials (e.g. *MAT_024, *MAT_036, …). The input of parameters is based on tabulated data, allowing the user to directly convert test data to numerical input.

- The keyword, *MAT_RIGID_DISCRETE or MAT_220, eliminates the need to define a unique rigid body for each particle when modeling a large number of rigid particles. This gives a large reduction in memory and wall clock time over separate rigid bodies. A single rigid material is defined which contains multiple disjoint pieces. Input is simple and unchanged, since all disjoint rigid pieces are identified automatically during initialization.
- The keyword, *NODE_MERGE, causes nodes with identical coordinates to be replaced during the input phase by the node encountered that has the smallest ID.
- The keyword, *PART_ANNEAL, is used to initialize the stress states at integration points within a specified part to zero at a given time during the calculation. This option is valid for parts that use constitutive models where the stress is incrementally updated. This option also applies to the Hughes-Liu beam elements, the integrated shell elements, thick shell elements, and solid elements.
- The keyword, *PART_DUPLICATE, provides a method of duplicating parts or part sets without the need to use the *INCLUDE_TRANSFORM option.
- To automatically generate elements to visualize rigid walls the DISPLAY option is now available for *RIGIDWALL_PLANAR and *RIGIDWALL_GEOMETRIC.
- A one point integrated pentahedron solid element with hourglass control is implemented as element type 115 and can be referenced in *SECTION_SOLID. Also, the 2 point pentahedron solid, type 15, no longer has a singular mode.
- The keyword *SECTION_ALE1D defines section properties for 1D ALE elements.
- The keyword *SECTION_ALE2D defines section properties for 2D ALE elements.
- The keywords *SET_BEAM_INTERSECT, *SET_SHELL_INTERSECT, *SET_SOLID_INTERSECT, *SET_NODE_INTERSECT, and *SET_SEGMENT_INTER-SECT, allows the definition of a set as the intersection, ∩, of a series of sets. The new set, SID, contains all common members.
- The keyword, *SET_SEGMENT_ADD, is now available for defining a new segment set by combining other segment sets.
- The two keywords, *DEFINE_ELEMENT_GENERALIZED_SHELL and *DEFINE_
ELEMENT_GENERALIZED_SOLID, are used to define general shell and solid element formulations to allow the rapid prototyping of new element formula-

tions. They are used in combination with the new keywords *ELEMENT_GENERLIZED_SHELL and *ELEMENT_GENERALIZED_SOLID.

- The two keywords, *ELEMENT_INTERPOLATION_SHELL and *ELEMENT_INTERPOLATION_SOLID, are used to interpolate stresses and other solution variables from the generalized shell and solid element formulations for visualization. They are used together with the new keyword *CONSTRAINED_NODE_-INTERPOLATION.
- The keyword, *ELEMENT_SHELL_NURBS_PATCH, is used to define 3D shell elements based on NURBS (Non-Uniform Ration B-Spline) basis functions. Currently four different element formulations, with and without rotational degrees of freedom are available.
- The keyword LOAD_SPCFORC is used to apply equivalent SPC loads, read in from the d3dump file during a full-deck restart, in place of the original constraints in order to facilitate the classical non-reflecting boundary on an outside surface.

Capabilities added in 2012 to create Version 97R6.1, of LS-DYNA:

- A new keyword *MAT_THERMAL_DISCRETE_BEAM defines thermal properties for ELFORM 6 beam elements.
- An option *CONTROL_THERMAL_SOLVER, invoked by TSF < 0, gives the thermal speedup factor via a curve. This feature is useful when artificially scaling velocity in metal forming.
- A nonlinear form of Darcy's law in *MAT_ADD_PORE_AIR allows curves to define the relationship between pore air flow velocity and pore air pressure gradient.
- An extention to the PART option in *SET_SEGMENT_GENERAL allows reference to a beam part. This allows for creation of 2D segments for traction application.
- Options “SET_SHELL”, “SET_SOLID”, “SET_BEAM”, “SET_TSHELL”, “SET_-SPRING” are added to *SET_NODE_GENERAL so users can define a node set using existing element sets.
- Options “SET_SHELL”, “SET_SOLID”, “SET_SLDIO”, “SET_TSHELL”, “SET_-TSHIO” are added to *SET_SEGMENT_GENERAL so users can use existing element sets to define a segment set.
- *BOUNDARY_PRESCRIBED_MOTION_SET_BOX prescribes motion to nodes that fall inside a defined box.
- IPNINT > 1 in *CONTROL_OUTPUT causes d3hsp to list the IPNINT smallest element timesteps in ascending order.
- Section and material titles are echoed to d3hsp.

- A new parameter MOARFL in *DEFINE_CONNECTION_PROPERTIES permits reduction in modeled area due to shear.
- A new option HALF_SPACE in *FREQUENCY_DOMAIN_ACOUSTIC_BEM enables treatment of a half-space in boundary element method, frequency domain acoustic analysis.
- A shell script “kill_by_pid” is created during MPP startup. When executed, this script will run “kill -9” on every LS-DYNA process started as part of the MPP job. This is for use at the end of submission scripts, as a “fail safe” cleanup in case the job aborts.
- A new parameter IAVIS in *CONTROL_SPH selects the artificial viscosity formulation for the SPH particles. If set to 0, the Monaghan type artificial viscosity formulation is used. If set to 1, the standard artificial viscosity formulation for solid elements is used which may provide a better energy balance but is less stable in specific applications such as high velocity impact.
- Contact friction may be included in *CONTACT_2D_NODE_TO_SOLID for SPH.
- A new keyword *ALE_COUPLING_NODAL_CONSTRAINT provides a coupling mechanism between ALE solids and non-ALE nodes. The nodes can be from virtually any non-ALE element type including DISCRETE_SPHERE, EFG, and SPH, as well as the standard Lagrangian element types. In many cases, this coupling type may be a better alternative to *CONSTRAINED_LAGRANGE_-IN_SOLID.
- The keyword *ALE_ESSENTIAL_BOUNDARY assigns essential boundary conditions to nodes of the ALE boundary surface. The command can be repeated multiple times and is recommended over use of EBC in *CONTROL_ALE..
- The keyword *DELETE_ALECPL in a small restart deck deletes coupling defined with *ALE_COUPLING_NODAL_CONSTRAINT. The command can also be used to reinstate the coupling in a later restart.
- *DEFINE_VECTOR_NODES defines a vector with two node points.
- *CONTACT_AUTOMATIC_SINGLE_SURFACE_TIED allows for the calculation of eigenvalues and eigenvectors for models that include *CONTACT_AUTOMATIC_SINGLE_SURFACE.
- A new parameter RBSMS in *CONTROL_RIGID affects rigid body treatment in Selective Mass Scaling (*CONTROL_TIMESTEP). When rigid bodies are in any manner connected to deformable elements, RBSMS = 0 (default) results in spurious inertia due to improper treatment of the nodes at the interface. RBSMS = 1 alleviates this effect but an additional cost is incurred.
- A new parameter T10JTOL in *CONTROL_SOLID sets a tolerance for issuing a warning when J_min/J_max goes below this tolerance value (i.e., quotient between minimum and maximum Jacobian value in the integration points) for

tetrahedron type 16. This quotient serves as an indicator of poor tetrahedral element meshes in implicit that might cause convergence problems.

- A new option MISMATCH for *BOUNDARY_ACOUSTIC_COUPLING handles coupling of structural element faces and acoustic volume elements (ELFORMs 8 and 14) in the case where the coupling surfaces do not have coincident nodes.
- A porosity leakage formulation in *MAT_FABRIC (*MAT_034, FLC < 0) is now available for particle gas airbags (*AIRBAG_PARTICLE).
- *BOUNDARY_PRESCRIBED_ACCELEROMETER is disabled during dynamic relaxation.
- A new parameter CVRPER in *BOUNDARY_PAP defines porosity of a cover material encasing a solid part.
- A parameter TIEDID in *CONTACT_TIED_SURFACE_TO_SURFACE offers an optional incremental normal update in SMP to eliminate spurious contact forces that may appear in some applications.
- A new option SPOTSTP = 3 in *CONTROL_CONTACT retains spot welds even when the spot welds are not found by *CONTACT_SPOTWELD.
- The SMP consistency option (ncpu < 0) now pertains to the ORTHO_FRICTION contact option.
- Forces from *CONTACT_GUIDED_CABLE are now written to ncforc (both ASCII and binout).
- Discrete beam materials 70, 71, 74, 94, 121 calculate axial force based on change in length. Output the change in length instead of zero axial relative displacement to ASCII file disbout (*DATABASE_DISBOUT).
- *DATABASE_RCFORC_MOMENT is now supported in implicit.
- After the first implicit step, the output of projected cpu and wall clock times is written and the termination time is echoed.
- *DATABASE_MASSOUT is upgraded to include a summary table and to optionally add mass for nodes belonging to rigid bodies.
- Generate and store resultant forces for the LaGrange Multiplier joint formulation so as to give correct output to jntforc (*DATABASE_JNTFORC).
- Control the number of messages for deleted and failed elements using parameter MSGMAX in *CONTROL_OUTPUT.
- Nodal and resultant force output is written to nodfor for nodes defined in *DATABASE_NODAL_FORCE_GROUP in *FREQUENCY_DOMAIN_SSD analysis (SMP only).
- Ncforc data is now written for guided cables (*CONTACT_GUIDED_CABLE) in MPP.

- Jobid handling is improved in l2a utility so that binout files from multiple jobs, with or without a jobid-prefix, can be converted with the single command "l2a -j *binout*". The output contains the correct prefix according to the jobid.
- ALE_MULTI-MATERIAL_GROUP (AMMG) info is written to matsum (both ASCII and binout).
- Shell formulation 14 is switched to 15 (*SECTION_SHELL) in models that include axisymmetric SPH.
- *ELEMENT_BEAM_PULLEY is permitted with *MAT_CABLE_DISCRETE_-BEAM.
- A warning during initialization is written if a user creates DKT triangles, either by ELFORM = 17 on *SECTION_SHELL or ESORT = 2 on *CONTROL_SHELL, that are thicker than the maximum edge length.
- Account is taken of degenerate acoustic elements with ELFORM 8. Tria and quad faces at acoustic-structure boundary are handled appropriately according to shape.
- The compression elimination option for 2D seatbelts, CSE = 2 in *MAT_SEAT-BELT is improved.
- Detailed material failure (*MAT_ADD_EROSION) messages in messag and d3hsp are suppressed when number of messages > MSGMAX (*CONTROL_-OUTPUT).
- Implement SMP consistency (ncpu < 0) in *MAT_COHESIVE_GENERAL (*MAT_186) solids and shells.
- Viscoelastic model in *MAT_077_O now allows up to twelve terms in Prony series instead of standard six.
- Large curve ID's for friction table (*CONTACT_… with FS = 2) are enabled.
- Efficiency of GISSMO damage in *MAT_ADD_EROSION is improved.
- *MAT_ADD_PERMEABILITY_ORTHOTROPIC is now available for pore pressure analysis (*…_PORE_FLUID).
- For *MAT_224 solids and shells, material damage serves as the failure variable in *CONSTRAINED_TIED_NODES_FAILURE.
- The behavior of *MAT_ACOUSTIC is modified when used in combination with dynamic relaxation (DR). Acoustic domain now remains unperturbed in the DR phase but hydrostatic pressure from the acoustic domain is applied to the structure during DR.
- Option for 3D to 2D mapping is added in *INITIAL_ALE_MAPPING.
- *CONTACT_ERODING_NODES_TO_SURFACE contact may be used with SPH particles.

- Total Lagrangian SPH formulation 7 (*CONTROL_SPH) is now available in MPP.
- The output formats for linear equation solver statistics now accommodate very large numbers as seen in large models.
- *CONTROL_OUTPUT keyword parameter NPOPT is now applicable to thermal data. If NPOPT = 1, then printing of the following input data to d3hsp is suppressed:
    - *INITIAL_TEMPERATURE
    - *BOUNDARY_TEMPERATURE
    - *BOUNDARY_FLUX
    - *BOUNDARY_CONVECTION
    - *BOUNDARY_RADIATION
    - *BOUNDARY_ENCLOSURE_RADIATION
- Beam energy balance information is written to TPRINT file.
- MPP performance for LS-DYNA/Madymo coupling is improved.
- Shell adaptivity (*CONTROL_ADAPTIVE) is improved to reduce the number of elements along curved surfaces in forming simulations.
- One-step unfolding (*CONTROL_FORMING_ONESTEP) is improved to accommodate blanks with small initial holes.
- Efficiency of FORM 3 isogeometric shells is improved.
- The processing of *SET_xxx_GENERAL is faster.
- *KEYWORD_JOBID now works even when using the *CASE command.
- Parts may be repositioned in a small restart by including *DEFINE_TRANSFORMATION and *NODE_TRANSFORM in the small restart deck to move nodes of a specified node set prior to continuing the simulation.

Capabilities added during 2012/2013 to create LS-DYNA R7.0:

- Three solvers, EM, CESE, and ICFD, and a volume mesher to support the latter two solvers, are new in Version 7. Brief descriptions of those solvers are given below. Keyword commands for the new solvers are in Volume III of the LS-DYNA Keyword User's Manual. These new solvers are only included in double precision executables.
- Keyword family: *EM_, the keywords starting with *EM refer to and control the Electromagnetic solver problem set up:
    - EM Solver Characteristics:
        - Implicit

        - Double precision
        - Dynamic memory handling
        - SMP and MPP
        - 2D axisymmetric solver / 3D solver
        - Automatic coupling with structural and thermal LS-DYNA solvers
        - FEM for conducting pieces only, no air mesh needed (FEM-BEM system)
        - Solid elements for conductors, shells can be insulators

    - EM Solver Main Features:

        - Eddy Current (a.k.a Induction-Diffusion) solver
        - Induced heating solver
        - Resistive heating solver
        - Imposed tension or current circuits
        - Exterior field
        - Magnetic materials (beta version)
        - Electromagnetic contact
        - EM Equation of states (Conductivity as a function of temperature)

    - EM Solver Applications (Non-exhaustive) :

        - Electromagnetic forming
        - Electromagnetic welding
        - Electromagnetic bending
        - Inductive heating
        - Resistive heating
        - Rail-gun
        - Ring expansions

- Keyword family: *CESE_, the keywords starting with *CESE refer to and control the Compressible CFD solver problem set up:

    - CESE Solver Characteristics:

        - Explicit
        - Double precision
        - Dynamic memory handling
        - SMP and MPP
        - 3D solver / special case 2D solver and 2D axisymmetric solver
        - Automatic coupling with structural and thermal LS-DYNA solvers
        - Eulerian fixed mesh or moving mesh (Either type input with *ELEMENT_SOLID cards or using *MESH cards)

    - CESE Solver Main Features:

    - The CESE (Conservation Element / Solution Element) method enforces conservation in space-time
    - Highly accurate shock wave capturing
    - Cavitation model
    - Embedded (immersed) boundary approach or moving (fitting) approach for FSI problems
    - Coupled stochastic fuel spray solver (See *STOCHASTIC keywords)
    - Coupling with chemistry (See *CHEMISTRY keywords) solver

  - CESE Solver Applications (Non-exhaustive) :

    - Shock wave capturing
    - Shock/acoustic wave interaction
    - Cavitating flows
    - Conjugate heat transfer problems
    - Many different kinds of stochastic particle flows, e.g, dust, water, fuel.
    - Chemically reacting flows, e.g, detonating flow, supersonic combustion.

- Keyword family: *ICFD_, the keywords starting with *ICFD refer to and control the incompressible CFD solver problem set up:

  - ICFD Solver Characteristics:

    - Implicit
    - Double precision
    - Dynamic memory handling
    - SMP and MPP
    - 2D solver / 3D solver
    - Makes use of an automatic volume mesh generator for fluid domain (See *MESH keywords)
    - Coupling with structural and thermal LS-DYNA solvers

  - ICFD Solver Main Features:

    - Incompressible fluid solver
    - Thermal solver for fluids
    - Free Surface flows
    - Two-phase flows
    - Turbulence models
    - Transient or steady-state problems
    - Non-Newtonian fluids
    - Boussinesq model for convection
    - Loose or strong coupling for FSI (Fluid-structure interaction)
    - Exact boundary condition imposition for FSI problems

- ICFD Solver Applications (Non-exhaustive) :
  - External aerodynamics for incompressible flows
  - Internal aerodynamics for incompressible flows
  - Sloshing, Slamming and Wave impacts
  - FSI problems
  - Conjugate heat transfer problems

- Keyword family: *MESH_, the keywords starting with *MESH refer to and control the tools for the automatic volume mesh generator for the CESE and ICFD solvers.
  - Mesh Generator Characteristics:
    - Automatic
    - Robust
    - Generic
    - Tetrahedral elements for 3D, Triangles in 2D
    - Closed body fitted mesh (surface mesh) needs to be provided for volume generation
  - Mesh Generator Main Features:
    - Automatic remeshing to keep acceptable mesh quality for FSI problems (ICFD only)
    - Adaptive meshing tools (ICFD only)
    - Anisotropic boundary layer mesh
    - Mesh element size control tools
    - Remeshing tools for surface meshes to ensure mesh quality
  - Mesh Generator Applications :
    - Used by the Incompressible CFD solver (ICFD).
    - Used by the Compressible CFD solver (CESE).

Other additions to Version 7 include:

- Add new parameter VNTOPT to *AIRBAG_HYBRID, that allows user more control on bag venting area calculation.
- Allow heat convection between environment and CPM bag (*AIRBAG_PARTICLE) bag. Apply proper probability density function to part's temperature created by the particle impact.
- *AIRBAG_PARTICLE and *SENSOR_SWITCH_SHELL_TO_VENT allows user to input load curve to control the venting using choking flow equation to get proper probability function for vents. Therefore, this vent will have the same vent rate as real vent hole.

- Add new option NP2P in *CONTROL_CPM to control the repartition frequency of CPM particles among processors (MPP only).
- Enhance *AIRBAG_PARTICLE to support a negative friction factor (FRIC or PFRIC) in particle to fabric contact. Particles are thus able to rebound at a trajectory closer to the fabric surface after contact.
- Use heat convection coefficient HCONV and fabric thermal conductivity KP to get correct effective heat transfer coefficient for heat loss calculation in *AIRBAG_PARTICLE. If KP is not given, H will be used as effective heat transfer coefficient.
- Extend CPM inflator orifice limit from 100 to unlimited (*AIRBAG_PARTICLE).
- Support dm_in_dt and dm_out_dt output to CPM chamber database (*DATABASE_ABSTAT) to allow user to study mass flow rate between multiple chambers.
- Previously, the number of ships (rigid bodies) in *BOUNDARY_MCOL, as specified by NMCOL, was limited to 2. Apparently, this was because the code had not been validated for more than 2 rigid bodies, but it is believed that it should not be a problem to remove this restriction. Consequently, this limit has been raised to 10, with the caveat that the user should verify the results for NMCOL > 2.
- Implemented a structural-acoustic mapping scheme (*BOUNDARY_ACOUSTIC_MAPPING), for mapping transient structural nodal velocity to acoustic volume surface nodes. This is useful if the structure finite element mesh and the acoustic boundary/finite element mesh are mismatched.
- *CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURACE_ORTHO_FRICTION can now be defined by part set IDs when supplemented by *DEFINE_FRICTION_ORIENTATION. Segment sets with orientation per *DEFINE_FRICTION_ORIENTATION are generated automatically.
- Contact force of *CONTACT_ENTITY is now available in intfor (*DATABASE_BINARY_INTFOR).
- *CONTACT_FORCE_TRANSDUCER_PENALTY will now accept node sets for both the slave and master sides, which should allow them to work correctly for eroding materials. BOTH sides should use node sets, or neither.
- Added option to create a backup penalty-based contact for a tied constraint-based contact in the input (IPBACK on Card E of *CONTACT).
- New option for *CONTACT_ENTITY. If variable SO is set to 2, then a constraint-like option is used to compute the forces in the normal direction. Friction is treated in the usual way.
- *CONTACT_ENTITY: allow friction coefficient to be given by a “coefficient vs time” load curve (input < 0 -> absolute value is the load curve ID). Also, if the friction coefficient bigger or equal 1.0, the node sticks with no sliding at all.

- Minor tweak to the way both MPP and SMP handle nodes sliding off the ends of beams in *CONTACT_GUIDED_CABLE.
- Frictional energy output in sleout (*DATABASE_SLEOUT) supported for *CONTACT_…_MORTAR.
- Tiebreak damage parameter output as "contact gap" in intfor file for *CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_TIEBREAK_MORTAR, OPTION = 9.
- Added MPP support for *CONTACT_2D_AUTOMATIC_SINGLE_SURFACE and *CONTACT_2D_AUTOMATIC_SURFACE_TO_SURFACE.
- Added keyword *CONSTRAINED_MULTIPLE_GLOBAL for defining multi-node constraints for imposing periodic boundary conditions.
- Enhancement for *CONSTRAINED_INTERPOLATION_SPOTWELD (SPR3): calculation of bending moment is more accurate now.
- If *CONSTRAINED_NODAL_RIGID_BODY nodes are shared by several processors with mass scaling on, the added mass is not summed up across processors. This results in an instability of the NRB. (MPP only)
- *ALE_REFINE has been replaced and expanded upon by the *CONTROL_REFINE family of commands. These commands invoke local mesh refinement of shells, solids, and ALE elements based on various criteria.
- Shells or solids in a region selected for refinement (parent element) are replaced by 4 shells or 8 solids, respectively. *CONTROL_REFINE_SHELL applies to shells, *CONTROL_REFINE_SOLID applies to solids and *CONTROL_REFINE_ALE and *CONTROL_REFINE_ALE2D applies to ALE elements. Each keyword has up to 3 lines of input. If only the 1st card is defined, the refinement occurs during the initialization. The 2nd card defines a criterion CRITRF to automatically refine the elements during the run. If the 3rd card is defined, the refinement can be reversed based on a criterion CRITM. All commands are implemented for MPP.
- *CONTROL_REFINE_MPP_DISTRIBUTION distributes the elements required by the refinement across the MPP processes.
- Eliminate automatic writing of a d3plot plot state after each 3D tetrahedral remeshing operation (*CONTROL_REMESHING) to reduce volume of output.
- Generate disbout output (*DATABASE_DISBOUT) for MPP and SMP binout files.
- Extend *DATABASE_MASSOUT to include option to output mass information on rigid body nodes.
- Added new keyword *CHANGE_OUTPUT for full deck restart to override default behavior of overwriting existing ASCII files. For small restart, this option

has no effect since all ASCII output is appended to the result of previous run already.

- Added new option (NEWLENGD) to 2nd field of 3rd card of *CONTROL_OUTPUT to write more detailed legend in ASCII output files. At present, only rcforc and jntforc are implemented.
- Increased default binary file size scale factor (x=) from 7 to 1024. That means the default binary file size will be 1 Gb for single version and 2 Gb for double version.
- Add echo of new "max frequency of element failure summaries" flag (FRFREQ in *CONTROL_OUTPUT) to d3hsp file.
- Support LSDA/binout output for new pllyout file (*DATABASE_PLLYOUT, *ELEMENT_BEAM_PULLEY) in both SMP and MPP.
- Allow degenerated hexahedrons (pentas) for cohesive solid elements (ELFORM = 19, 20) that evolve from an extrusion of triangular shells. The input of nodes on the element cards for such a pentahedron is given by: N1, N2, N3, N3, N4, N5, N6, N6.
- Add new option to activate drilling constraint force for shells in explicit calculations. This can be defined by parameters DRCPSID (part set) and DRCPRM (scaling factor) on *CONTROL_SHELL.
- Add SMP ASCII database "pllyout" (*DATABASE_PLLYOUT) for *ELEMENT_BEAM_PULLEY.
- *FREQUENCY_DOMAIN_ACOUSTIC_BEM:
    - Added an option to output real part of acoustic pressure in time domain.
    - Enabled BEM acoustic computation following implicit transient analysis.
    - Implemented coupling between steady state dynamics and collocation acoustic BEM.
    - Implemented Acoustic Transfer Vector (ATV) to variational indirect BEM acoustics.
    - Enabled boundary acoustic mapping in BEM acoustics.
- *FREQUENCY_DOMAIN_ACOUSTIC_FEM:
    - Added boundary nodal velocity to binary plot file d3acs.
    - Implemented pentahedron elements in FEM acoustics.
    - Enabled using boundary acoustic mapping in FEM acoustics.
- *FREQUENCY_DOMAIN_FRF:
    - Updated FRF to include output in all directions (VAD2 = 4).
    - Added treatment for FRF with base acceleration (node id can be 0).

- *FREQUENCY_DOMAIN_RANDOM_VIBRATION:

  - Updated calculation of PSD and RMS von Mises stress in random vibration environment, based on Sandia National Laboratories report, 1998.

- *FREQUENCY_DOMAIN_RANDOM_VIBRATION_FATIGUE:

  - Implemented an option to incorporate initial damage ratio in random vibration fatigue.

- *FREQUENCY_DOMAIN_RESPONSE_SPECTRUM:

  - Implemented double sum methods (based on Gupta-Cordero coefficient, modified Gupta-Cordero coefficient, and Rosenblueth-Elorduy coefficient).
  - Updated calculating von Mises stress in response spectrum analysis.
  - Implemented treatment for multi simultaneous input spectra.
  - Improved double sum methods by reducing number of loops.

- *FREQUENCY_DOMAIN_SSD:

  - Added the option to output real and imaginary parts of frequency response to d3ssd.
  - Added the option to output relative displacement, velocity and acceleration in SSD computation in the case of base acceleration. Previously only absolute values were provided.

- Implemented keyword *FREQUENCY_DOMAIN_MODE_{OPTION} so that user can select the vibration modes to be used for frequency response analysis.

- Implemented keyword *SET_MODE_{OPTION} so that user can define a set of vibration modes, to be used for frequency response analysis.

- Implemented keyword *FREQUENCY_DOMAIN_PATH to define the path of binary databases containing mode information, used in restarting frequency domain analysis, e.g. frf, ssd, random vibration.

- Compute normal component of impulse for oblique plates in *INITIAL_MINE_-IMPULSE. The feature is no longer limited to horizontal plates.

- Disable license security for *INITIAL_IMPULSE_MINE. The feature is no longer restricted.

- Enabled hourglass type 7 to work well with *INITIAL_FOAM_REFERENCE_-GEOMETRY so that initial hourglass energy is properly calculated and foam will spring back to the initial geometry.

- Accommodate erosion of thin shells in *LOAD_BLAST_ENHANCHED.

- *LOAD_VOLUME_LOSS has been changed such that after the analysis time exceeds the last point on the curve of volume change fraction versus time, the volume change is no longer enforced.
- *LOAD_BODY_POROUS new option AOPT added to assign porosity values in material coordinate system.
- Added *LOAD_SEGMENT_FILE.
- Add new sensor definition, *SENSOR_DEFINE_ANGLE. This card traces the angle formed between two lines.
- *SENSOR_DEFINE_NODE can be used to trace the magnitude of nodal values (coordinate, velocity or accleration) when VID is "0" or undefined.
- Add two new parameters to *SENSOR_DEFINE_ELEMENT, scale factor and power, so that user can adjust the element-based sensor values (strain, stress, force, …).
- Change history variables 10-12 in *MAT_054/*MAT_ENHANCED_COMPOSITE_DAMAGE (thin shells only) to represent strains in material coordinate system rather than in local element coordinate system. This is a lot more helpful for postprocessing issues. This change should not lead to different results other than due to different round-off errors.
- New features and enhancements to *MAT_244/*_MAT_UHS_STEEL:
    - Added implicit support for MAT_244.
    - Changed the influence of the austenite grain size in Mat244 according to Li et al.
    - Changed the start temperatures to fully follow WATT et al and Li et al.
    - Hardness calculation is now improved when noncontinuous cooling is applied i.e., tempering.
    - Added temperature dependent Poisson ratio and advanced reaction kinetics.
    - Added new advanced option to describe the thermal expansion coefficients for each phase.
    - Added option to use Curve ID or a Table ID for describing the latent heat generation during phase transormations.
    - Added support for table definition for Youngs modulus. Now you can have one temperature dependent curve for each of the 5 phases
- Added support for implicit to *MAT_188.
- Added material model *MAT_273/*MAT_CDPM/*MAT_CONCRETE_DAMAGE_PLASTIC_MODEL. This model is aimed at simulations where failure of concrete structures subjected to dynamic loadings is sought. The model is based on effective stress plasticity and has a damage model based on both plastic and elastic strain measures. Implemented for solids only but both for explicit and

implicit simulations. Using an implicit solution when damage is activated may trigger a slow convergense. IMFLAG = 4 or 5 can be useful.

- Added an option in *MAT_266 (*MAT_TISSUE_DISPERSED) so that the user can tailor the active contribution with a time dependent load curve instead of using the internal hardcoded option. See ACT10 in the User's Manual.
- *MAT_173/*MAT_MOHR_COULOMB is available in 2D.
- Enable *MAT_103 and *MAT_104 to discretize the material load curves according to the number of points specified by LCINT in *CONTROL_SOLUTION.
- Implement Prony series up to 18 terms for shells using *MAT_076/*MAT_GENERAL_VISCOELASTIC.
- Added *DEFINE_STOCHASTIC_VARIATION and the STOCHASTIC option for *MATs 10, 15, 24, 81, 98 for shells, solids, and type 13 tets. This feature defines a stochastic variation in the yield stress and damage/failure of the aforementioned material models.
- Add Moodification for *DEFINE_CONNECTION_PROPERTIES, PROPRUL = 2: thinner weld partner is first partner, PROPRUL = 3: bottom (nodes 1-2-3-4) weld partner is first partner.
- Add spotweld area to debug output of *DEFINE_CONNECTION_PROPERTIES which is activated by *CONTROL_DEBUG.
- Add support of *MAT_ADD_EROSION option NUMFIP < 0 for standard (non-GISSMO) failure criteria. Only for shells.
- Improve implicit convergence of *MAT_ADD_EROSION damage model GISSMO by adding damage scaling (1-D) to the tangent stiffness matrix.
- Provide plastic strain rates (tension/compression, shear, biaxial) as history variables no. 16, 17, and 18 for *MAT_187.
- Add new variables to user failure routine matusr_24 (activated by FAIL < 0 on *MAT_024 and other materials): integration point numbers and element id.
- Add new energy based, nonlocal failure criterion for *MAT_ADD_EROSION, parameters ENGCRT (critical energy) and RADCRT (critical radius) after EPSTHIN. Total internal energy of elements within a radius RADCRT must exceed ENGCRT for erosion to occur. Intended for windshield impact.
- Add new option to *MAT_054 for thin shells: Load curves for rate dependent strengths and a rate averaging flag can be defined on new optional card 9.
- Add new option for *MAT_MUSCLE: Input parameter SSP < 0 can now refer to a load curve (stress vs. stretch ratio) or a table (stress vs. stretch ratio vs. normalized strain rate).
- Expand list of variables for *MAT_USER_DEFINED_MATERIAL_MODELS by characteristic element size and element id.

- Enable *MAT_USER_DEFINED_MATERIAL_MODELS to be used with tetrahedron element type 13. New sample routines "umat41_t13" and "umat41v_t13" show corresponding pressure calculation in the elastic case.
- Add a new feature to *MAT_125 allowing C1 and C2 to be used in calculation of back stress. When plastic strain < 0.5%, C1 is used, otherwise C2 is used as described in Yoshida's paper.
- Extend non-linear strain path (_NLP_FAILURE) in *MAT_037 to implicit.
- *MAT_173/*MAT_MOHR_COULOMB now works in ALE. A new option has been added to suppress the tensile limit on hydrostatic stress recommended for ALE multi-material use.
- Upgraded *MAT_172/*MAT_CONCRETE_EC2.
    - Corrections to DEGRAD option.
    - Concrete and reinforcement types 7 and 8 have been added to reflect changes to Eurocode 2.
    - Extra history variables for reinforcement stress and strain are now output as zero for zero-fraction reinforcement directions.
- Added RCDC model for solid *MAT_082.
- Added Feng's failure model to solid *MAT_021.
- Added *MAT_027 for beams.
- Added *DEFINE_HAZ_PROPERTIES and *DEFINE_HAZ_TAILOR_WELDED_-BLANK for modifying material behavior near a spot weld.
- Added fourth rate form to viscoplastic Johnson-Cook model (*MAT_015).
- Added option to *MAT_224 to not delete the element if NUMINT = -200.
- New damage initiation option 3 in multi fold damage criteria in *MAT_ADD_-EROSION. Very similar to option 2 but insensitive to pressure.
- Added rotational resistance in *MAT_034/*MAT_FABRIC. Optionally the user may specify the stiffness, yield and thickness of and elastic-perfectly-plastic coated layer of a fabric that results in a rotational resistance during the simulation.
- FLDNIPF < 0 in *MAT_190/*MAT_FLD_3-PARAMETER_BARLAT for shell elements means that failure occurs when all integration points within a relative distance of -FLDNIPF from the mid surface has reached the fld criterion.
- A computational welding mechanics *MAT_270/*MAT_CWM material is available that allows for element birth based on a birth temperature as well as annealing based on an annealing temperature. The material is in addition a thermo-elasto-plastic material with kinematic hardening and temperature dependent properties.

- Added *MAT_271/*MAT_POWDER, a material for manufacturing (i.e., compaction and sintering) of cemented carbides. It is divided into an elastic-plastic compaction model that is supposed to be run in a first phase, and a visco-elastic sintering model that should be run in a second phase. This model is for solid elements.

- For IHYPER = 3 on a *MAT_USER_DEFINED_… shell material, the deformation gradient is calculated from the geometry instead of incremented by the velocity gradient. The deformation gradient is also passed to the user defined subroutines in the global system together with a transformation matrix between the global and material frames. This allows for freedom in how to deal with the deformation gradient and its transformations in orthotropic (layered) materials.

- The Bergstrom-Boyce viscoelastic rubber model is now available in explicit and implicit analysis as *MAT_269/*MAT_BERGSTROM_BOYCE_RUBBER. The Arruda-Boyce elastic stress is augmented with a Bergstrom-Boyce viscoelastic stress corresponding to the response of a single entangled chain in a polymer gel matrix.

- Added a new parameter IEVTS to *MAT_USER_DEFINED_MATERIAL_MODELS (*MAT_041-050). IEVTS is optional and is used only by thick shell formulation 5. It points to the position of E(a) in the material constants array. Following E(a), the next 5 material constants must be E(b), E(c), v(ba), v(ca), and v(cb). This data enables thick shell formulation 5 to calculate an accurate thickness strain, otherwise the thickness strain will be based on the elastic constants pointed to by IBULK and IG.

- Implemented enhancements to fabric material (*MAT_034), FORM = 14. Stress-strain curves may include a portion for fibers in compression. When unload/reload curves with negative curve ID are input (curve stretch options), the code that finds the intersection point now extrapolates the curves at their end rather than simply printing an error message if an intersection point cannot be found before the last point in either curve.

- Map 1D to 3D by beam-volume averaging the 1D data over the 3D elements (*INITIAL_ALE_MAPPING).

- In a 3D to 3D mapping (*INITIAL_ALE_MAPPING), map the relative displacements for the penalty coupling in *CONSTRAINED_LAGRANGE_IN_SOLID.

- The [name].xy files associated with *DATABASE_ALE_MAT are now created when sense switches sw1, sw2, quit, or stop are issued.

- *ALE_ESSENTIAL_BOUNDARY is available in 2D.

- *DATABASE_FSI is available for 2D (MPP).

- *ALE_ESSENTIAL_BOUNDARY implemented to apply slip-only velocity BC along ALE mesh surface.

- *CONTROL_ALE flag INIJWL = 2 option added to balance initial pressure state between ALE Soil and HE.
- Include SPH element (*ELEMENT_SPH) in time step report.
- Time step and internal energy of 2D axisymmetric SPH elements are calculated in a new way more consistent with the viscosity force calculation.
- Only apply viscosity force to x and y components of 2D axisymmetric SPH element, not on hoop component.
- MAXV in *CONTROL_SPH can be defined as a negative number to turn off velocity checking.
- Improve calculation of 2D axisymmetric SPH contact force in *DEFINE_SPH_TO_SPH_COUPLING.
- Added the following material models for SPH particles: *MAT_004/*MAT_ELASTIC_PLASTIC_THERMAL (3D only) and *MAT_106/*MAT_ELASTIC_VISCOPLASTIC_THERMAL
- Added a new parameter DFACT for *DEFINE_SPH_TO_SPH_COUPLING. DFACT invokes a viscous term to damp the coupling between two SPH parts and thereby reduce the relative velocity between the parts.
- Added BOUNDARY_CONVECTION and BOUNDARY_RADIATION for explicit SPH thermal solver.
- *CONTROL_REMESHING_EFG:
  - Add eroding failed surface elements and reconstructing surface in EFG adaptivity.
  - Add a control parameter for monotonic mesh resizing in EFG adaptivity.
  - Add searching and correcting self-penetration for adaptive parts in 3D tetrahedron remeshing.
- Enhance 3D axisymmetric remeshing with 6-node/8-node elements
- (*CONTROL_REMESHING):
  - Use RMIN/RMAX along with SEGANG to determine element size.
  - Remove the restriction that the reference point of computational model has to be at original point (0, 0, 0).
  - Rewrite the searching algorithm for identifying the feature lines of cross-sections in order to provide more stable remeshing results.
- Improve rigid body motion in EFG shell type 41.
- Support EFG pressure smoothing in EFG solid type 42 for *MAT_ELASTIC_VISCOPLASTIC_THERMAL.
- Add visco effect for implicit EFG solid type 42.

- Add new EFG solid type 43 (called Meshfree-Enriched FEM, MEFEM) for both implicit and explicit. This element formulation is able to relieve the volumetric locking for nearly-incompressible material (eg. rubber) and performs strain smoothing across elements with common faces.
- EFG shell adaptivity no longer requires a special license.
- Application of EFG in an implicit analysis no longer requires a special license.
- Add *SENSOR_CONTROL for prescribed motion constraints in implicit.
- Update *INTERFACE_LINKING_NODE in implicit to catch up with explicit, including adding scaling factors.
- Add support for *DATABASE_RCFORC_MOMENT for implicit.
- Enhance Iterative solvers for Implicit Mechanics.
- Add, after the first implicit time step, the output of projected cpu and wall clock times. This was already in place for explicit. Also echo the termination time.
- Add variable MXDMP in *CONTROL_THERMAL_SOLVER to write thermal conductance matrix and right-hand side every MXDMP time steps.
- Add keyword *CONTROL_THERMAL_EIGENVALUE to calculate eigenvalue(s) of each thermal conductance matrix.
- Added thermal material model *MAT_THERMAL_ORTHOTROPIC_TD_LC. This is an orthtropic material with temperature dependent properties defined by load curves.
- Changed structured file format for control card 27 (first thermal control card). Several input variables used i5 format limiting their value to 99,999. A recent large model exceeded this limit. The format was changed to i10. This change is not backward compatible. Old structured input files will no longer run unless control card 27 is changed to the new i10 format. This change does not affect the KEYWORD file.
- Add thermal material *MAT_T07/*MAT_THERMAL_CWM for welding simulations, to be used in conjunction with mechanical counterpart *MAT_-270/*MAT_CWM.
- Modify decomposition costs of *MAT_181 and *MAT_183.
- Introduce new timing routines and summary at termination.
- Echo "MPP contact is groupable" flag to d3hsp
- Bodies using *MAT_RIGID_DISCRETE were never expected to share nodes with non-rigid bodies, but this now works in MPP.
- There is no longer any built-in limitation on the number of processors that may be used in MPP.

- Echo contents of the MPP pfile (including keyword additions) to the d3hsp and mes0000 files.
- Add new keyword *CONTROL_MPP_PFILE, which allows for insertion of text following this command to be inserted into the MPP pfile (p = pfile).
- Change in MPP treatment of *CONSTRAINED_TIE-BREAK. They now share a single MPI communicator, and a single round of communication. This should improve performance for problems with large numbers of these, without affecting the results.
- Added two input variables for *CONTROL_FORMING_ONESTEP simulation, TSCLMIN is a scale factor limiting the thickness reduction and EPSMAX defines the maximum plastic strain allowed.
- Added output of strain and stress tensors for onestep solver *CONTROL_FORMING_ONESTEP, to allow better evaluation of formability.
- Improved *CONTACT_AUTO_MOVE: before changes the termination time, and it causes problems when several tools need to be moved. Now *CONTACT_AUTO_MOVE does not change the termination time, but changes the current time. In this way, several tools can be moved without the need to worry about the other tool's move. This is especially useful in multi-flanging and hemming simulations.
- Made improvements to previously undocumented keyword *INTERFACE_-BLANKSIZE, including adding the options_INITIAL_TRIM, and_INITIAL_-ADAPTIVE. This keyword was developed for blank size development in sheet metal forming. Generally, for a single forming process, only the option_DEVELOPMENT is needed and inputs are an initial estimated blank shape, a formed blank shape, and a target blank shape in either mesh or boundary coordinates. Output will be the calculated/corrected initial blank shape. Initial blank mesh and formed blank mesh can be different (e.g. adaptive). For a multi-stamping process involving draw, trimming and flanging, all three options are needed. Related commands for blank size estimation are *CONTROL_FORMING_ONESTEP, and for trim line development, *CONTROL_FORMING_UNFLANGING.
- Made improvements and added features to previously undocumented keyword *CONTROL_FORMING_UNFLANGING, this keyword unfolds flanges of a deformable blank, e.g., flanged or hemmed portions of a sheet metal part, onto a rigid tooling mesh using the implicit static solver. It is typically used in trim line mapping during a draw die development process. The roots of the flanges or hemmed edges are automatically processed based on a user input of a distance tolerance between the flanges/hemmed edges and rigid tool. It includes the ability to handle a vertical flange wall. Other keywords related to blank size development are, *CONTROL_FORMING_ONESTEP, and *INTERFACE_-BLANKSIZE_DEVELOPMENT.

- Added keyword *CONTROL_FORMING_OUTPUT which allows control of d3plot output by specifying distances to tooling home. It works with automatic position of stamping tools using *CONTROL_FORMING_AUTOPOSITION_PARAMETER.
- Added the LOCAL_SMOOTH option to *INTERFACE_COMPENSATION_NEW which features smoothing of a tool's local area mesh, which could otherwise become distorted due to, e.g., bad/coarse mesh of the original tool surface, tooling pairs (for example, flanging post and flanging steel) do not maintain a constant gap and several compensation iterations. This new option also allows for multiple regions to be smoothed. Local areas are defined by *SET_LIST_NODE_SMOOTH.
- Added output to rcforc for *DEFINE_DE_TO_SURFACE_COUPLING.
- Implement traction surface for *DEFINE_DE_TO_SURFACE_COUPLING.
- Add keyword *DATABASE_BINARY_DEMFOR with command line option dem = dem_int_force. This will turn on the DEM interface force file for DEM coupling option. The output frequency is controlled by the new keyword.
- Add new feature *DEFINE_DE_INJECTION to allow DEM particle dropping from user defined plane.
- Add new option_VOLUME to *ELEMENT_DISCRETE_SPHERE. This will allow DEM input based on per unit density and use *MAT card to get consistent material properties.
- Added FORM = -4 for *ELEMENT_SHELL_NURBS_PATCH. Rotational dofs are automatically set at control points at the patch boundaries, whereas in the interior of the patch only translational dofs are present. This helps for joining multiple nurbs patches at their C0-boundaries.
- Disabled FORM = 2 and 3 for *ELEMENT_SHELL_NURBS_PATCH. These formulations are experimental and not fully validated yet.
- Added energy computation for isogeometric shells (*ELEMENT_SHELL_NURBS_PATCH) to matsum.
- Allow isogeometric shells (*ELEMENT_SHELL_NURBS_PATCH) to behave as rigid body (*MAT_RIGID).
- Added “g” as abbreviation for gigawords in specification of memory on execution line, e.g, memory = 16g is 16 billion words.
- Suppress non-printing characters in *COMMENT output.
- Add command line option “pgpkey” to output the current public PGP key used by LS-DYNA. The output goes to the screen as well as a file named “lstc_ pgpkey.asc” suitable for directly importing into GPG.

- When reading the NAMES file, allow a “+” anywhere on a line to indicate there will be a following line, not just at the end. This was never intended, but worked before r73972 and some customers use it that way.
- Check for integer overflow when processing command line arguments and the memory value on the *KEYWORD card.
- Added new capability for *INTERFACE_LINKING_NODE to scale the displacements of the moving interface.
- Support for *KEYWORD_JOBID with internal *CASE driver.
- *DAMPING_FREQUENCY_RANGE now works for implicit dynamic solutions. An error check has been added to ensure that the timestep is small enough for the damping card to work correctly.
- Added new option *DAMPING_FREQUENCY_RANGE_DEFORM to damp only the deformation instead of the global motion.
- Added *DEFINE_VECTOR_NODES. A vector is defined using two node IDs.
- Add sense switch “prof” to output current timing profile to message (SMP) file or mes#### (MPP) files. Also, for MPP only, collect timing information from processor and output to prof.out when sense switch “prof” is detected.

Capabilities added during 2013/2014 to create LS-DYNA R7.1:

- Add MUTABLE option for *PARAMETER so that parameter values can be redefined later in the input deck.
- Change MPP treatment of two-sided *CONTACT_FORCE_TRANSDUCER so that proper mass and moment values can be output to the rcforc file.
- MPP support for non-zero birthtime for *CONTACT_SINGLE_EDGE.
- Add new command line option “ldir=” for setting a *local* working directory. In MPP, this has the same effect as setting the “directory { local }” pfile option (and it overrides that option). For SMP, it indicates a directory where local, working files should be placed.
- Add support for SMOOTH option in MPP groupable contact.
- Add new keyword card *CONTROL_REQUIRE_REVISION to prevent the model from being run in old versions of LS-DYNA.
- Add part set specification for dynamic relaxation with implicit using *CONTROL_DYNAMIC_RELAXATION. This is a new feature specified with idrflg = 6 on *CONTROL_DYNAMIC_RELAXATION. This allows implicit to be used for the dynamic relaxation phase for models involving parts being modeled with SPH and/or ALE while excluding those parts from the dynamic relaxation phase.

- Add new feature for implicit automatic time step control to cooperate with thermal time step control. On *CONTROL_IMPLICIT_AUTO, IAUTO = 2 is the same as IAUTO = 1 with the extension that the implicit mechanical time step is limited by the active thermal time step.
- On *CONTROL_IMPLICIT_SOLUTION, add negative value of MAXREF for implicit mechanics. Nonlinear iteration will terminate after |MAXREF| iterations. With MAXREF < 0 convergence is declared with a warning. Simulation will continue. Positive values of MAXREF still cause failure of convergence to be declared leading to either a time step reduction or an error termination.
- Add *CONTROL_IMPLICIT_MODAL_DYNAMIC keywords and features. This elevates the modal dynamic features of IMASS = 2 on *CONTROL_IMPLICIT_-DYNAMICS. It also adds additional features of damping and mode selection and stress computations.
- New material model *MAT_DRY_FABRIC / MAT_214, which can be used in modeling high strength woven fabrics with transverse orthotropic behavior.
- Add *ALE_COUPLING_NODAL_PENALTY, penalty-based nodal coupling with ALE.
- Add type 8 *ELEMENT_SEATBELT_PRETENSIONER which takes energy-time curve, instead of pull-in or force curve.
- Add type 9 *ELEMENT_SEATBELT_PRETENSIONER for energy-based buckle / anchor pretensioner.
- Add *DATABASE_BINARY_FSILNK. This feature stores coupling pressure from *CONSTRAINED_LAGRANGE_IN_SOLID in a binary time history file for use in a separate model that does not include ALE.
- Add *LOAD_SEGMENT_FSILNK. Use pressure loads stored in aforementioned binary time history file to load model that does not have ALE elements.
- Add new keyword *DEFINE_SPH_DE_COUPLING to allow SPH particles to contact discrete element spheres (DES).
- Add MOISTURE option to *MAT_076 solids. Allows moisture content to be input as a function of time. Material parameters are then scaled according to the moisture and a moisture strain is also introduced.
- Add *RIGIDWALL_FORCE_TRANSDUCER to output forces from rigidwalls acting on node sets.
- Add LOG_INTERPOLATION option to *MAT_024. This offers an alternate means of invoking logarithmic interpolation for strain rate effects. The other way is to input the natural log of strain rate in the table LCSS.
- Add capability in *MAT_ADD_EROSION (NUMFIP < -100) to set stress to zero in each shell integration point as it reaches the failure criterion. When |NUMFIP|-100 integration points have failed, the shell is eroded. In contrast,

when NUMFIP > 0, failed integration points continue to carry full load as though they were unfailed until element erosion occurs.

- Add new keyword, *PARAMETER_TYPE, for use by LS-PrePost when combining keyword input files. The appropriate offset is applied to each ID value defined using *PARAMETER_TYPE, according to how that ID is used.
- Allow use of load curve to specify damping as a function of time in *DAMPING_RELATIVE.
- Add a segment based (SOFT = 2) contact option to include the overlap area in the contact stiffness calculation. This is good for improving the friction calculation and possibly for implicit convergence. The option is turned on by setting FNLSCL > 0 and DNLSCL = 0. As DNLSCL = 0, the contact stiffness is not nonlinear. This new option is also useful when used with another improvement that was made to the FS = 2 friction coefficient by table lookup option in segment based contact. When the above mentioned FNLSCL > 0, option is used, the FS = 2 option is now very accurate.
- Add a new RCDC damage option, *MAT_PLASTICITY_WITH_DAMAGE_ORTHO_RCDC1980 which is consistent with the WILKINS paper. It uses the principal values of stress deviators and a different expression for the A_d term.
- Add a TIETYP option to *CONTACT_2D_AUTOMATIC. By default the tied contact automatically uses constraint equations when possible for 2D tied contact. If a conflict is detected with other constraints, or to avoid 2-way constraints, penalty type ties are used when constraints are not possible. The TIETYP option, when set to 1, causes all ties to use the penalty method. This is useful if in spite of the code's best efforts to avoid problems, there is still a conflict in the model.
- Add a scale factor for scaling the frictional stiffness for contact. The parameter is FRICSF on optional card E and it's only supported for segment based (SOFT = 2) contact. This was motivated by a rubber vs. road skidding problem where the friction coefficent had static, dynamic and decay parameters defined. The growth of the frictional force was too slow so the static coulomb value could not be achieved. By scaling the frictional stiffness higher, the coulomb value could approach the static value.
- Add keyword *CONTACT_2D_AUTOMATIC_FORCE_TRANSDUCER. Like the 3D force transducers, it does no contact calculation but only measures the contact forces from other contact definitions. When only a slave side is defined, the contact force on those segments is measured. Currently, two surface force transducers are not available.
- Add options to *MAT_058:
  - Load curves for rate dependent strain values (E11C, E11T, …) can be defined on new optional card 9.

  - Load curves for rate dependent strengths (XC, XT, …) and a rate averaging flag can be defined on new optional card 8.
  - Abscissa values in above curves are taken to be natural log of strain rate when the first value is negative.
  - Add optional transverse shear damage to *MAT_058.

- Add MAT_261 and MAT_262 for general use. *MAT_261 is *MAT_LAMINATED_FRACTURE_DAIMLER_PINHO. *MAT_262 is *MAT_LAMINATED_FRACTURE_DAIMLER_CAMANHO.
- Add pentahedra cohesive solid element types (TYPE = 21 & 22). Type = 21 is the pentahedra version of Type = 19 and Type = 22 is the pentahedra version of Type = 20. Using ESORT.gt.0 in *CONTROL_SOLID will automatically sort out the pentahedra elements (19 to 21 and 20 to 22).
- Add *DEFINE_DE_BY_PART to define control parameters for DES by part ID, including damping coefficient, friction coefficient, spring constant, etc. If defined, it will overwrite the parameters in *CONTROL_DISCRETE_ELEMENT.
- Add new feature for *MAT_030 (*MAT_SHAPE_MEMORY) as optional 3rd card. Curves or tables (strain rate dependency) can be defined to describe plastic loading and unloading behavior.
- New feature for *ELEMENT_BEAM_PULLEY. Beam elements BID1 and BID2 can now both be defined as “0” (zero). In that case, adjacent beam elements are automatically detected. Therefore, the first two beam elements with nodal distance < 1.0e-6 to the pulley node (PNID) will be chosen.
- Add new feature to *MAT_ADD_EROSION's damage model GISSMO. By default, damage is driven by equivalent plastic strain. Now, users can optionally define another history variable as driving quantity by setting DMGTYP.
- Add volumetric plastic strain to *MAT_187 as history variable 6.
- Add internal energy calculation for *ELEMENT_BEAM_PULLEY.
- Add viscoplastic option to *MAT_157: new parameter VP on Card 5, Column 6.
- Add new keyword *MAT_ADD_COHESIVE which is intended to make 3D material models available for cohesive elements.
- Add new parameters to *MAT_CABLE_DISCRETE / *MAT_071. MXEPS (Card 2, Column 4) is equal the maximum strain at failure and MXFRC (Card 2, Column 5) is equal to the maximum force at failure
- Add *MAT_124 as potential weld partner material for PROPRUL = 2/3 of *DEFINE_CONNECTION_PROPERTIES.
- Add new material *MAT_TOUGHENED_ADHESIVE_POLYMER (TAPO) or *MAT_252 for epoxy-based, toughened, ductile adhesives.

- Add new option to *MAT_002_ANIS: parameter IHIS on Card 4, Column 8. IHIS = 0: terms C11, C12, … from Cards 1, 2, and 3 are used. IHIS = 1: terms C11, C12, … initialized by *INITIAL_STRESS_SOLID's extra history variables.
- Add new option to *MAT_102. Instead of constant activation energy Q, one can define a load curve LCQ on Card 2, Column 7:
    - LCQ.GT.0: Q as function of plastic strain
    - LCQ.LT.0: Q as function of temperature
- Add new option to *MAT_071 (MAT_CABLE_DISCRETE_BEAM). New parameter FRACL0 (Card 2, Column 3) is fraction of initial length that should be reached over time period of TRAMP. That means the cable element length gets modified from L0 to FRACL0*L0 between t = 0 and t = TRAMP.
- Add internal energy calculation for SPR models *CONSTRAINED_INTERPOLATION_SPOTWELD (SPR3) and *CONSTRAINED_SPR2. Their contribution was missing in energy reports like glstat.
- Add new failure model OPT = 11 to *MAT_SPOTWELD/*MAT_100 for beam elements.
- Add three new failure criteria for shell elements to *MAT_ADD_EROSION on optional card 4, columns 6-8:
    - LCEPS12: load curve in-plane shear strain limit vs. element size.
    - LCEPS13: load curve cross-thickness shear strain limit vs. element size.
    - LCEPSMX: load curve in-plane major strain limit vs. element size.
- Add new capability to *MAT_ADD_EROSION damage model GISSMO. Strain rate scaling curve LCSRS can now contain natural logarithm values of strain rates as abscissa values. This is automatically assumed when the first value is negative.
- Add new parameter NHMOD to *MAT_266. The constitutive model for the isotropic part can now be chosen:
    - NHMOD = 0: original implementation (modified Neo-Hooke)
    - NHMOD = 1: standard Neo-Hookeon (as in umat45)
- New keyword *DEFINE_TABLE_MATRIX is an alternative way of defining a table and the curves that the table references from a single unformatted text file, e.g., as saved from an Excel spreadsheet.
- Change long format so that all data fields are 20 columns and each line of input can hold up to 200 columns. In this way, the number of input lines is the same for long format as for standard format.
    - 8 variables per line in long format = 160 columns

  - 10 variables per line in long format = 200 columns

- Add a new option (SOFT = 6) in *CONTACT_FORMING_NODES_TO_SURFACE for blank edge and guide pin contact.
- Add user-defined criteria for mesh refinement (or coarsening) in *CONTROL_-REFINE_….
- Add new contact option that currently only works for MPP SINGLE_SURFACE contact with SOFT = 0 or 1. If SRNDE (field 4 of optional card E) is a 1, then free edges of the contact definition will be rounded WITHOUT extending the segments. Rather than having cylindrical caps on the ends of the segments, the "corners" of the squared off thickness are rounded over.
- Add geometric contact entity type -3 "finite cylinder".
- Add irate = 2 to *CONTROL_IMPLICIT_DYNAMICS to turn off rate effects for both implicit and explicit.
- Add quadratic 8-node and 6-node shells (shell formulations 23 and 24).
- Add LOG_LOG_INTERPOLATION option for table defining strain rate effects in *MAT_083, *MAT_181, and *MAT_183.
- Add automatic generation of null shells for quadratic shell contact (*PART_DUPLICATE_NULL_OVERLAY).
- Add beam contact forces to rcforc output (*DATABASE_RCFORC).
- Add SHL4_TO_SHL8 option to *ELEMENT_SHELL to automatically convert 4-node shells to 8-node quadratic shells.
- Add 3-node beam element with quadratic interpolation that is tailored for the piping industry. It includes 12 degrees of freedom, including 6 ovalization degrees of freedom, per node for a total of 36 DOF. An internal pressure can be given that can stiffen and elongate the pipe.
  - ELFORM = 14 in *SECTION_BEAM.
  - *ELEMENT_BEAM_ELBOW.
  - NEIPB in *DATABASE_EXTENT_BINARY to direct output of elbow loop-stresses to d3plot. Otherwise, output goes to ASCII file elbwls.k.
  - Supported by a subset of material models including mats 3, 4, 6, 153, 195.
- Add discrete element option DE to *DATABASE_TRACER.
  - Includes variable RADIUS. average result of all
    - RADIUS > 0: Reports the average result of all DE particles in a spherical volume having radius = RADIUS and centered at the tracer.
    - RADIUS < 0: Reports result of the closest particle to the tracer.

  - If a tracer node NID is given, then the tracer moves with this node. The node must belong to a DES.

- Add new options *PART_COMPOSITE_LONG and *ELEMENT_SHELL_COMPOSITE_LONG. In contrast to "COMPOSITE", one integration point is defined per card. This is done to allow for more informations, e.g. new variable "ply id".
- Add support of *MAT_ADD_EROSION option NUMFIP < 0 for standard (non-GISSMO) failure criteria. Only for shells.
- Add viscoplastic behavior to *MAT_157, i.e., parameter LCSS can now refer to a table with strain rate dependent yield curves.
- Add singular finite element with midside nodes for 2D plane strain fracture analysis (ELFORM = 55 in *SECTION_SHELL). This is an 8-noded element and can induce a singular displacement field by moving mid-side nodes to quarter locations.
- If HCONV < 0 in *AIRBAG_PARTICLE, |HCONV| is a curve of heat convection coefficient vs. time.
- Add new option DECOMPOSITION for *AIRBAG_PARTICLE -- MPP only.This will automatically invoke the recommended decomposition commands, *CONTROL_MPP_DECOMPOSITION_BAGREF (if applicable) and *CONTROL_-MPP_DECOMPOSITION_ARRANGE_PARTS, for the bag.
- Add new blockage option for vents in *AIRBAG_PARTICLE:
  - blockage considered
    - .eq.0: no
    - .eq.1: yes
    - .eq.2: yes, exclude external vents
    - .eq.3: yes, exclude internal vents
    - .eq.4: yes, exclude all vents
- Add option in *CONTROL_CPM to consider CPM in the time step size calculation.
- When using *AIRBAG_PARTICLE with IAIR = 2, user should keep mole / particle similar between inflator gas and initial air particles to ensure the correct elastic collision. If different by more than 10%, code will issue warning message and provide the suggested initial air particle number.
- Enable *DEFINE_CURVE_FUNCTION for *SECTION_POINT_SOURCE_MIXTURE and *SECTION_POINT_SOURCE.
- Make *BOUNDARY_PRESCRIBED_MOTION_SET compatible with *CONTROL_REFINE

- Change *BOUNDARY_ACOUSTIC_COUPLING_MISMATCH to rank order opposing acoustic faces and structural segments by proximity, thereby accelerating the preprocessing stage, enhancing reliability and allowing some liberalization of the search parameters.
- Implement hemispherical geometry for particle blast (*DEFINE_PBLAST_-GEOMETRY).
- Add explosive type for *PARTICLE_BLAST.
- For particle-based blast *PARTICLE_BLAST:
  - Include random distribution of initial air molecules
  - Modify algorithm to account for the non-thermally-equilibrated state of high velocity gas.
- Improve particle contact method for particle-based blast loading *PARTICLE_-BLAST.
- *CONTACT now works for parts refined using *CONTROL_REFINE_SOLID or *CONTROL_REFINE_SHELL.
- Improve calculation of shell element contact segment thicknesses, particularly at material boundaries.
- MPP: Add output to rcforc file for *CONTACT_AUTOMATIC_TIEBREAK to record the # of nodes tied, and the total tied area.
- MPP: Add calculation of "contact gap" for master side of FORMING contact.
- MPP: Add support for table-based friction (FS = 2.0) to groupable contact.
- Implement splitting-pinball contact, Belytschko & Yeh (1992, 1993). This new contact option is invoked by setting SOFT = 2, SBOPT = 3 and DEPTH = 45. A penetration check method based on LS-PrePost version 4.0 is implemented for the new bilinear-patch-based contact, SOFT = 2, DEPTH = 45 & Q2TRI = 0. The new method provides more accurate intersection information when Q2TRI = 0.
- Add support for birth time for *CONTACT_2D_AUTOMATIC_TIED.
- Improve the segment based single surface contact search for thick segment pairs that are too close together. The code was not working well with triangluar segments. This change affects models with shell segments that have thickness greater than about 2/3 of the segment length.
- Enable segment based quad splitting options to work when shell sets or segment sets are used to define the surface that will be split. This is really a bug fix because there was no check to prevent this and the result was writing past the allocated memory for segment connectivites.
- Allow *CONSTRAINED_INTERPOLATION to use node set to define the independent nodes.

- Add a length unit to the tolerance used for the checking of noncoincident nodes in *CONSTRAINED_JOINTs excluding spherical joints. The old tolerance was 1.e-3. The new tolerance is 1.e-4 times the distance between nodes 1 and 3. The error messages were changed to warnings since this change might otherwise cause existing models to stop running.
- Add d3hsp output for *CONSTRAINED_INTERPOLATION_SPOTWELD (SPR3) and *CONSTRAINED_SPR2. Can be deactivated by setting NPOPT = 1 on *CONTROL_OUTPUT.
- Support NFAIL1 and NFAIL4 of *CONTROL_SHELL in coupled thermal-mechanical analysis, i.e. erode distorted elements instead of error termination.
- PTSCL on *CONTROL_CONTACT can be used to scale contact force exerted on shell formulations 25, 26, 27 as well as shell formulations 2, 16 (IDOF = 3).
- Use SEGANG in *CONTROL_REMESHING to define positive critical angle (unit is radian) to preserve feature lines in 3D tetrahedral remeshing (ADPOPT = 2 in *PART).
- For 3D solid adaptive remeshing including ADPOPT = 2 and ADPOPT = 3 (*PART), the old mesh will be used automatically if the remesher fails generating a new mesh.
- Add option INTPERR on *CONTROL_SHELL (Optional Card 3, Column 8). By default, warning messages INI+143/144/145 are written in case of non-matching number of integration points between *INITIAL_STRESS_SHELL and *SECTION_SHELL. Now with INTPERR = 1, LS-DYNA can terminate with an error.
- Add variable D3TRACE on *CONTROL_ADAPTIVE: The user can now force a plot state to d3plot just before and just after an adaptive step. This option is necessary for tracing particles across adaptive steps using LS-PrePost.
- By putting MINFO = 1 on *CONTROL_OUTPUT, penetration info is written to message files for mortar contact., see also *CONTACT_…. Good for debugging implicit models, not available for explicit.
- Change the default scale factor for binary file sizes back to 70. This value can be changed using “x=” on the execution line. In version R7.0, the default value of x is 4096, and that sometimes leads to difficulty in postprocessing owing to the large size of the d3plot file(s).
- Enable *CONTROL_OUTPUT flag, EOCS, which wasn't having any effect on the shells output to elout file.
- *DATABASE_FSI_SENSOR: Create sensors at solid faces in 3D and at shell sides in 2D.
- *DATABASE_PROFILE: Implement the option DIR = 4 to plot data with curvilinear distributions and the flag UPDLOC to update the profile positions.
- In *CONTROL_SHELL, add options for deletion of shells based on:

  - diagonal stretch ratio (STRETCH)
  - w-mode amplitude in degrees (W-MODE)

- New element formulation ELFORM = 45 in *SECTION_SOLID: Tied Meshfree-enriched FEM (MEFEM). This element is based on the 4-noded MEFEM element (ELFORM = 43, *SECTION_SOLID). Combined with *CONSTRAINED_TIED_-NODES_FAILURE, *SET_NODE_LIST and cohesive model, this element can be used to model dynamic multiple-crack propagation along the element boundaries.
- New high order tetrahedron CPE3D10 based on Cosserat Point theory can be invoked by specifying element formulation ELFORM = 16 and combining this with hourglass formulation IHQ = 10. See *SECTION_SOLID and *HOURGLASS.
- Add database D3ACS for collocation acoustic BEM (*FREQUENCY_DOMAIN_-ACOUSTIC_BEM) to show the surface pressure and normal velocities.
- Implement biased spacing for output frequencies for random vibration (*FREQUENCY_DOMAIN_RANDOM_VIBRATION).
- Add frequency domain nodal or element velocity output for acoustic BEM (*FREQUENCY_DOMAIN_ACOUSTIC_BEM).
- Implement boundary acoustic mapping to acoustic BEM in MPP (*BOUNDARY_ACOUSTIC_MAPPING). This is enabled only for segment sets at present.
- Implement panel contribution analysis capability to Rayleigh method (*FREQUENCY_DOMAIN_ACOUSTIC_BEM_PANEL_CONTRIBUTION).
- Implement a scheme to map velocity boundary condition from dense BEM mesh to coarse mesh to speed up the computation (*FREQUENCY_DOMAIN_-ACOUSTIC_BEM).
- Add user node ID for acoustic field points in D3ATV (*FREQUENCY_DOMAIN_ACOUSTIC_BEM). Now D3ATV is given for multiple field points, and multiple frequencies.
- Add database D3ATV for acoustic transfer vector binary plot (*FREQUENCY_-DOMAIN_ACOUSTIC_BEM_ATV, *DATABASE_FREQUENCY_BINARY_-D3ATV).
- Implement acoustic panel contribution analysis to collocation BEM and dual collocation BEM (*FREQUENCY_DOMAIN_ACOUSTIC_BEM).
- Enable *FREQUENCY_DOMAIN_MODE in response spectrum analysis (*FREQUENCY_DOMAIN_RESPONSE_SPECTRUM).
- Implement an option to read in user-specified nodal velocity history data for running BEM acoustics (*FREQUENCY_DOMAIN_ACOUSTIC_BEM).

- Extend Kirchhoff acoustic method to MPP (*FREQUENCY_DOMAIN_-ACOUSTIC_BEM).
- Extend response spectrum analysis to multiple load spectra cases (*FREQUENCY_DOMAIN_RESPONSE_SPECTRUM).
- Add BAGVENTPOP for *SENSOR_CONTROL. This allows user more flexibilty controlling the pop-up of the venting hole of *AIRBAG_HYBRID and *AIRBAG_WANG_NEFSKE
- Add command *SENSOR_DEFINE_FUNCTION. Up to 15 *DEFINE_SENSORs can be referenced in defining a mathematical operation.
- LAYER of *SENSOR_DEFINE_ELEMENT can now be an integer "I" representing the Ith integration point at which the stress/strain of the shell or tshell element will be monitored.
- Add control of *LOAD_MOVING_PRESSURE by using *SENSOR_CONTROL.
- Add thick shells to the ETYPE option list of *SENSOR_DEFINE_ELEMENT.
- Add *CONTROL_MPP_MATERIAL_MODEL_DRIVER in order to enable the Material Model Driver for MPP (1 core).
- Add table input of thermal expansion coefficient for *MAT_270. Supports temperature-dependent curves arranged according to maximum temperature.
- Add table input of heat capacity for *MAT_T07. Supports temperature dependent curves arranged according to maximum temperature.
- Add two more kinematic hardening terms for *MAT_DAMAGE_3/MAT_153, c2 & gamma2.
- Add materials *MAT_CONCRETE_DAMAGE_REL3/*MAT_072R3 and *MAT_-CSCM_CONCRETE/*MAT_159 to Interactive Material Model Driver.
- Enable *MAT_JOHNSON_COOK/*MAT_015 for shell elements to work with coupled structural / thermal analysis.
- Allow *MAT_SOIL_AND_FOAM/*MAT_005 to use positive or negative abscissa values forload curve input of volumetric strains.
- Add *MAT_ACOUSTIC elform = 8 support for pyramid element case using 5-pt integration.
- Add support to *MAT_219 (*MAT_CODAM2) for negative AOPT values which point to coordinate system ID's.
- Modify *MAT_224 so it uses the temperatures from the thermal solution for a coupled thermal-mechanical problem.
- Add alternative solution method (Brent) for *MAT_015 and *MAT_157 in case standard iteration fails to converge.

- Add shell element IDs as additional output to messag file for *MAT_036's warning “plasticity algorithm did not converge”.
- For *MAT_USER_DEFINED_MATERIAL_MODELS, the subroutines crvval and tabval can be called with negative curve / table id which will extract values from the user input version of the curve or table instead of the internally converted “100-point” curve / table.
- In the damage initiation and evolution criteria of *MAT_ADD_EROSION (invoked by IDAM < 0), add the option Q1 < 0 for DETYP = 0. Here, $|Q1|$ is the table ID defining the ufp (plastic displacement at failure) as a function of triaxiality and damage value, i.e., ufp = ufp(eta, D), as opposed to being constant which is the default.
- In *MAT_RHT, ONEMPA = -6 generates parameters in g, cm, and μS and ONEMPA = -7 generates parameters in g, mm, and mS
- In *MAT_SIMPLIFIED_RUBBER/FOAM, STOL > 0 invokes a stability analysis and warning messages are issued if an unstable stretch point is found within a logarithmic strain level of 100%.
- Implement *DATABASE_ALE to write time history data (volume fractions, stresses, …) for a set of ALE elements. Not to be confused with *DATABASE_ALE_MAT.
- Implement *DELETE_PART in small restarts for ALE2D parts.
- Add conversion of frictional contact energy into heat when doing a coupled thermal-mechanical problem for SPH (variable FRCENG in *CONTROL_CONTACT). This option applys to all 3D contact types supported by SPH particles.
- For keyword *DEFINE_ADAPTIVE_SOLID_TO_SPH, add support of explicit SPH thermal solver for the newly generated SPH particles which were converted from solid elements. The temperatures of those newly generated SPH particles are mapped from corresponding solid elements.
- Implement DE to surface tied contact *DEFINE_DE_TO_SURFACE_TIED. The implementation includes bending and torsion.
- Implement keyword *DEFINE_DE_HBOND to define heterogeneous bond for discrete element spheres (DES). DES (*ELEMENT_DISCRETE_SPHERE) with different material models can be bonded.
- Implement keyword *INTERFACE_DE_BOND to define multiple failure models for various bonds within one part or between different parts through the keyword *DEFINE_DE_HBOND.
- Implement *DEFINE_DE_TO_BEAM_COUPLING for coupling of discrete element spheres to beam elements.

- Add variable MAXGAP in *DEFINE_DE_BOND to give user control of distance used in judging whether to bond two DES together or not, based on their initial separation.
- Add IAT = -3 in *CONTROL_REMESHING_EFG, which uses FEM remapping scheme in EFG adaptivity. Compared to IAT = -2, -1, 1, 2, IAT = -3 is faster and more robust but less accurate.
- Add control flag MM in *CONTROL_REMESHING_EFG to turn on/off monotonic mesh resizing for EFG 3D general remeshing (ADPOPT = 2 in *PART).
- *CONTROL_IMPLICIT_BUCKLING - Extend Implicit Buckling Feature to allow for Implicit problems using Inertia Relief. This involves adding the Power Method as a solution technology for buckling eigenvalue problems. Using the power method as an option for buckling problems that are not using inertia relief has been added as well.
- Extend Implicit Buckling to allow for Intermittent extraction by using negative values of NMODE on *CONTROL_IMPLICIT_BUCKLING similar to using negative values of NEIG on *CONTROL_IMPLICIT_EIGENVALUE.
- Extend implicit-explicit switching specified on *CONTROL_DYNAMIC_RELAXATION to allow explicit simulation for the dynamic relaxation phase and implicit for the transient phase.
- New implementation for extracting resultant forces due to joints for implicit mechanics.
- New implementation of extracting resultant forces due to prescribed motion for implicit mechanics.
- Add support for IGAP > 2 in implicit, segment based (SOFT = 2) contact.
- Add constraint-based, thermal nodal coupling for *CONSTRAINED_LAGRANGE_IN_SOLID. HMIN < 0 turns it on.
- Add FRCENG = 2 on CONTROL_CONTACT keyword.
    - if FRCENG = 1, convert contact frictional energy to heat.
    - if FRCENG = 2, do not convert contact frictional energy to heat.
- Add effect of thermal time scaling (TSF in *CONTROL_THERMAL_SOLVER) to 2D contact.
- Add new pfile decomposition region option: partsets. Takes a list of part sets (*SET_PART) from the keyword input and uses them to define a region, e.g., region { partsets 102 215 sy 1000 } This example would take partsets, scale y by 1000, and decompose them and distribute them to all processors.
- Reduce MPP memory usage on clusters.
- Add MPP support for *ELEMENT_SOURCE_SINK.

- Add new pfile options:

  - decomp { d2r_as_rigid }
  - decomp { d2ra_as_rigid }

  which cause materials appearing in "*DEFORMABLE_TO_RIGID" and "*DEFORMABLE_TO_RIGID_AUTOMATIC" to have their computational costs set as if they were rigid materials during the decomposition.

- Add option ISRCOUT to *INCLUDE_STAMPED_PART to dump out the transformed source/stamp mesh.
- *CONTROL_FORMING_OUTPUT: Allow NTIMES to be zero; support birth and death time; support scale factor in curve definition.
- Add a new option (INTFOR) to *CONTROL_FORMING_OUTPUT to control the output frequency of the INTFOR database.
- Add new features (instant and progressive lancing) in *ELEMENT_LANCING for sheet metal lancing simulation.
- Add a new keyword: *CONTROL_FORMING_INITIAL_THICKNESS.
- Add a new option for springback compensation: *INCLUDE_COMPENSATION_ORIGINAL_TOOLS.
- Add a new keyword: *INTERFACE_COMPENSATION_NEW_PART_CHANGE.
- Add a new keyword (*DEFINE_CURVE_BOX_ADAPTIVITY) to provide better control of mesh refinement along two sides of the curve.
- Isogeometric analysis: contact is available in MPP.
- Normalize tangent vectors for local coordinate system for the rotation free isogeometric shells.
- Add support for dumping shell internal energy density for isogeometric shells (*ELEMENT_SHELL_NURBS_PATCH) via interpolation shells.
- Add support for dumping of strain tensor (STRFLG.eq.1) for isogeometric shells (*ELEMENT_SHELL_NURBS_PATCH) via interpolation shells.
- Add H-field, magnetization and relative permeability to d3plot output.
- *ICFD_INITIAL: Add a reference pressure (pressurization pressure) for when no pressure is imposed on the boundaries.
- Add the initialization of all nodes at once by setting PID = 0.
- Add the non-inertial reference frame implementation defined by the keyword *ICFD_DEFINE_NONINERTIAL.
- Add several new state variables to LSO. Please refer to the LSO manual to see how to print out the list of supported variables.

- Add support for FSI with thick shells.
- 2D shells are now supported for FSI in MPP. In the past only beams could be used in MPP and beams and shells could be used in SMP.
- The keyword ICFD_CONTROL_FSI has a new field to control the sensitivity of the algorithm to find the solid boundaries used in FSI calculations.
- The 2D mesh now generates semi-structured meshes near the boundaries.
- Add heat flux boundary condition using ICFD_BOUNDARY_FLUX_TEMP.
- Add divergence-free and Space Correlated Synthetic Turbulence Inlet Boundary Condition for LES (Smirnov et al.) using *ICFD_BOUNDARY_PRESCRIBED_-VEL.
- *ICFD_BOUNDARY_PRESCRIBED_VEL: Add inflow velocities using the wall normal and a velocity magnitude using the 3rd field VAD.
- Add the activation of synthetic turbulence using the 3rd field VAD.
- Add the option to control the re-meshing frequency in both keywords: see *ICFD_CONTROL_ADAPT_SIZE and *ICFD_CONTROL_ADAPT.
- *ICFD_CONTROL_TURB_SYNTHESIS: control parameters for the synthetic turbulence inflow.
- *ICFD_BOUNDARY_PRESCRIBED_MOVEMESH: Allows the mesh to slide on the boundaries following the cartesian axis.
- Add a PART_SET option for *CESE_BOUNDARY_…_PART cards.
- Bring in more 2D mesh support, both from the PFEM mesher and a user input 2D mesh (via *ELEMENT_SOLID with 0 for the last 4 of 8 nodes).
- Enable the 2D ball-vertex mesh motion solver for the 2D CESE solver.
- Add new input cards:
    - *CESE_BOUNDARY_CYCLIC_SET
    - *CESE_BOUNDARY_CYCLIC_PART
- Add code for 2D CESE sliding boundary conditions.
- Add support in CESE FSI for 2D shells in MPP.
- Add support for CESE FSI with thick shells.
- Add 2D & 2D-axisymmetric cases in the CESE-FSI solver (including both immersed boundary method & moving mesh method) .
- Add the CSP reduced chemistry model with 0D, 2D, and 3D combustion. The 2D and 3D combustion cases couple with the CESE compressible flow solver.
- Add the G-scheme reduced chemistry model only for 0D combustion.
- Add two different reduced chemistry models.

  - ◦ The Computational Singular Perturbation (CSP) reduced model is implemented with existing compressible CESE solver. The CSP is now working on 0-dimensional onstant volume and pressure combustion, 2-D, and 3-D combustion problems.
  - ◦ The new reduced chemistry model, G-scheme, is implemented, but currently works only 0-dimensional problems such as constant combustors.

- Jobid can now be changed in a restart by including "jobid=" on the restart execution line. Previously, the jobid stored in d3dump could not be overwritten.
- Part labels (PID) can be up to 8 characters in standard format; 20 characters in long format.
- Labels for sections (SID), materials (MID), equations of state (EOSID), hourglass IDs (HGID), and thermal materials (TMID) can be up to 10 characters in standard format; 20 characters in long format.
- Create bg_switch and kill_by_pid for SMP. Both files will be removed at the termination of the run.
- Increase the overall length of command line to 1000 characters and length of each command line option to 50 characters.
- Increase MPP search distance for tied contacts to include slave and master thicknesses.
- For *CONTACT_AUTOMATIC_…_MORTAR, the mortar contact now supports contact with the lateral surface of beam elements.
- On *CONTACT_...._MORTAR, IGAP.GT.1 stiffens the mortar contact for large penetrations. The mortar contact has a maximum penetration depth DMAX that depends on geometry and input parameters; if penetration is larger than this value the contact is released. To prevent this release, which is unwanted, the user may put IGAP.GT.1 which stiffens the behavior for penetrations larger than 0.5*DMAX without changing the behavior for small penetrations. This should hopefully not be as detrimental to convergence as increasing the overall contact stiffness.
- For initialization by prescribed geometry in dynamic relaxation (IDRFLG = 2, *CONTROL_DYNAMIC_RELAXATION), add an option where displacements are not imposed linearly but rather according to a polar coordinate system. This option was added to accommodate large rotations.
- The flag RBSMS on *CONTROL_RIGID is now active for regular and selective mass scaling to consistently treat interfaces between rigid and deformable bodies
- Remove static linking for l2a as many systems do not have the required static libraries.
- Add IELOUT in *CONTROL_REFINE to handle how child element data is handled in elout (*DATABASE_HISTORY_SOLID and *DATABASE_HISTO-

RY_SHELL). Child element data are stored if IELOUT = 1 or if refinement is set to occur only during initialization.

- Include eroded hourglass energy in hourglass energy in glstat file to be consistent with KE & IE calculations so that the total energy = kinetic energy + internal energy + hourglass energy + rigidwall energy.
- Remove *DATABASE_BINARY_XTFILE since it is obsolete.
- When using *PART_AVERAGED for truss elements (beam formulation 3), calculate the time step based on the total length of the combined macro-element instead of the individual lengths of each element.
- Enable writing of midside nodes to d3plot or 6- and 8-node quadratic shell elements.
- Write complete history variables to dynain file for 2D solids using *MAT_NULL and equation-of-state.
- Shell formulations 25, 26, and 27 are now fully supported in writing to dynain file (*INTERFACE_SPRINGBACK_LSDYNA).
- Shell formulations 23 (quad) and 24 (triangle) can now be mixed in a single part. When ESORT = 1 in *CONTROL_SHELL, triangular shells assigned by *SECTION_SHELL to be type 23 will automatically be changed to type 24.
- Enable hyperelastic materials (those that use Green's strain) to be used with thick shell form 5. Previously, use of these materials (2, 7, 21, 23, 27, 30, 31, 38, 40, 112, 128, 168, and 189) with thick shell 5 has been an input error.
- Update acoustic BEM to allow using *DEFINE_CURVE to define the output frequencies (*FREQUENCY_DOMAIN_ACOUSTIC_BEM).
- When using *CONTROL_SPOTWELD_BEAM, convert *DATABASE_HISTORY_BEAM to *DATABASE_CROSS_SECTION and *INITIAL_AXIAL_FORCE_BEAM to *INITIAL_STRESS_CROSS_SECTION for the spotweld beams that are converted to hex spotwelds.
- Improve output of *INITIAL_STRESS_BEAM data to dynain via *INTERFACE_SPRINGBACK_LSDYNA. Now, large format can be chosen, history variables are written, and local axes vectors are included.
- Update *MAT_214 (*MAT_DRY_FABRIC) to allow fibers to rotate independently.
- Enable regularization curve LCREGD of *MAT_ADD_EROSION to be used with FLD criterion, i.e. load curve LCFLD. Ordinate values (major strain) will be scaled with the regularization factor.
- Modify *MAT_ADD_EROSION parameter EPSTHIN:
  - EPSTHIN > 0: individual thinning for each IP from z-strain (as before).
  - EPSTHIN < 0: averaged thinning strain from element thickness (new).

- Enable regularization curve LCREGD of *MAT_ADD_EROSION to be used with standard (non-GISSMO) failure criteria. Users can now define a failure criterion plus IDAM = 0 plus LCREGD = scaling factor vs. element size to get a regularized failure criterion.
- *MAT_ADD_EROSION: equivalent von Mises stress SIGVM can now be a function of strain rate by specifying a negative load curve ID.
- *SECTION_ALE1D and *SECTION_ALE2D now work on multiple processors (SMP and MPP).
- *CONSTRAINED_LAGRANGE_IN_SOLD ctype 4/5 now converts friction energy to heat. Note it only works for ALE elform 12.

Capabilities added September 2013 – January 2015 to create LS-DYNA R8.0:

See release notes (published separately) for further details.

- Add RDT option for *AIRBAG_SHELL_REFERENCE_GEOMETRY.
- LCIDM and LCIDT of *AIRBAG_HYDRID can now be defined through *DEFINE_CURVE_FUNCTION.
- New variable RGBRTH in *MAT_FABRIC to input part-dependent activation time for airbag reference geometry.
- Negative PID of *AIRBAG_INTERACTION considers the blockage of partition area due to contact.
- Enhancements to *AIRBAG_PARTICLE:
  - New blockage (IBLOCK) option for vents.
  - External work done by inflator gas to the structure is reported to glstat.
  - Enhance segment orientation checking of CPM bag and chambers.
  - Allow user to excluded some parts surface for initial air particles.
  - Support compressing seal vent which acts like flap vent.
  - Support Anagonye and Wang porosity equation through *MAT_FABRIC.
  - Add keyword option _MOLEFRACTION.
- Add_ID keyword option to *AIRBAG_REFERENCE_GEOMETRY and *AIRBAG_SHELL_REFERENCE which includes optional input of variables for scaling the reference geometry.
- Enable *DEFINE_CURVE_FUNCTION for *AIRBAG_SIMPLE_AIRBAG_MODEL.
- Calculate heat convection (HCONV) between environment and airbag in consistent fashion when TSW is used to switch from a particle airbag to a control volume.

- For *AIRBAG_PARTICLE, add ENH_V = 2 option for vent hole such that two-way flow can occur, i.e., flow with or against the pressure gradient.
- *BOUNDARY_ALE_MAPPING: add the following mappings: 1D to 2D, 2D to 2D, 3D to 3D.
- *SET_POROUS_ALE: new keyword to define the properties of an ALE porous media by an element set. The porous forces are computed by *LOAD_BODY_POROUS.
- *ALE_FSI_SWITCH_MMG: applies also now to 2D.
- *ALE_SWITCH_MMG: new keyword to switch multi-material groups based on criteria defined by the user with *DEFINE_FUNCTION.
- *CONTROL_ALE: Allow PREF (reference pressure) to be defined by materials.
- Implement *ALE_COUPLING_NODAL_DRAG to model the drag force coupling between discrete element spheres or SPH particles and ALE fluids.
- Implement *ALE_COUPLING_RIGID_BODY as an efficient alternative for constraint type coupling between ALE fluids and a Lagrangian rigid body.
- Error terminate if *BOUNDARY_SPC_NODE_BIRTH_DEATH is applied to a node that belongs to a rigid body.
- Modify *BOUNDARY_PRESCRIBED_ORIENTATION_VECTOR to accommodate bodies which undergo no changes in orientation.
- Add a new keyword *BOUNDARY_SPC_SYMMETRY_PLANE.
- Solid part or solid part set is now allowed for *PARTICLE_BLAST.
- Add ambient pressure boundary condition flag BC_P for *PARTICLE_BLAST.
- New command *DEFINE_PBLAST_GEOMETRY allows the high explosive domain for*PARTICLE_BLAST to be defined by various geometric shapes.
- Allow multiple *PARTICLE_BLAST definitions.
- Add *DATABASE_PBSTAT to output particle blast statistics.
- Output the initial volume and initial mass of HE particles and air particles for *PARTICLE_BLAST to d3hsp.
- Add the command *CESE_BOUNDARY_BLAST_LOAD to allow a blast described by the *LOAD_BLAST_ENHANCED command to be used as a boundary condition in CESE.
- Modify the FSI interface reflective boundary condition pressure treatment in some calculations for the moving mesh and immersed boundary solvers.
- Change the CESE derivatives calculation method to use the current values of flow variables.

- Add two new MAT commands for CESE solver, *CESE_MAT_000 and *CESE_-MAT_002.
- Add a non-inertial reference frame solver for fluid and FSI problems using the moving-mesh method.
- For the moving mesh CESE solver, replace the all-to-all communication for conjugate heat and FSI quantities with a sparse communication mechanism.
- Add structural element erosion capability to the immersed boundary method CESE FSI solver (serial capability only).
- Add 2D cyclic boundary conditions capability.
- Add a NaN detection capability for the CESE solver.
- Switch all CESE boundary conditions that use a mesh surface part to define the boundary to use the character string "MSURF" instead of "PART" in the option portion of the keyword name.
- Add missing temperature interpolation in time for imposing solid temperatures as a boundary condition in the CESE solver.
- Optimize the IDW-based mesh motion for the CESE moving mesh solver.
- Treat the input mesh as 3D by default for the CESE solver.
- All of the chemistry features mentioned below are coupled only to the CESE compressible flow solver when 2D or 3D calculations are involved.
- Chemical source Jacobians have been added.
- Introduce *CHEMISTRY_CONTROL_PYROTECHNIC and *CHEMISTRY_PROPELLANT_PROPERTIES for airbag applications. In conjuction with these commands, basic airbag inflator models are implemented.
- The pyrotechnic inflator model using NaN3/Fe2O3 propellant is newly implemented. To connect with the existing ALE airbag solver, two load curves, mass flow rate and temperature, are saved in "inflator_outfile" as a function of time. This model computes three sub-regions: combustion chamber, gas plenum, and discharge tank. Each region can be initialized with different *CHEMISTRY_-COMPOSITION models, which means that user can compute Propellant+Gas hybrid mode.
- The following 0-dimensional combustion problems have been improved: constant volume, constant pressure, and CSP.
- For iso-combustion. temperature and species mass fractions as a function of time are displayed on screen and saved in "isocom.csv" to plot with LS-PrePost.
- Another chemical ODE integration method has been implemented.
- The output file of the pyrotechnic inflator is updated so that this file can be read from ALE solver for an airbag simulation.

- 2-D and 3-D TNT gaseous blast explosives, categorized as TBX (thermobaric explosives), are implemented for the Euler equation systems (CESE-only). Also, 3-D TNT blast + aluminum combustion for serial problems is now implemented.
- Implement a mix modeling method for use with CESE solvers.
- Modify *CHEMISTRY-related keyword commands to allow multiple chemistry models in the same problem.
- Add command *CHEMISTRY_MODEL which identifies the files that define a Chemkin chemistry model.
- Modify the following commands such that the files related to the chemistry model have been removed. These commands are only used to select the type of chemistry solver:
  - *CHEMISTRY_CONTROL_CSP
  - *CHEMISTRY_CONTROL_FULL
  - *CHEMISTRY_CONTROL_1D
- Modify *CHEMISTRY_DET_INITIATION where the files related to the chemistry model have been removed, and the Model ID used is inferred through a reference to a chemistry composition ID.
- Modify *CHEMISTRY_COMPOSITION and *CESE_CHEMISTRY_D3PLOT to add model ID.
- Add *CONTACT_TIED_SHELL_EDGE_TO_SOLID for transferring moments from shells into solids.
- Add frictional energy calculation for beams in *CONTACT_AUTOMATIC_GENERAL.
- Enhance ERODING contacts for MPP. The new algorithm uses a completely different approach to determining the contact surface. The old algorithm started from scratch when identifying the exterior of the parts in contact. The new algorithm is smarter about knowing what has been exposed based on what is eroded, and is faster.
- Force EROSOP = 1 for all ERODING type contacts, with a warning to the user if they had input it as 0.
- Add error check in case of a contact definition with an empty node set being given for the slave side.
- Modify output of ncforc (*DATABASE_NCFORC) in order to support output in a local coordinate system.
- For ERODING contacts, reduce memory allocated for segments so each interior segment is only allocated once.
- Add keyword *DEFINE_CONTACT_EXCLUSION (MPP only) to allow for nodes tied in some contacts to be ignored in certain other contacts.

- Rewrite meshing of *CONTACT_ENTITY to use dynamic memory, which removes the previous limit of 100 meshed contact entities. There is now no limit.
- Remove undocumented release condition for MPP's *CONTACT_AUTOMATIC_TIEBREAK, options 5 and greater.
- Add new experimental "square edge" option to select SOFT = 0,1 contacts. This new option applies only to AUTOMATIC_SINGLE_SURFACE and the segment-to-segment treatment of AUTOMATIC_GENERAL, and is invoked by setting SRNDE = 2 on *CONTACT's Optional Card E. This new option does not apply to SOFT = 2; SOFT = 2 square edge option is set using SHLEDG in *CONTROL_CONTACT.
- BT and DT in *CONTACT can be set to define more than one pair of birthtime/death-time for the contact by pointing to a curve or table. These pairs can be unique for the dynamic relaxation phase and the normal phase of the simulation.
- Add EDGEONLY option to *CONTACT_AUTOMATIC_GENERAL to exclude node-to-segment contact and consider only edge-to-edge and beam-to-beam contact.
- VDC defines the coefficient of restituion when variable CORTYP is defined. Available for *CONTACT_AUTOMATIC_NODES_TO_SURFACE, *CONTACT_AUTOMATIC_SURFACE_TO_SURFACE, and *CONTACT_AUTOMATIC_SINGLE_SURFACE; SOFT = 0 or 1 only.
- Enhancements for *CONTACT_AUTOMATIC_GENERAL:
  - Add beam to beam contact option CPARM8 in *PART_CONTACT (MPP only).
  - Add option whereby beam generated on exterior shell edge will be shifted into the shell by half the shell thickness. In this way, the shell-edge-to-shell-edge contact starts right at the shell edge and not at an extension of the shell edge (see OPT2 = 10, CPARM8, MPP Card 1).
- Implement *CONTROL_CONTACT PENOPT = 3 option to *CONTACT_AUTOMATIC_NODES_TO_SURFACE and *CONTACT_ERODING_NODES_TO_SURFACE for SMP.
- Update segment based (SOFT = 2) contact to improve accuracy at points away from the origin. The final calculations are now done with nodal and segment locations that have been shifted towards the origin so that coordinate values are small.
- Enable user defined friction (*USER_INTERFACE_FRICTION; subroutine usrfrc) for MPP contact SOFT = 4.

- Unify automatic tiebreak messages for damage start and final failure. SMP and MPP should now give the same output to d3hsp and messag. This affects *CONTACT_AUTOMATIC_...TIEBREAK, OPTIONs 6, 7, 8, 9, 10, and 11.
- *CONTACT_ADD_WEAR: Associates wear calculations to a forming contact interface whose quantities can be posted in the intfor database file. Adaptivity is supported.
- *CONTACT_..._MORTAR:
  - Detailed warning outputs activated for mortar contact, also clarifies echoed data in d3hsp.
  - Contact thickness made consistent with other contacts in terms of priority between ISTUPD on CONTROL_SHELL, SST on CONTACT and OPTT on PART_CONTACT.
  - Efficency improvement of bucket sort in mortar contact allowing for significant speedup in large scale contact simulations.
- *CONTACT_..._MORTAR, *DEFINE_FRICTION, *PART_CONTACT:
  - Mortar contact supports FS = -1.0, meaning that frictional coefficients are taken from *PART_CONTACT parameters.
  - Mortar contact supports FS.EQ.-2 meaning that friction is taken from *DEFINE_FRICTION.
- *CONTACT_AUTOMATIC_SINGLE_SURFACE_MORTAR: Using IGNORE.LT.0 for single surface mortar contact will ignore penetrations of segments that belong to the same part.
- Friction factors are now a function of temperature for *CONTACT_..._THERMAL_FRICTION.
- *SET_POROUS_LAGRANGIAN: new keyword to define the porosity of Lagrangian elements in an element set. The porous forces are computed by *CONSTRAINED_LAGRANGE_IN_SOLID ctype = 11 or 12.
- *CONSTRAINED_LAGRANGE_IN_SOLID: CTYPE = 12 is now also available in 2D.
- Add helix angle option for *CONSTRAINED_JOINT_GEARS.
- Change keyword from *CONSTRAINED_BEARING to *ELEMENT_BEARING.
- Enhance explicit to use the implicit inertia relief constraints. This allows implicit-explicit switching for such problems.
- Add new input options to *CONTROL_IMPLICIT_INERTIA_RELIEF.
  - user specified number of nodes
  - user specified list of modes to constrain out.

- Implement *CONSTRAINED_BEAM_IN_SOLID. This feature is basically an overhauled constraint couping between beams and Lagrangian solids that includes features that make it more attractive in some cases than *CONSTRAINED_LAGRANGE_IN_SOLID, for example, in modeling coupling of rebar in concrete.
- Allow *CONSTRAINED_INTERPOLATION to use node set to define the independent nodes.
- Add new feature MODEL.GE.10 to *CONSTRAINED_INTERPOLATION_-SPOTWELD (SPR3). This allows parameters STIFF, ALPHA1, RN, RS, and BETA to be defined as *DEFINE_FUNCTIONs of thicknesses and maximum engineering yield stresses of connected sheets.
- Add failure reports for *CONSTRAINED_SPR2.
- Add more d3hsp output for *CONSTRAINED_INTERPOLATION_SPOTWELD and *CONSTRAINED_SPR2. Can be deactivated by setting NPOPT = 1 on *CONTROL_OUTPUT.
- Add option to *CONSTRAINED_JOINT: Relative penalty stiffness can now be defined as function of time when RPS < 0 refers to a load curve. Works for SPHERICAL, REVOLUTE, CYLINDRICAL in explixit analyses.
- Variable MODEL invokes new SPR4 option in *CONSTRAINED_INTERPOLATION_SPOTWELD.
- *CONSTRAINED_JOINT_GEARS: Gear joint now supports bevel gears and similar types, i.e., the contact point does not necessarily have to be on the axis between the gear centers.
- *CONSTRAINED_MULTIPLE_GLOBAL: Support multiple constraints defined on the extra DOFs of user-defined elements.
- Make the *CONTROL_SHELL PSNFAIL option work with the W-MODE deletion criterion for shells.
- New subcycling scheme activated for *CONTROL_SUBCYCLE and *CONTROL_SUBCYCLE_MASS_SCALED_PART. By default the ratio between the largest and smallest time step is now 16 and the external forces are evaluated every time step. The old scheme had a hard wired ratio of 8. The ratios can be optionally changed by *CONTROL_SUBCYCLE_K_L where K is the maximum ratio between time steps for internal forces and L is likewise the ratio for external forces.
- *DATABASE_PROFILE:
  - output kinetic and internal energy profiles,
  - output volume fraction profiles,
  - add a parameter MMG to specify the ALE group for which element data can be output.

- *DATABASE_ALE_MAT: can now use *DEFINE_BOX to compute the material energies, volumes and masses for elements inside boxes (instead of the whole mesh).
- *DATABASE_TRACER_GENERATE: new keyword to create ALE tracer particles along iso-surfaces.
- *DATABASE_FSI: add option to output moments created by FSI forces about each node in a node set. These moments about nodes are reported in dbfsi.
- Add *DATABASE_BEARING to write brngout data pertaining to *ELEMENT_-BEARINGs.
- Include eroded hourglass energy in hourglass energy in glstat file to be consistent with KE & IE calculations so that the total energy = kinetic energy + internal energy + hourglass energy + rigidwall energy.
- Add support for new database pbstat (*DATABASE_PBSTAT) for *PARTICLE_-BLAST.
    - internal energy and translational energy of air and detonation products
    - force/pressure of air and detonation products for each part
- *DATABASE_EXTENT_INTFOR: New parameter NWEAR on optional card governs the output of wear depth to the intfor database.
- Using CMPFLG = -1 in *DATABASE_EXTENT_BINARY will work just as CMPFLG = 1, except that for *MAT_FABRIC (form 14 and form -14) and *MAT_-FABRIC_MAP the local strains and stresses will be engineering quantities instead of Green-Lagrange strain and 2nd Piola-Kirchhoff stress.
- For some materials and elements, thermal and plastic strain tensors can be output to d3plot database, see STRFLG in *DATABASE_EXTENT_BINARY.
- Add option for output of detailed (or long) warning/error messages to d3msg. See MSGFLG in *CONTROL_OUTPUT. Only a few "long" versions of warnings/errors at this time but that list is expected to grow.
- Add two new options for rigid body data compression in d3plot; see DCOMP in *DATABASE_EXTENT_BINARY.
- Add option to write revised legend to jntforc, secforc, rcforc, deforc and nodout files via input flag NEWLEG in *CONTROL_OUTPUT. This helps to avoid confusion over unassigned IDs and duplicated IDs.
- If any input data is encrypted and dynain is requested, the code issues an error message and stops the job.
- Solid part or solid part set is now allowed for *DEFINE_DE_TO_SURFACE_-COUPLING.
- Implement *DELETE_PART for Discrete Element Sphere.

- The unit of contact angle changed from radian to degree for *CONTROL_DISCRETE_ELEMENT.
- Implement Archard's wear law to *DEFINE_DE_TO_SURFACE_COUPLING for discrete element spheres. Wear factor is output to DEM binout database.
- Add damping energy and frictional energy of discrete elements to "damping energy" and "sliding interface energy" terms in glstat.
- Introduce a small perturbation to the initial position of newly generated discrete elements for *DEFINE_DE_INJECTION. This allows a more random spatial distribution of the generated particles.
- *INTERFACE_DE_HBOND replaces *INTERFACE_DE_BOND. Used to define the failure models for bonds linking various discrete element (DE) parts within one heterogeneous bond definition (*DEFINE_DE_HBOND).
- *DEFINE_ADAPTIVE_SOLID_TO_DES: Embed and/or transform failed solid elements to DES (*ELEMENT_DISCRETE_SPHERE) particles. The DES particles inherit the material properties of the solid elements. All DES-based features are available through this transformation, including the bond models and contact algorithms. This command is essentially to DES what *DEFINE_ADAPTIVE_-SOLID_TO_SPH is to SPH particles.
- Add EM orthotropic materials where the electric conductivity is a 3x3 tensor, see new card, *EM_MAT_003.
- Add new keyword family, *EM_DATABASE_... which triggers the output of EM quantites and variables. All EM related ASCII outputs now start with em_***. Keywords are :

  - EM_DATABASE_CIRCUIT
  - EM_DATABASE_CIRCUIT0D
  - EM_DATABASE_ELOUT
  - EM_DATABASE_GLOBALENERGY
  - EM_DATABASE_NODOUT
  - EM_DATABASE_PARTDATA
  - EM_DATABASE_POINTOUT
  - EM_DATABASE_ROGO
  - EM_DATABASE_TIMESTEP

- Add capability to plot magnetic field lines in and around the conductors at given times, see *EM_DATABASE_FIELDLINE. ASCII output files are generated (lspp_fieldLine_xx) and are readable by LSPP in order to plot the field lines. In the future, LSPP will be capable of directly generating the field lines.
- Add EM quantities in *DEFINE_CURVE_FUNCTION:

  - EM_ELHIST for element history (at element center).
  - EM_NDHIST for node history.

    - EM_PAHIST for part history (integrated over the part).

- Add *EM_EOS_TABULATED2 where a load curve defines the electrical conductivity vs time.
- Introduce capability to use the EM solver on (thin) shells: An underlying solid mesh (hexes and prisms) is built where the EM is solved and the EM fields are then collapsed onto the corresponding shell. The EM mat for shells is defined in *EM_MAT_004. This works for EM solvers 1, 2 and 3 and the EM contact is available for shells.
- Add different contact options in the *EM_CONTACT card.
- Add new methods to calculate electric contact resistance between two conductors for Resistive Spot Welding applications (RSW). See *EM_CONTACT_RESISTANCE.
- Add Joule Heating in the contact resistance (*EM_CONTACT_RESISTANCE). The Joule heating is evenly spread between the elements adjacent to the faces in contact.
- Add new circuit types 21 and 22 (see *EM_CIRCUIT) allowing users to put in their own periodic curve shape when using the inductive heating solver. This is useful in cases where the current is not a perfect sinusoidal.
- Provide default values for NCYCLEBEM and NCYCLEFEM (=5000) and set default value of NUMLS to 100 in *EM_CIRCUIT.
- Add two additional formulations, FORM = 3 and 4, to *PART_MODES.
- Add 20-node solid element, ELFORM = 23 in *SECTION_SOLID.
- Add H8TOH20 option to *ELEMENT_SOLID to convert 8-node to 20-node solids.
- Add option SOLSIG to *CONTROL_OUTPUT which will permit stresses and other history variables for multi-integration point solids to be extrapolated to nodes. These extrapolated nodal values replace the integration point values normally stored in d3plot. NINTSLD must be set to 8 in *DATABASE_EXTENT_BINARY when a nonzero SOLSIG is specified. Supported solid formulations are solid elements are: -1, -2, 2, 3, 4, 18, 16, 17, 23.
- Activate contact thickness input from *PART_CONTACT for solids.
- Made many enhancements for *PART_MODES for robustness and MPP implementation.
- Add new cohesive shell element (elform = 29) for edge-to-edge connectivity between shells. This element type takes bending into account and supports MPP and implicit solvers.
- Error terminate with message, STR+1296, if same node is defined multiple times in *ELEMENT_MASS_MATRIX.

- Add support for negative MAXINT option in *DATABASE_EXTENT_BINARY for thick shell elements.
- *ELEMENT_TSHELL: Add "BETA" as option for *ELEMENT_TSHELL to provide an orthotropic material angle for the element.
- Add Rayleigh damping (*DAMPING_PART_STIFFNESS) for triangular shell element types 3 and 17.
- Add new keyword *ELEMENT_BEAM_SOURCE. Purpose: Define a nodal source for beam elements. This feature is implemented for truss beam elements (ELFORM = 3) with material *MAT_001 and for discrete beam elements (ELFORM = 6) with material *MAT_071.
- Add new option to *DEFINE_ELEMENT_DEATH. New variable IDGRP defines a group id for simultaneous deletion of elements.
- Convert cohesive solid type 20 and 22 to incremental formulation to properly handle large rotations. Also use consistent mass.
- Add Smoothed Particle Galerkin (SPG) method for solid analysis (ELFORM = 47) and corresponding keyword option *SECTION_SOLID_SPG. SPG is a true particle method in Galerkin formulation that is suitable for severe deformation problems and damage analysis.
- Enhance *ELEMENT_LANCING by supporting *PARAMETER, *PARAMETER_EXPRESSION.
- Add a new feature, *CONTROL_FORMING_TRIMMING, for 2D and 3D trimming of a 3-layer, sandwich laminate blank via *DEFINE_CURVE_TRIM.
- Add 3D normal trimming of solid elements via *DEFINE_CURVE_TRIM_3D.
- Add new features for solid elements 2D trimming *DEFINE_CURVE_TRIM_NEW:
  - Allow support of arbitrary trimming vector (previously only global z direction was allowed).
  - Improve trimming algorithm for speed up.
  - Allow trimming curves to project to either the top or bottom surface.
- Add a new AUTO_CONSTRAINT option to *CONTROL_FORMING_ONESTEP which is convenient for blank nesting.
- Add new features to *CONTROL_FORMING_SCRAP_FALL. Previously the user was required to define the trimmed blank properly. Now the blank is trimmed by the cutting edge of the trim steel, which is defined by a node set and a moving vector.
- Enhance *CONTROL_FORMING_SCRAP_FALL: Allow the node set (NDSET) on the trim steel edge to be defined in any order.
- Improve *CONTROL_FORMING_ONESTEP:

    - Reposition the initial part before unfolding, using the center element normal.
    - Add a message showing that the initial unfolding is in process.

- Add 2D trimming for solid elements *DEFINE_CURVE_TRIM_NEW, support *DEFINE_TRIM_SEED_POINT_COORDINATES.
- Add *CONTROL_FORMING_AUTOCHECK to detect and fix flaws in the mesh for the rigid body that models the tooling.
- Add new features to *CONTROL_FORMING_UNFLANGING:
    - The incoming flange mesh will be automatically checked for mesh quality and bad elements fixed.
    - Allow thickness offset of deformable flange to use the blank thickness from user's input.
    - Allow definition any node ID in the outer boundary of the flange, to speed up the search when holes are present in the part.
    - Add a new parameter CHARLEN to limit the search region.
    - Allow holes to exist in the flange regions.
    - Output a suggested flange part after unflanging simulation, with the failed elements deleted from the unflanged part.
    - Automatically define a node set and constraints for the flange boundary nodes through the user definition of three nodes.
    - Add output of forming thickness, effective strain and trim curves after unflanging simulation.
- Add a new keyword *CONTROL_FORMING_TRIM to replace *ELEMENT_-TRIM.
- Add a new keyword: *CONTROL_FORMING_UNFLANGING_OUTPUT: Failed elements are removed to come up with the trim curves.
- Add new features to *INTERFACE_BLANKSIZE_DEVELOPMENT including allowing for trimming between initial and final blank.
- Enhance *CONTROL_FORMING_OUTPUT for controlling the number of states.
- Add *CONTROL_FORMING_TRIM_MERGE to close a user specified (gap) value in the trim curves, so each trim curve will form a closed loop, which is required for a successful trimming.
- Add *CONTROL_FORMING_MAXID to set a maximum node ID and element ID for the incoming dynain file (typically the blank) in the current simulation.
- Enhance *FREQUENCY_DOMAIN_ACOUSTIC_BEM:
    - Update the boundary condition definition for BEM acoustics so that impedance and other user defined boundary conditions can be combined with time domain velocity boundary condition.

  - Implement Burton-Miller BEM to MPP.
  - Implement impedance boundary condition to Burton-Miller BEM.
  - Implement half space option (*FREQUENCY_DOMAIN_ACOUSTIC_BEM_HALF_SPACE) to variational indirect BEM.
  - Implement half space option to acoustic scattering problems.
  - Extend acoustic ATV computation to elements, in addition to nodes.
  - Support element based ATV output in d3atv.
  - Add an option (_MATV) to run modal acoustic transfer vector. Implement MATV to MPP.
  - Implement running BEM Acoustics based on modal ATV (SSD excitation only).

- *FREQUENCY_DOMAIN_ACOUSTIC_FEM: Enable running FEM acoustics based on restarting SSD (*FREQUENCY_DOMAIN_SSD).
- Add *FREQUENCY_DOMAIN_ACOUSTIC_INCIDENT_WAVE to define the incident waves for acoustic scattering problems. To be used with *FREQUENCY_DOMAIN_ACOUSTIC_BEM.
- Add *FREQUENCY_DOMAIN_ACOUSTIC_SOUND_SPEED to define frequency dependent complex sound speed, which can be used in BEM acoustics. By using complex sound speed, the damping in the acoustic system can be considered. To be used with *FREQUENCY_DOMAIN_ACOUSTIC_BEM.
- *FREQUENCY_DOMAIN_FRF: Add mode dependent rayleigh damping to frf and ssd (DMPMAS and DMPSTF).
- *FREQUENCY_DOMAIN_RESPONSE_SPECTRUM:
  - Add output of nodout_spcm and elout_spcm, to get nodal results and element results at user specified nodes and elements.
  - Add von Mises stress computation.
- *FREQUENCY_DOMAIN_RANDOM_VIBRATION: Add semi-log, and linear-linear interpolation on PSD curves (parameter LDFLAG).
- *FREQUENCY_DOMAIN_SSD:
  - Add strain computation.
  - Add parameter LC3 to define the duration of excitation for each frequency.
  - Implement fatigue analysis option (_FATIGUE) based on ssd (sine sweep).
  - Add option to use *DAMPING_PART_MASS and *DAMPING_PART_STIFFNESS in SSD (DMPFLG = 1).
- Add *MAT_ADD_FATIGUE to define material's SN fatigue curve for application in vibration fatigue and SSD fatigue analysis.
- Add *FREQUENCY_DOMAIN_ACCELERATION_UNIT to facilitate the acceleration unit conversion.

- The icfd_mstats.dat file now outputs the ten worst quality element locations (ICFD solver).
- Add option in *ICFD_CONTROL_OUTPUT allowing terminal output to be written to messag file.
- Add keyword *ICFD_CONTROL_OUTPUT_SUBDOM to output only part of the domain. Available for vtk, dx and gmv formats.
- Add new keyword family, *ICFD_DATABASE_... which triggers the output of ICFD variables. All ICFD related output files now start with icfd_***.
- Add new keyword family *ICFD_SOLVER_TOL_... which allows the user to control tolerances and iteration number for the fractional step solve, the mesh movement solve, and the heat equation solve.
- Curves in *ICFD_BOUNDARY_PRESCRIBED_VEL each provide a scaling factor vs. x,y, or z coordinate, respectively. These scaling factors are applied to the velocity boundary condition.
- Enable free-slip condition for FSI walls (ICFD solver).
- Add new variable IDC to *ICFD_CONTROL_FSI that allows the modification of the scaling parameter that multiplies the mesh size to detect contact.
- Add automatic squeezing to the ICFD elements of the boundary layer when there are two very close surfaces with poor (coarse) mesh resolution.
- Add the initialization for all nodes using *ICFD_INITIAL with PID = 0.
- Add a curve (LCIDSF in *ICFD_CONTROL_TIME) that scales the CFL number as a function of time.
- Add a Heaviside function that allows the solution of simple multiphase problems (ICFD).
- Add the computation of the heat convection coefficient (ICFD).
- Add MPP support for y+ and shear for output (ICFD).
- Add uniformity index (ICFD).
- Add *ICFD_CONTROL_TAVERAGE to control the restarting time for computing the time average values.
- Implement the XMl format for vtk. See *ICFD_CONTROL_OUTPUT.
- Improve temperature stabilization for thermal problems (ICFD).
- Add the Generalized Flow Through Porous Media model monolithically coupled to the incompressible Navier-Stokes model. See keyword *ICFD_MAT for the new options.
- Add the Anisotropic version of the Generalized Flow in Porous Media. See *ICFD_MAT for details.

- Add the capability to define the porous properties using the Pressure-Velocity (P-V) experimental curves. See *ICFD_MAT.
- Compute drag forces around anisotropic/isotropic porous domains (ICFD).
- Extend implicit debug checking when LPRINT = 3 on *CONTROL_IMPLICIT_SOLVER.
- Add option for implicit dynamic relaxation so that only a subset of parts is active during the dynamic relaxation phase.
- Extend implicit time step control via IAUTO < 0 in *CONTROL_IMPLICIT_AUTO to linear analysis.
- Add self piercing rivet capability to implicit (*CONSTRAINED_SPR2, *CONSTRAINED_INTERPOLATION_SPOTWELD).
- Add MTXDMP in *CONTROL_IMPLICIT_SOLVER to dump the damping matrix from implicit mechanics.
- Improve stress and strain computation induced by mode shapes. See MSTRES in *CONTROL_IMPLICIT_EIGENVALUE.
- Add variable MSTRSCL to *CONTROL_IMPLICIT_EIGENVALUE for user control of geometry scaling for the stress computation.
- Make SMP and MPP treatment of autospc constraints consistent. See AUTOSPC on *CONTROL_IMPLICIT_SOLVER.
- Enhance output for *ELEMENT_DIRECT_MATRIX_INPUT (superelements) to describe how they are attached to the LS-DYNA model.
- Enhance superelement computation (*CONTROL_IMPLICIT_MODES or *CONTROL_IMPLICIT_STATIC_CONDENSATION):
    - The computation of the inertia matrix in the presense of rigid bodies is correct.
    - Adjust superelement computation to accept initial velocities.
    - Add null beams for the visualization of superelements.
- Enhance implicit to allow the use of *CONSTRAINED_RIVET in conjunction with axisymmetric shell element problems.
- Add output of performance statistics for the MPP implicit eigensolver to mes0000.
- Add Stress computation to modal dynamics (*CONTROL_IMPLICIT_MODAL_DYNAMIC).
- Allow unsymmetric terms to the assembled stiffness matrix from some implicit features.

- Enhance implicit-explicit switching (IMFLAG < 0 in *CONTROL_IMPLICIT_GENERAL) so that curve |IMFLAG| can be defined using *DEFINE_CURVE_FUNCTION.
- Upgrade the implicit implementation of rack and pinion and screw joints so the joint is driven by relative motion of the assembly instead of absolute motion.
- Add *CONTACT_1D to implicit mechanics.
- *CONTROL_IMPLICIT_ROTATIONAL_DYNAMICS is added to study Rotordynamics using the implicit time integrator.
- *MAT_SEATBELT is supported for implicit by introducing bending stiffness.
- *INITIAL_LAG_MAPPING added to initialize a 3D Lagrangian mesh from the last cycle of a 2D Lagrangian simulation.
- *ELEMENT_SHELL_NURBS_PATCH:
  - Add support for dumping of strain tensor and shell internal energy density for isogeometric shells via interpolation shells.
  - Add conventional mass-scaling for isogeometric shells.
- *LOAD_BODY_POROUS: applies also now to 1D and 2D problems.
- Add *LOAD_SEGMENT_CONTACT_MASK, which currently works in MPP only. This feature masks the pressure from a *LOAD_SEGMENT_SET when the pressure segments are in contact with another material.
- Curve LCID of *LOAD_NODE can be defined by *DEFINE_CURVE_FUNCTION.
- *USER_LOADING: pass more data to user-defined loading subroutine loadud including nodal moment, nodal rotational displacement and velocity, and nodal translational mass and rotational inertia.
- Add load curves for dynamic relaxation for *LOAD_THERMAL_VARIABLE.
- *LOAD_SEGMENT_NONUNIFORM, *LOAD_SEGMENT_SET_NONUNIFORM: By specifying a negative load curve ID the applied load becomes a follower force, i.e., the direction of the load is constant with respect to a local coordinate system that rotates with the segment.
- Make several enhancements to *MAT_172.
- *MAT_HYPERELASTIC_RUBBER (*MAT_077_H) has new thermal option for material properties.
- Add *MAT_ORTHOTROPIC_PHASE_CHANGE, *MAT_ELASTIC_PHASE_CHANGE, and
- *MAT_MOONEY-RIVLIN_PHASE_CHANGE whereby elements change phase as they cross a plane in space.

- Add P1DOFF to 2D seatbelt material, *MAT_SEATBELT_2D, to specify a part ID offset for the internally created 1D seatbelt elements.
- All load curves for *MAT_067 can be defined via *DEFINE_FUNCTION.
- Enhance *MAT_CWM:
    - Add support for shell elements.
    - Add support for hardening curves. Yield stress can be supplied as table depending on plastic strain and temperature.
- Check diagonal elements of C-matrix of *MAT_002/MAT_{OPTION}TROPIC_ ELASTIC and error terminate with message, STR+1306, if any are negative.
- Add a keyword option called MIDFAIL for *MAT_024, (MAT_PIECEWISE_LINEAR_PLASTICITY). When MIDFAIL appears in the keyword, failure by plastic strain will only be checked at the mid-plane. If the mid-plane fails, then the element fails. If there are an even number of integration points through the thickness, then the two points closest to the middle will check for failure and the element fails when both layers fail.
- Enable solid and solid assembly spot welds (*MAT_SPOTWELD) to use the NF parameter for force filtering.
- Add the shear angle in degrees as the first history variable for shell material *MAT_214 (DRY_FABRIC).
- Expand from 2 to 5 the number of additional cards that can be used for the user defined weld failure, OPT = 12 or OPT = 22 on *MAT_SPOTWELD. Now a total of 46 user variables are possible.
- Add a solid spot weld material option in *MAT_SPOTWELD to treat the stress state as uniaxial. This option is available for solid assemblies also.
- Add *MAT_FABRIC form 24 which is a modified version of form 14. The main improvement is that the Poisson's effects work correctly with the nonlinear curves for fiber stress. Also, the output of stress and strain to d3plot are engineering stress and strain instead of 2nd PK stress and Green's strain. Added an option to input curves in engineering stress and strain rather than 2nd PK stress vs. Green's strain. To use this, set DATYP = -2 on *DEFINE_CURVE.
- Increase maximum number of plies from 8 to 24 in a sublaminate with *MAT_-CODAM2.
- Add *MAT_THERMAL_CHEMICAL_REACTION to model a material undergoing a chemical reaction such as an epoxy used in manufacturing composite materials.
- *MAT_058:
    - Add option to use nonlinear (elastic) stress-strain curves instead of constant stiffnesses (EA, EB, GAB).

    - Add option to use strain-rate dependent nonlinear (elastic) stress-strain curves instead of constant stiffnesses (EA, EB, GAB).
    - Add option to define proper poisson ratios PRCA and PRCB (also added in *MAT_158).

- Add option to use yield curve or table in *MAT_100 (*MAT_SPOTWELD) for solid elements.
- Add *MAT_157 for solid elements. This includes an optional variable IHIS that invokes *INITIAL_STRESS_SOLID to initialize material properties on an element-by-element basis. This was developed to allow a user to map/initialize anisotropic material properties from an injection molding simulation.
- *MAT_157 (shells):
    - Add anisotropic scale factor for plastic strain rate (VP = 1 only).
    - Improve local stress projection for VP = 1.
    - Add optional variable IHIS, similar to that described for solids above.
- Add strain rate dependence to *MAT_103 for solids via a table (isotropic hardening only).
- *MAT_136 (*MAT_CORUS_VEGTER): Implemented an alternative, implicit plasticity algorithm (define N.lt.0) for enhanced stability.
- *MAT_244 (*MAT_UHS_STEEL):
    - In plasticity with non-linear hardening, temperature effects and strain rate effects are now dealt with the same way they are implemented in *MAT_106. In particular, strain rate now refers to the plastic strain rate.
    - Allow for the definition of start temperatures for each phase change, for cooling and heating.
    - Account for elastic transformation strains, given as a curve wrt temperature.
    - Add feature to *MAT_244 for welding simulations. Similar to *MAT_270, material can be initialized in a quiet (ghost) state and activated at a birth temperature.
- Furthermore, annealing is accounted for.
- - Modify formula for Pearlite phase kinetics based on Kirkaldy and Venugoplan (1983).
-
- *MAT_249 (*MAT_REINFORCED_THERMOPLASTIC): Implement new material formulation for shells, which is based on additive split of stress tensor.

  - For the thermoplastic matrix, a thermo-elasto-plastic material is implemented,where the temperature dependence is defined by load curves/tables in the input file.
  - Includes hyperelastic fiber contribution.
  - For any integration point, up to three different fiber directions can be defined. Their (non-linear) response to elongation and shear deformations can also be defined with load curves.
  - Includes input parameters for anisotropic transverse shear stiffness.

- *MAT_T07 (*MAT_THERMAL_CWM): Add HBIRTH and TBIRTH which are specific heat and thermal conductivity, resp., used for time t < TISTART.
- One additional parameter (exponent GAMMA) for B-K law of *MAT_138.
- MAT_187: Speed-up of load curve lookup for curves with many points.
- Add new option "MAGNESIUM" to *MAT_233. Differences between tension and compression are included.
- Add enhanced damage model with crack closure effects to *MAT_104.
- Some improvements for *MAT_075 (BILKHU/DUBOIS_FOAM): Volumetric strain rate can now be averaged over NCYCLE cycles, original input curve LCRATE is instead of a rediscretized curve, and averaged strain rate is stored as history variable #3.
- Add new history variables to *MAT_123: A mixed failure indicator as history variable #10 and triaxiality as #11.
- Decrease memory requirements for *MAT_ADD_EROSION by 50%.
- Add *MAT_098 for tetrahedral solid type 13.
- Add new history variable #8 to *MAT_157 for shell elements: "Anisotropic equivalent plastic strain".
- Add tangent stiffness to *MAT_224 for implicit analyses with solid and shell elements.
- Put internal enery on "plastic strain" location for *MAT_027 solids.
- Add new option *MAT_224: BETA .LT. 0: strain rate dependent amount given by load curve ID = -BETA
- Add new flag to switch off all MAT_ADD_EROSION definitions globally.
- This will be the 1st parameter "MAEOFF" on new keyword *CONTROL_MAT.
- Add option to define a load curve for isotropic hardening in *MAT_135.
- *MAT_CDPM is reimplemented by its original developers (Peter Grassl and Dimitros Xenos at University of Glasgow) for enhanced robustness. A new parameter EFC is introduced governing damage in compression and the bilinear law is exchanged for an exponential one.

- *MAT_3-PARAMETER_BARLAT: HR = 7 is complemented with biaxial/shear hardening curves.
- *MAT_FABRIC_MAP:
    - A stress map material for detailed stress response in fabrics, stress can be prescribed through tables PXX and PYY corresponding to functions of biaxial strain states.
    - A compaction effect due to packing of yarns in compression is obtained by specifying BULKC (bulk modulus) and JACC (critical jacobian for the onset of compaction effect). This results ib increasing pressure that resists membrane elements from collapsing and/or inverting.
    - Strain rate effects can be obtained by specifying FXX and FYY which in effect scales the stress based on engineering strain rate. A smoothing effect is applied by using a time window DT.
    - A hysteresis option TH is implemented for stability, given in fraction dissipated energy during a cycle. Can also depend on the strain state through a table.
- *MAT_GENERAL_HYPERELASTIC_RUBBER, *MAT_OGDEN_RUBBER: By specifying TBHYS.LT.0 a more intuitive interpolation of the damage vs. deviatoric strain energy is obtained. It requires however that the damage and strain energy axes are swapped.
- *MAT_SIMPLIFIED_RUBBER: For AVGOPT.LT.0 the absolute value represents a time window over which the strain rates are averaged. This is for suppressing extensive noise used for evaluating stress from tables.
- *MAT_FABRIC: The bending stiffness contribution in material 34, ECOAT/SCOAT/TCOAT, is now supported in implicit calculations.
- Add *MAT_122_3D which is an extension of *MAT_122 to solid elements. This material model combines orthotropic elastic behavior with Hill's 1948 anisotropic plasticity theory and its applicability is primarily to composite materials.
- MPP groupable tied contact: Output messages about initial node movement due to projection like non-groupable routines do.
- MPP tied contact initialization:
    - Change a tolerance in groupable tied contact bucketsort to match the non-groupable code, and fix the slave node thickness used for beam nodes during initial search in non-groupable contact to match groupable contact.
    - Update the slave node from beam thickness calculation for type 9,11, and 12 beams.
- For MPP, set a "last known location" flag to give some indication of where the processors were if an error termination happens. Each writes a message to their

own message file. Look for a line that says "When error termination was triggered, this processor was".

- MPP BEAMS_TO_SURFACE contact: Remove "beam" node mass from the penalty stiffness calculation when soft = 1 is used, which matches SMP behavior.
- Make sure the pfile.log file gets created in case of termination due to *CONTROL_STRUCTURED_TERM.
- Add two new decomposition region-related pfile options "nproc" and "%proc" so that any given decomposition region can be assigned to some subset of all the processors. nproc takes a single argument, which is a specific number of processors. %proc takes a single argument, which is a percentage of processors to use. The old options "lump" and "distribute" are still available and are mapped to the new options thusly:
    - lump => "nproc 1"
    - distribute => "%proc 100.0"
- Tweak MPP beam-to-beam contact routine for better handling of parallel beams.
- MPP: Add support for new solid and shell cost routines, invoked with the pfile option "decomp { newcost }". Will be expanded to include beams, thick shells, etc. in the future.
- MPP contact: add support for IGAP > 2 added to the SINGLE_SURFACE, AUTOMATIC_GENERAL, and *_TO_SURFACE contacts.
- Improve the way MPP computes slave node areas for AUTOMATIC_TIEBREAK contacts (and other that use areas). This should result in less mesh dependency in the failure condition of AUTOMATIC_TIEBREAK contacts.
- MPP: synchronize rigid body flags for shared nodes during rigid-to-deformable switching so that these nodes are handled consistently across processors.
- Add new pfile decomposition region option “partsets”. Takes a list of part sets (SET_PART) from the keyword input and uses them to define a region.
- Apply decomposition transformation (if defined) to:
    - *CONTROL_MPP_DECOMPOSITION_PARTS_DISTRIBUTE
    - *CONTROL_MPP_DECOMPOSITION_PARTSET_DISTRIBUTE
    - *CONTROL_MPP_DECOMPOSITION_ARRANGE_PARTS.
- Honor TIEDPRJ flag on *CONTROL_CONTACT for MPP groupable tied interfaces.
- Increase initial search distance in MPP tied contact to include slave and master thicknesses.
- Tweak MPP_INTERFERENCE contact to better handle deep initial penetrations.

- MPP: Reorganize how *RIGIDWALL_PLANAR_FORCES is handled, which greatly improves scaling.
- Add new MPP pfile option: directory { local_dirs { path1 path2 path3 } which will assign different local working directories to different processors, to balance the I/O load.
- Miscellaneous MPP enhancements:
  - Restructure and reduce memory usage of 3D ALE searching of neighboring algorithm. Now, the code can handle hundreds of millions ALE elements during decomposition.
  - Support *PARTICLE_BLAST.
  - Support SPH 2D contact.
  - Greatly speed up reconstruction of eroding contact surface, (soft = 0,1) when using large number of cores.
- Add the following options for small restarts:
  - *CHANGE_VELOCITY_GENERATION,
  - *CHANGE_RIGIDWALL_option,
  - PSNFAIL option to *CONTROL_SHELL
- MPP full deck restart: Restore behavior consistent with SMP which is that only the nodes of materials being initialized (not all nodes) are initialized from d3full.
- MPP: add full deck restart support for AUTOMATIC_TIEBREAK contact types.
- Implement *DELETE_PART for seatbelt parts. The associated slipring, retractors and pretensioners will be deactivated as well.
- Add support for MPP restarts with USA coupling.
- Add NREP option to *SENSOR_CONTROL to repeat NREP cycles of switches given on Card 2.
- Implement *SENSOR_CONTROL TYPEs BELTPRET, BELTRETRA and BELTSLIP control the pretensioners, retractors and sliprings of a 2D seatbelt.
- Add function SENSORD to *DEFINE_CURVE_FUNCTION to return the value of a sensor.
- Replace *SENSOR_DEFINE_ANGLE with more general *SENSOR_DEFINE_MISC. MTYPEs include ANGLE, RETRACTOR, RIGIDBODY, and TIME.
- Add rcforc output for *CONTACT_2D_NODE_TO_SOLID (supported for ASCII output only; not binout).
- Add temperature output (when applicable) to sphout file (*DATABASE_SPHOUT).

- Add support of *MAT_ALE_VISCOUS for SPH particles. This allows modeling of non-viscous fluids with constant or variable viscosity, i.e, non-newtonian type fluid using SPH.
- Add support of *EOS for *MAT_272 with SPH particles.
- Add support of *MAT_255, *MAT_126, and *MAT_26 (with AOPT = 2 only) for SPH particles.
- Add new keyword command *SECTION_SPH_INTERACTION: Combined with CONT = 1 in *CONTROL_SPH card, this keyword is used to define the partial interaction between SPH parts through normal interpolation method and partial interaction through the contact option. All the SPH parts defined through this keyword will interact with each other through normal interpolation method automatically.
- Add support for *DATABASE_TRACER for axisymmetric SPH (IDIM = -2 in *CONTROL_SPH).
- ICONT in *CONTROL_SPH now affects *DEFINE_SPH_TO_SPH_COUPLING in the sense of enabling or disabling the coupling for deactivated particles.
- The commands *STOCHASTIC_TBX_PARTICLES and *CHEMISTRY_CONTROL_TBX are now available for use (along with the CESE solver) in TBX-based explosives simulations.
- Multi-nozzle injection mode is implemented for spray injection.
- Add logic to skip thermal computations during dynamic relaxation for a coupled thermal-structual problem (i.e. when SOLN = 2 on the *CONTROL_SOLUTION keyword). This does not affect the use of *LOAD_THERMAL keywords during dynamic relaxation.
- Implement *DEFINE_CURVE_FUNCTION for convection, flux, radiation boundary
- conditions in thermal-only analyses, both 2D and 3D.
- *BOUNDARY_CONVECTION, *BOUNDARY_FLUX, and thermal dynamics are implemented for 20 node brick element.
- Include the reading of thermal data to *INCLUDE_BINARY.
- Allow *DEFINE_FUNCTION_TABULATED to be used in any place that requires a function of 1 variable. Specifically, as a displacement scale factor with *INTERFACE_LINKING_NODE.
- Add new MUTABLE option for *PARAMETER and *PARAMETER_EXPRESSION to indicate that it is OK to redefine a specific parameter even if *PARAMETER_DUPLICATION says redefinition is not allowed. Also, only honor the first *PARAMETER_DUPLICATION card.

- Add functions DELAY and PIDCTL to *DEFINE_CURVE_FUNCTION for simulating PID (proportional-integral-derivative) controllers.
- *DEFINE_TABLE: Add check of table's curves for mismatching origin or end points.
- Update ANSYS library to version 16.0.
- Enhance report of "Elapsed time" in d3hsp.
- Add keyword *INCLUDE_UNITCELL to create a keyword file containing user-defined unit cell information with periodic boundary conditions.
- Add *INCLUDE_AUTO_OFFSET: the node and element IDs of the include file will be checked against IDs of the previously read data to see if there is any duplication. If duplicates are found, they will be replaced with another unique ID.

Capabilities added to create LS-DYNA R9.0:

See release notes (published separately) for further details.

- ***AIRBAG**
  - Disable CPM airbag feature during DR and reactivate in the transient phase.
  - *AIRBAG_WANG_NEFSKE_POP_ID pop venting based on RBIDP is now supported correctly (MPP only).
  - *AIRBAG_INTERACTION:
    - Fixed MPP airbag data sync error to allow final pressure among interacted airbags to reach equilibrium.
  - *AIRBAG_PARTICLE:
    - When IAR = -1 and Pbag or Pchamber is lower than Patm, ambient air will inflate the bag through external vents and also fabric porosity.
    - Treat heat convection when chamber is defined.
    - Output pres+ and pres- to CPM interface forces file for internal parts.
    - Allow IAIR = 4 to gradually switch to IAIR = 2 to avoid instability.
    - Allow using shell to define inflator orifice. The shell center and normal will be used as orifice node and flow vector direction.
    - Bug fix for porous leakage for internal fabric parts using CPM.
    - New feature to collect all ring vents into a single vent in order to correctly treat enhanced venting option. All the vent data will only be output to the first part defined in the part set.

        - Evaluate airbag volume based on relative position to avoid truncation. The bag volume becomes independent of coordinate transformation.
        - Support explicit/implicit switch and dynamic relaxation for *AIRBAG_PARTICLE.
        - Support vent/fabric blockage for CPM and ALE coupled analysis.
        - New option in *CONTROL_CPM to allow user defined smoothing of impact forces.
        - Fixed bug affecting *AIRBAG_PARTICLE_ID with PGP encryption.

- ***ALE**

    - *ALE_REFERENCE_SYSTEM_GROUP: For prtype = 4, allow the ALE mesh to follow the center of mass of a set of nodes.
    - *CONTROL_ALE:

        - Add a variable DTMUFAC to control the time step related to the viscosity from
        - *MAT_NULL (if zero, the viscosity does not control the time step).
        - Implement a 2D version of BFAC and CFAC smoothing algorithm.

    - *ALE_SMOOTHING: Automatically generate the list of 3 nodes for the smoothing constraints and implement for MPP.
    - *SECTION_ALE2D, *SECTION_SOLID_ALE: Allow a local smoothing controlled by AFAC,...,DFAC.
    - *ALE_SWITCH_MMG: Allow the variables to be modified at the time of the switch.
    - *CONTROL_REFINE_ALE: Add a variable to delay the refinement after removal (DELAYRGN), one to delay the removal after the refinement (DELAYRMV), and one to prevent any removal in a certain radius around latest refinements (RADIUSRMV).
    - *ALE_STRUCTURED_MESH: Implemented structured ALE mesh solver to facilitate rectilinear mesh generation and to run faster.

- ***BOUNDARY**

    - *BOUNDARY_AMBIENT_EOS: Implement *DEFINE_CURVE_FUNCTION for the internal energy and relative volume curves.
    - *BOUNDARY_AMBIENT: Apply ambient conditions to element sets.
    - Fix for adaptivity dropping SPCs in some cases (MPP only).
    - Added conflict error checking between rigid body rotational constraints and joints between rigid bodies (*CONSTRAINED_JOINT) with *BOUNDARY_PRESCRIBED_ORIENTATION.
    - The first rigid body of the prescribed orientation cannot have any rotational constraints. Only spherical joints or translational motors can be used between the two rigid bodies of the prescribed orientation. For now explicit

  will be allowed to continue with these as warnings.Implicit will terminate at end of input checking.
  - Instead of error terminating with warning message, STR+1371, when *BOUNDARY_PRESCRIBED_MOTION and *BOUNDARY_SPC are applied to same node and dof, issue warning message, KEY+1106, and release the conflicting SPC.
  - Fix erroneous results if *SET_BOX option is used for *BOUNDARY_PRESCRIBED_MOTION.
  - Fix *BOUNDARY_PRESCRIBED_ACCELEROMETER_RIGID for MPP. It may error terminate or give wrong results if more than one of this keyword are used.
  - Fix segmentation fault when using *BOUNDARY_PRESCRIBED_ORIENTATION with vad = 2, i.e. cubic spline interpolation.
  - Fix incorrect behavior if multiple *BOUNDARY_SPC_SYMMETRY_PLANE, i.e. > 1, are used.
  - Fix incorrect motion if *BOUNDARY_PRESCRIBED_MOTION_RIGID_LOCAL is on a rigid part which is merged with a deformable part that has been switched to rigid using *DEFORMABLE_TO_RIGID.
  - Fix incorrect external work when using *BOUNDARY_PRESCRIBED_MOTION with or without_RIGID option. The dof specified in *BPM was not considered when computing the external work. Also, when multiple *BPM applied to the same node/rigid body with different dof may also cause incorrect computation of external work.
  - Fix incorrect velocities when using *BOUNDARY_PRESCRIBED_MOTION_RIGID_LOCAL and *INITIAL_VELOCITY_RIGID_BODY for rigid bodies.
  - Implement check for cases where *MAT_ACOUSTIC nodes are merged with structural nodes on both sides of a plate element and direct the user to the proper approach to this situation - *BOUNDARY_ACOUSTIC_COUPLING.
  - *BOUNDARY_ACOUSTIC_COUPLING with unmerged, coincident node coupling now implemented in MPP.
  - MPP logic corrected so *MAT_ACOUSTIC and *BOUNDARY_ACOUSTIC_COUPLING features may be used with 1 MPP processor.
  - Fixed bug for *BOUNDARY_PRESCRIBED_MOTION if part label option is used.

- **BLAST**

  - Improve *LOAD_BLAST_ENHANCED used with ALEPID option in *LOAD_BLAST_SEGMENT:
  - Rearrange the ambient element type 5 and its adjacent element into same processor to avoid communications.
  - Eliminate several n-by-n searches for segment set and ambient type 5 with its neighboring elements to speed up the initialization.

  - Change the name of keyword *DEFINE_PBLAST_GEOMETRY to *DEFINE_PBLAST_HEGEO. Both names will be recognized.

- ***CESE (Compressible Flow Solver)**

  - Modified the CESE moving mesh CHT interface condition calculation to deal with some occasional MPP failures that could occur with mesh corner elements.
  - Improved the CESE spatial derivatives approximation in order to bring better stability to the CESE solvers.
  - The 3D SMP and MPP CESE immersed boundary solvers now work with structural element erosion.
  - A new energy conservative conjugate heat transfer method has been added to the following 2D and 3D CESE Navier-Stokes equation solvers:

    - Fixed mesh (requires use of *CESE_BOUNDARY_CONJ_HEAT input cards)
    - Moving mesh FSI
    - Immersed boundary FSI

  - Prevent the fluid thermal calculation from using too short a distance between the fluid and structure points in the new IBM CHT solvers.
  - In the under resolved situation, prevent the CHT interface temperature from dipping below the local structural node temperature.
  - Add detection of blast wave arrival at CESE boundary condition face first sensing the leading edge of the pulse (used with *LOAD_BLAST_ENHANCED).
  - Set CESE state variable derivatives to more stable values for the blast wave boundary condition.
  - Corrected time step handling for the CESE Eulerian conjugate-heat transfer solver. This affected only the reported output time.
  - Added CESE cyclic BC capability to the moving mesh CESE solver.
  - Fixed some issues with 2D CESE solvers where the mesh is created via *MESH cards.
  - For the CESE solver coupled with the structural solver (FSI), corrected the time step handling.
  - For the CESE mesh motion solvers, and the ICFD implicit ball-vertex mesh motion solver, added a mechanism to check if all of the imposed boundary displacements are so small that it is not necessary to actually invoke the mesh motion solver. This is determined by comparing the magnitude of the imposed displacement at a node with the minimum distance to a virtual ball vertex (that would appear in the ball-vertex method). The relative scale for this check can be input by the user via field 4 of the *CESE_CONTROL_MESH_MOV card.

- ◦ Changed the NaN check capability for the CESE solvers to be activated only upon user request. This is input via a non-zero entry in field 7 of the *CESE_CONTROL_SOLVER card.
- ◦ Much like the ICFD solver, added a mechanism to adjust the distance used by the contact detection algorithm for the *CESE_BOUNDARY_FSI cards, as well as the new moving mesh conjugate heat transfer solvers. This is available through field 6 of the *CESE_CONTROL_SOLVER card.
- ◦ Added a correction to the moving mesh CESE solver geometry calculation.
- ◦ Corrected the initial time step calculation for both the 2D and 3D moving mesh CESE solvers.
- ◦ For the moving mesh CESE solver, replaced the all-to-all communication for fsi quantities with a sparse communication mechanism.

- ***CHEMISTRY**

  - ◦ The immersed boundary FSI method coupled with the chemistry solver is released.

    - ▪ Only Euler solvers, both in 2D and 3D, are completed with full chemistry.
    - ▪ Using this technique, CESE FSI Immerged Boundary Method coupled to the chemistry solver can be applied to high speed combustion problems such as explosion, detonation shock interacting with structures, and so on.
    - ▪ Some examples are available on our ftp site.

- ***CONTACT**

  - ◦ Fix MPP groupable contact problem that could in some cases have oriented the contact surfaces inconsistently.
  - ◦ Fix bug in *CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_TIED_WELD.
  - ◦ Fix seg fault when using *CONTACT_AUTOMATIC_SINGLE_SURFACE_TIED with consistency mode, .i.e. ncpu < 0, for SMP.
  - ◦ Fix false warnings, SOL+1253, for untied nodes using *CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_TIEBREAK and *CONTACT_AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE_TIEBREAK.
  - ◦ Fix *CONTACT_TIED_SHELL_EDGE_TO_SURFACE when rigid nodes are not tied even when ipback = 1. This applies to SMP only.
  - ◦ Issue warning if SOFT = 4 is used with an unsupported contact type, and reset it to 1.
  - ◦ Change "Interface Pressure" report in intfor file from abs(force/area) to -force/area, which gives the proper sign in case of a tied interface in tension.

- Increase MPP contact release condition for shell nodes that contact solid elements in SINGLE_SURFACE contact.
- Fix for MPP IPBACK option for creating a backup penalty-based tied contact.
- Fix for MPP orthotropic friction in contact.
- Fix for MPP *CONTACT_SLIDING_ONLY that was falsely detecting contact in some cases.
- Skip constraint based contacts when computing the stable contact time step size.
- Add error trap if node set is input for slave side of single surface contact.
- MPP: some fixes for constrained tied contact when used with adaptivity. The behavior of the slave nodes in adaptive constraints was not correct if they were also master nodes of a tied interface. This has been fixed, and support for the rotations required for CONTACT_SPOTWELD have also been added.
- MPP: update to AUTOMATIC_TIEBREAK option 5 to release the slave nodes (and report them as having failed) when the damage curve reaches 0.
- Fix made to routine that determines the contact interface segments, which was not handling pentahedral thick shell elements correctly.
- MPP: fix for strange deadlock that could happen if a user defines a *CONTACT_FORCE_TRANSDUCER that has no elements in it and so gets deleted.
- MPP contact: add support for *DEFINE_REGION to define an active contact region. Contact occurring outside this region is ignored. This is only for MPP contact types:

  - AUTOMATIC_SINGLE_SURFACE
  - AUTOMATIC_NODES_TO_SURFACE
  - AUTOMATIC_SURFACE_TO_SURFACE
  - AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE

- MPP fix for table based friction in non-groupable contact.
- MPP: add frictional work calculation for beams in *CONTACT_ AUTOMATIC_GENERAL.
- Added new option "FTORQ" for contact. Currently implemented only for beams in *CONTACT_AUTOMATIC_GENERAL in MPP. Apply torque to the nodes to compensate for the torque introduced by friction. Issue error message when users try to use SOFT = 2/DEPTH = 45 contact for solid elements.
- R-adaptivity, ADPOPT = 7 in *CONTROL_ADAPTIVE, is now available for SMP version of *CONTACT_SURFACE_TO_SURFACE,_NODES_TO_ SURFACE,_AUTOMATIC_SURFACE_TO_SURFACE, and_ AUTOMATIC_NODES_TO_SURFACE (SOFT = 0 or 1 only).

- The options AUTOMATIC_SURFACE_TO_SURFACE_COMPOSITE has been added to model composite processing. The same option may be used to model certain types of lubrication, and AUTOMATIC_SURFACE_TO_SURFACE_LUBRICATION may be used instead of the COMPOSITE option for clarity. (The two keyword commands are equivalent.)
- Added AUTOMATIC_SURFACE_TO_SURFACE_TIED_WELD to model the simulation of welding. As regions of the surfaces are heated to the welding temperature and come into contact, the nodes are tied.
- Added *CONTACT_TIED_SHELL_EDGE_TO_SOLID. This contact transmits the shell moments into the solid elements by using forces unlike the SHELL_EDGE_TO_SURFACE contact with solid elements. This capability is easier for users than *CONSTRAINED_SHELL_TO_SOLID. The input is identical to *CONTACT_TIED_SHELL_EDGE_TO_SURFACE (except for the keyword).
- Fix incorrect motion of displayed rigidwall between 0.0 < time < birth_time when birth time > 0.0 for *RIGIDWALL_GEOMETRIC_FLAT_MOTION_DISPLAY. The analysis was still correct. Only the displayed motion of the rigidwall is incorrect.
- Fix corrupted intfor when using parts/part sets in *CONTACT_AUTOMATIC_.... This affects SMP only.
- Fix incorrect stonewall energy when using *RIGIDWALL_PLANAR_ORTHO.
- Fix unconstrained nodes when using *CONTACT_TIED_SURFACE_TO_SURFACE_CONSTRAINED_OFFSET resulting in warning message, SOL+540. This affects SMP only.
- Fix spurious repositioning of nodes when using *CONTACT_SURFACE_TO_SURFACE for SMP.
- Enable MAXPAR from optional card A to be used in *CONTACT_TIED_SURFACE_TO_SURFACE. It was originally hard-coded to 1.07.
- The shells used for visualisation of *RIGIDWALL_PLANAR_MOVING_DISPLAY and *RIGIDWALL_PLANAR_MOVING_DISPLAY in d3plot were not moving with the rigidwall. This is now fixed.
- Fix incorrect frictional forces if_ORTHO_FRICTION is used in *CONTACT_AUTOMATIC_SURFACE_TO_SURFACE.
- Fix seg fault when using *CONTACT_ENTITY and output to intfor file with MPP, i.e. s = intfor in command line.
- Fix ineffective birth time for *CONTACT_TIED_NODES_TO_SURFACE.
- Fix untied contacts when using *CONTACT_TIED... with *MAT_ANISOTROPIC_ELASTIC_PLASTIC/*MAT_157.
- Fix MPP hang up when using *CONTACT_ENTITY.
- Allow *CONTACT_AUTOMATIC_GENERAL to use MAXPAR from contact optional card A instead of using the hard coded value of 1.02. This will better detect end to end contact of beams. This applies to SMP only.
- Fix *CONTACT_TIED_SHELL_EDGE_TO_SURFACE for SMP which ignores MAXPAR in contact optional card A.

- Fix seg fault when using *CONTACT_GUIDED_CABLE.
- Fix segmentation fault when using *CONTACT_AUTOMATIC_SINGLE_SURFACE_TIED in consistency mode, i.e. ncpu < 0 in command line, for SMP.
- Fix incorrect contacts when using *CONTACT_AUTOMATIC_GENERAL_INTERIOR for beams with large differences in thickness and when the thinner beams are closer to each other than to the thicker beams. Affects SMP only.
- Fixed force transducers with MPP segment based contact when segments are involved with multiple 2 surface force transducers. The symptom was that some forces were missed for contact between segments on different partitions.
- Fixed an MPP problem in segment based contact that cased a divide by zero during the bucket sort. During an iteration of the bucket sort, all active segments were somehow in one plane which was far from the origin such that a dimension rounded to zero. The fix for this should effect only this rare case and have no effect on most models.
- Fixed thermal MPP segment based contact. The message passing of thermal energy due to friction was being skipped unless peak force data was written to the intfor file.
- Fixed MPP segment based implicit contact. A flaw in data handling caused possible memory errors during a line search.
- Fixed implicit dynamic friction for segment based contact. For sliding friction, the implicit stiffness was reduced to an infinitesimal value. Also, the viscous damping coefficient is now supported for implicit dynamic solutions.
- Fixed segment based contact when the data has all deformable parts that are switched to rigid at the start of the calculation and then switched back to deformable prior to contact occurring. A flaw was causing contact to be too soft. This is now corrected.
- Fixed a flaw in segment based contact with DEPTH = 25 that could allow penetration to occur.
- Improved edge-to-edge contact checking (DEPTH = 5,25,35) and the sliding option (SBOPT = 4,5) in areas where bricks have eroded when using segment based eroding contact.
- Improved the initial penetration check (IGNORE = 2 on *CONTROL_CONTACT) of segment based contact to eliminate false positives for shell segments. Previously, the search was done using mid-plane nodes and the gap or penetration adjusted to account for segment thicknesses after. The new way projects the nodes to the surface first and uses the projected surface to measure penetration. For brick segments with zero thickness there should be no difference. For shell segments, the improved accuracy will me more noticeable for thicker segments.
- Improved segment based contact when SHAREC = 1 to run faster when there are rigid bodies in the contact interface.

- Fixed a possible problem during initialization of segment based contact. Options that use neighbor segment data such as the sliding option and edge-to-edge checking could access bad data if the same nodes were part of both the slave and master surfaces. This would not be a normal occurrence, but could happen.
- Updated segment based contact to improve accuracy at points away from the origin. The final calculations are now done with nodal and segment locations that have been shifted towards the origin so that coordinate values are small.
- The reporting of initial penetrations and periodic intersection reports by segment based contact was corrected for MPP solutions which were reporting incorrect element numbers.
- Fixed memory errors in 2D automatic contact initialization when friction is used.
- Fixed 2D force transducers in the MPP version which could fail to report master surface forces. Also fixed 2 surface 2D force transducers when the smp parallel consistency option is active.
- Fixed *CONTACT_2D_AUTOMAITC_SINGLE_SURFACE and SURFACE_TO_SURFACE which could exhibit unpredictable behavior such as a force spike or penetration.
- Fixed a serious MPP error in the sliding option of *CONTACT_2D_AUTOMATIC that could lead to error termination.
- Fixed a problem with birth time for *CONTACT_2D_AUTOMATIC_TIED when used with sensor switching. Also, fixed a problem in the contact energy calculation that could lead to abnormal terminations. Finally, I made the process of searching for nodes to tie more robust as some problem was found with nodes being missed.
- Fixed a 2D automatic contact bug that occurred if a segment had zero length. An infinite thickness value was calculated by A/L causing the bucket sort to fail.
- Added support for *CONTACT_ADD_WEAR for smp and mpp segment based (SOFT = 2) contact. This option enables wear and sliding distance to be measured and output to the intfor file.
- Added support to segment based contact for the SRNDE parameter on optional card E of *CONTACT.
- Added support to segment based eroding contact for SBOXID and MBOXID on card 1 of *CONTACT.
- Added support for *ELEMENT_SOURCE_SINK used with segment based contact. With this update, inactive elements are no longer checked for contact.
- Added a segment based contact option to allow the PSTIFF option on *CONTROL_CONTACT to be specified for individual contact definitions. The new parameter is PSTIFF on *CONTACT on optional card F, field 1. Prior to this change, setting PSTIFF on *CONTROL_CONTACT set all contact to use the alternate penalty stiffness method. With this update, PSTIFF

on *CONTROL_CONTACT now sets a default value, and PSTIFF on card F can be used to override the default value for an individual contact interface.

- Added support for REGION option on optional card E of *CONTACT when using segment based, SOFT = 2 contact. This works for all supported keywords, SMP and MPP.
- Added master side output in the MPP version for 2-surface force transducers when used with segment based (soft = 2) contact.
- Added contact friction energy to the sleout database file for
  - _2D_AUTOMATIC_SURFACE_TO_SURFACE and
  - _2D_AUTOMATIC_SINGLE_SURFACE contact.
- Enabled segment based contact (SMP and MPP) to work with type 24 (27-node) solid elements.
- Enabled the ICOR parameter on *CONTACT, optional card E to be used with segment based (SOFT = 2) contact.
- Fixed output to d3hsp for *CONTACT_DRAWBEAD using negative curve ID for LCIDRF
- Add slave node thickness and master segment thickness as input arguments to the *USER_INTERFACE_FRICTION subroutine usrfrc (SMP).
- Forming mortar contact can now run with deformable solid tools and honors ADPENE to account for curvatures and penetrations in adaptive step. This applies to h- as well as r-adaptivity.
- Single surface and surface-to-surface mortar contact accounts for rotational degrees of freedom when contact with beam elements. This allows for beams to "roll" on surfaces and prevents spurious friction energy to be generated when in contact with rotating parts.
- Maximum allowable penetration in forming and automatic mortar contacts is hereforth .5*(tslav+tmast)*factor where tmast = thickness of slave segment and tmast = thickness of master segment. The factor is hardwired to 0.95, but is subject to change. Prior to this it was .5*tslav, which seems inadequate (too small) in coping with initial penetrations in automotive applications using standard modeling approaches.
- Up to now, mortar contact has only acted between flat surfaces, now account is taken for sharp edges in solid elements (the angle must initially be larger than 60 degrees), may have to increase the corresponding stiffness in the future.
- When solid elements are involved in mortar contact the default stiffness is increased by a factor of 10. This is based on feedback from customers indicating that the contact behavior in those cases has in general been too soft. This may change the convergence characteristics in implicit but the results should be an improvement from earlier versions.
- The OPTT parameter on *PART_CONTACT for the contact thickness of beams is now supported in mortar contact.

- *CONTACT_ADD_WEAR: A wear law, Archard's or a user defined, can be associated with a contact interface to assess wear in contact. By specifying WTYPE < 0 a user defined wear subroutine must be written to customize the wear law. For the Archard's wear law, parameters can depend on contact pressure, relative sliding velocity and temperature. Contacts supported are *CONTACT_FORMING_SURFACE_TO_SURFACE, *CONTACT_FORMING_ONE_WAY_TO_SURFACE and *CONTACT_AUTOMATIC_SURFACE_TO_SURFACE. To output wear data set NWEAR = 1 or NWEAR = 2 on *DATABASE_EXTENT_INTFOR. If NWEAR is set to 2 then the sliding distance is output to the intfor file, in addition to the wear depth. Otherwise only wear depth is output. Also, the parameter NWUSR specifies the number of user wear history variables to be output in case a user defined wear routine is used. By specifying CID (contact interface id) to a negative number, the wear depth will couple to the contact in the simulation in the sense that the penetration is reduced with wear. The effect is that contact pressure will be redistributed accordingly but is only valid for relatively small wear depths. A formulation for larger wear depths lie in the future which will require modification of the actual geometry.
- Fixed bug affecting *CONTACT_RIGID_NODE_SURFACE (broken at rev. 86847). The bug was in reading *NODE_RIGID_SURFACE.
- A bug fix in *CONTACT_DRAWBEAD_INITILIZE. - The bug was caused by a sudden increase in effective strain after the element passed the drawbead. When the increase in strain is too big, the search algorithm was not working reasonably in the material routine. At the drawbead intersection point, an element could be initialized twice by two bead curves, and cause abnormal thickness distribution.
- Fix a bug in *CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_SMOOTH which removes the limitation that the contact must be defined by segment set.
- SMOOTH option does not apply to FORMING_SURFACE_TO_SURFACE contact. When the SMOOTH option is used, we now write a warning message and disregard the SMOOTH option.

- ***CONSTRAINED**

  - *CONSTRAINED_LAGRANGE_IN_SOLID: Implement *CONSTRAINED_LAGRANGE_IN_SOLID_EDGE in 2D.
  - Fixed bug in *DAMPING_RELATIVE. If the rigid part PIDRB is the slave part in *CONSTRAINED_RIGID_BODIES, the damping card did not work correctly. There is a work-around for previous LS-DYNA versions: set PIDRB to the master part in *CONSTRAIEND_RIGID_BODIES, not the slave part.
  - *CONSTRAINED_RIGID_BODY_INSERT: This keyword is for modeling die inserts. One rigid body, called slave rigid body, is constrained to move

with another rigid body, called the master rigid body, in all directions except for one.

- A variety of enhancements for *CONSTRAINED_INTERPOLATION.
    - Enhanced the error message when nodes involved in the constraint have been deleted.
    - Removed printing of 0 node ID in MPP.
    - Added a warning if there are too many (now set at 1000) nonzeroes in a constraint row for *CONSTRAINED_INTERPOLATION or *CONSTRAINED_LINEAR to protect implicit's constraint processing. These constraints will be processed differently in future releases. We modified the constraint processing software to robustly handle constraint rows with thousands of nonzero entries. We added error checking for co-linear independent nodes as these constraints allow singularities in the model.
- Improved implicit's treatment of the constraints for *CONSTRAINED_BEAM_IN_SOLID.
- Added error checking on the values of the gear ratios in *CONSTRAINED_JOINTS.
- *CONSTRAINED_BEAM_IN_SOLID:
    - Thick shell elements supported.
    - Wedge elements supported.
    - Debonding law by user-defined subroutine (set variable AXFOR > 1000).
    - Debonding law by *DEFINE_FUNCTION (set variable AXFOR < 0).
- Error terminate with message, SOL+700, if CIDA and CIDB is not defined for *CONSTRAINED_JOINT_STIFFNESS_GENERALIZED.
- Fix incorrect constraints on rotary dof for adaptivity.
- Fix incorrect motion if NRBF = 2 in *DEFORMABLE_TO_RIGID_AUTOMATIC and if any of the *CONSTRAINED_NODAL_RIGID_BODY nodes belongs to a solid element.
- Fix input error when using large load curve ID for FMPH, FMT, FMPS in card 3 of *CONSTRAINED_JOINT_STIFFNESS with_GENERALIZED or_TRANSLATIONAL options.
- Fix seg fault if using tables for FMPH of *CONSTRAINED_JOINT_STIFFNESS and if the angle of rotation is less than the the abscissa of the table or load curves.
- Fixed an problem with *CONSTRAINED_BEAM_IN_SOLID when used in a model that also uses segment based eroding contact in the MPP version. This combination now works.

- Improved the precision of spot weld constraints (*CONSTRAINED_SPOTWELD) to prevent possible divide by zeroes when the inertia tensor is inverted. This affects the single precision version only.
- Fix for damage function in *CONSTRAINED_INTERPOLATION_SPOTWELD, MODEL = 2.
- Add some user-friendly output (rigid body id) to d3hsp for *CONSTRAINED_NODAL_RIGID_BODY_INERTIA.
- Add new option to *CONSTRAINED_SPR2 to connect up to 6 shell element parts (metal sheets) with only one rivet location node. This is invoked by defining extra part IDs for such a multi-sheet connection.
- Add more flexibility to *CONSTRAINED_SPR2: Load curve function exponent values originally hardwired as "8" can now be defined with new input parameters EXPN and EXPT.
- Fixed bug wherein the joint ID in *CONSTRAINED_JOINT_COOR was read incorrectly.
- Fixed duplicate ID for *CONSTRAINED_SPOTWELD, ..._NODE_SET,_POINTS and_SPR2.
- Fix keyword reader for SPR4 option in *CONSTRAINED_INTERPOLATION_SPOTWELD, where BETA2 was replaced by BETA3.
- Significantly reduce the memory demand in the initialization stage of *CONSTRAINED_MULTIPLE_GLOBAL for implicit analysis.
- The unit cell mesh and constraint generated by *INCLUDE_UNITCELL now supports job ID.

- ***CONTROL**

  - Terminate and print error KEY+1117 for cases that use *INCLUDE_TRANSFORM in 3d r-adaptvity. More work is needed to make this combination work.
  - Changed SOL+41 message ("reached minimum step") from an error to a warning and terminate normally. This message is triggered when the DTMIN criterion set in *CONTROL_TERMINATION is reached.
  - Fixed bug in which h-adaptivity missed some ADPFREQ-based adaptations when IREFLG < 0 (*CONTROL_ADAPTIVE).
  - Fixed bug: MS1ST in *CONTROL_TIMESTEP causes non-physical large mass and inertia on Nodal Rigid Bodies if Dynamic Relaxation is active. The error occurs at the start of the transient solution. The mass can become very large, so the model may appear to be over-restrained.
  - Add new input check for curves. After rediscretizing curves, check to see how well the original values can be reproduced. If the match is poor, write out See variable CDETOL in *CONTROL_OUTPUT.
  - Added the ability to specify unique values LCINT for each curve, which override the value set in *CONTROL_SOLUTION. Note: the largest value of LCINT that appears will be used when allocating memory for each load

curve, so a single large value can cause significant increases in the memory required for solution.

- The DELFR flag in *CONTROL_SHELL has new options for controlling the deletion of shell elements. This feature is aimed at eliminating single, detached elements and/or elements hanging on by one shared node.
- Fix spurious deletion of elements when using TSMIN.ne.0.0 in *CONTROL_TERMINATION, ERODE = 1 in *CONTROL_TIMESTEP and initialized implicitly in dynamic relaxation.
- Fix spurious error, STR+755, if using *DAMPING_FREQUENCY_RANGE with *CONTROL_ADAPTIVE.
- Add new feature to *CONTROL_SOLUTION, LCACC, to truncate load curve to 6 significant figures for single precision & 13 significant figures for double precision. The truncation is done after applying the offset and scale factors.
- Fix "*** termination due to mass increase ***' error when using mass scaling with *ELEMENT_MASS_PART.
- Fix input error 'node set for nodal rigid body # not found' when using *PART_INERTIA with *CONTROL_SUBCYCLE.
- Fixed the negative DT2MS option on *CONTROL_TIMESTEP for thick shell types 5, 6, and 7.
- Fixed bug in *CONTROL_CHECK_SHELL if PSID.lt.0 (part set ID) is used
- Add new option NORBIC to *CONTROL_RIGID to bypass the check of rigid body inertia tensors being too small.
- Add new option ICRQ to *CONTROL_SHELL for continuous treatment of thickness and plastic strain across element edges for shell element types 2, 4, and 16 with max. 9 integration points through the thickness.
- Add new option ICOHED to *CONTROL_SOLID. If this value is set to 1, solid cohesive elements (ELFORM 19-22) will be eroded when neighboring (nodewise connected) shell or solid elements fail.
- Beam release conditions are now properly supported in selective mass scaling, see IMSCL on *CONTROL_TIMESTEP.
- Modified MSGMAX in *CONTROL_OUPUT: MSGMAX Maximum number of each error/warning message

    - > 0 number of message to screen output, all messages written to messag/d3hsp
    - < 0 number of messages to screen output and message/d3hsp
    - = 0 the defaul is 50

- Fix bugs in 3D solid adaptivity (*CONTROL_ADAPTIVE,ADPOPT = 7) so that the solid adaptivity will still work when there are any of the following in the model:

    - thick shells (*SECTION_TSHELL),
    - massless nodes,

        - *LOAD_SEGMENT_{option}.

    - Added PARA = 2 to *CONTROL_PARALLEL which actives consistent force assembly in parallel for SMP. An efficient parallel algorithm is implemented for better performance when the consistency flag is turned on. It shows better scaling with more cpus. This option is overridden by parameter "para=" on the execution line.

- **DEM(Discrete Element Method)**

    - Added output of following DES history variables to d3plot:

        - nodal stress and force
        - pressure
        - density
        - force chain
        - damage calculation when *DEFINE_DE_BOND is defined

    - Added output of following DES history variables to demtrh (*DATABASE_TRACER_DE):

        - coordination number
        - porosity and void ratio
        - stress
        - pressure
        - density

    - Output ASCII format for demrcf if BINARY.eq.3.
    - Implement gauss distribution of DE sphere radius for *DEFINE_DE_INJECTION. The mean radius is 0.5*(rmin+rmax) and standard deviation is 0.5*(rmax-rmin).
    - For DE sphere, implement the stress calculation for REV (Representative Elementary Volume) using *DATABASE_TRACER_DE and specific RADIUS.
    - Add *BOUNDARY_DE_NON_REFLECTING for defining non-reflecting boundary conditions for DE spheres.
    - For *CONTROL_DISCRETE_ELEMENT, add the option to create the liquid bridge if the initial distance between two DE spheres is smaller than predefined gap.
    - Added *DATABASE_DEMASSFLOW, see *DEFINE_DE_MASSFLOW_PLANE, for measuring the mass flow of DE spheres through a surface. The surface is defined by part or part set. Output file is 'demflow'.
    - Add *DEFINE_DE_INJECTION_ELLIPSE, to define a circular or elliptical injection plane.

  - Add *DEFINE_PBLAST_AIRGEO for *PARTICLE_BLAST which defines initial geometry for air particles.
  - Add DEM stress calculation when coupling with segment (*DEFINE_DE_TO_SURFACE_COUPLING).
  - Fix error in demtrh file output (Window platform only).

- **EFG (Element Free Galerkin)**

  - Fix bug for ELFORM = 41 implicit when there are 6-noded/4-noded elements.

- ***ELEMENT**

  - Fix a 2d seatbelt bug triggered by having both 1d and 2d seatbelts, and a 1d pretensioner of type 2, 3 or 9.
  - Fix MPP bug initializing multiscale spotweld in the unexpected case where the spotweld beam is merged with the shells rather than tied via contact.
  - Fix bug for *INCLUDE_UNITCELL.
  - *CONTROL_REFINE_...: Implement the parent-children transition in *CONTACT_2D_SINGLE_SURFACE when a shell refinement occurs.
  - Fix error traps for *ELEMENT_SEATBELT_... , for example, error termination due to convergence failure in retractors. These error traps worked but could lead to a less graceful termination than other LS-DYNA error traps.
  - Correct calculation of wrap angle in seatbelt retractor.
  - Add MPP support for *ELEMENT_LANCING.
  - *ELEMENT_SEATBELT:

    - Fix a MPP belt bug that can happen when buckle pretensioner is modeled as a type-9 pretensioner.
    - 2D belt and 1D belt now can share the same *MAT_SEATBELT.
    - The section force for 2d belt is recoded to provide more robust and accurate results.
    - The loading curve LLCID of *MAT_SEATBELT can be a table defining strain-rate dependent stiffness curve.
    - IGRAV of *ELEMENT_SEATBELT_ACCELEROMETER can be a curve defining gravitation flag as a function of time.

  - Add *NODE_THICKNESS to override shell nodal thickness otherwise determined via *SECTION_SHELL, *PART_COMPOSITE, or *ELEMENT_SHELL_THICKNESS.
  - Fix input error when using *DEFINE_ELEMENT_DEATH with BOXID > 0 for MPP.
  - Implement subcycling for thick shells.
  - Fix ineffective *DEFINE_HAZ_PROPERTIES when solid spotwelds and hex spotweld assemblies are both present.

- Fix incorrect beta written out for *ELEMENT_SHELL_BETA in dynain file when *PART_COMPOSITE keyword is present in the original input.
- Fix NaN output to elout_det and spurious element deletion if NODOUT = STRAIN or STRAIN_GL or ALL or ALL_GL.
- Fix incorrect reading of TIME in card 3 of *ELEMENT_SEATBELT_SENSOR SBSTYP = 3 when long = s in command line.
- *PART_COMPOSITE: Increased the explicit solution time step for thin shell composite elements. The existing method was overly conservative. The new method is based on average layer stiffness and density.
- In conjunction with the above change in composite time step calculation, increase nodal inertia in the rare cases of *PART_COMPOSITE in which the bending stability is not satisfied by the membrane stability criterion. The inertia is only increased in the cases where it is necessary; for most models this change has no effect, but this can occur in the case of sandwich sections with stiffer skins around a less stiff core.
- Corrected rotational inertia of thin shells when layers have mixed density and the outer layers are more dense than inner layers. The fix will mostly affect elements that are very thick relative to edge length.
- Fixed default hourglass control when the *HOURGLASS control card is used but no HG type is specified. We were setting to type 1 instead of 2. Also, fixed the default HG types to match the User's Manual for implicit and explicit.
- Fixed the part mass that was reported to d3hsp when *ELEMENT_SHELL_SOURCE_SINK is used. The inactive elements were being included causing too high mass.
- Prevent inactive shell elements (from *ELEMENT_SHELL_SOURCE_SINK) from controlling the solution time step.
- Fixed the reported strain tensor in elements created by *ELEMENT_SHELL_SOURCE_SINK when strain output is requested. The history was being retained from the previous elements with the same ID.
- Fixed torsion in linear beam form 13. A failure to add the torsional moment at node 2 caused an inability to reach equilibrium in the torsional mode.
- Fixed solid element 4 so that rigid body translation will not cause strain and stress due to round-off error.
- Mixed parallel consistency when used with solid element type 20. A buffer was not being allocated leading to a memory error.
- Changed the MPP behavior of discrete beams (ELFORM = 6) when attached to elements that fail. They were behaving like null beams, in the sense that it was possible for beam nodes to become dead due to attached elements failing, and discrete beams would be no longer visualized even if the beams themselves had not failed. With this change, the MPP discrete beams now behave like other beams in that the beams have to fail before they are removed. MPP and SMP behavior is now consistent.

- Improved the precision of the type 2 Belytschko Schwer resultant beam to prevent energy growth in single precision.
- Fixed the NLOC option on *SECTION_SHELL for the BCIZ triangle elements (ELFORM = 3) and the DKT triangle elements (ELFORM = 17). The offset was scaled by the solution time step so typically the offset was much smaller than expected.
- Fixed elout stress output for shell element forms 23 and 24. The in-plane averaging was incorrect causing wrong output.
- Changed *ELEMENT_TSHELL so that both the COMPOSITE and BETA options can be read at the same time. Prior to the fix, only the first one would be read.
- Fixed all thick shells to work with anisotropic thermal strains which can be defined by *MAT_ADD_THERMAL. Also, this now works by layer for layered composites.
- Fixed implicit solutions with thick shells with *MAT_057 when there are also solid elements in the model that use *MAT_057. Thick shells support only the incremental update of the F tensor but a flag was set incorrectly in the material model.
- Fixed *MAT_219 when used with thick shell types 3, 5, and 7. A failure to initialize terms for the time step caused a possible wrong time step.
- Fixed orthotropic user defined materials when used with thick shell elements. The storing of the transformation matrix was in the wrong location leading to wrong stress and strain.
- For thick shell composites that use element forms 5 and 7, the user can now use laminated shell theory along with the TSHEAR = 1 on *SECTION_TSHELL to get a constant shear stress through the thickness with a composite.
- Fixed the initialization of *MAT_CODAM2/*MAT_219 when used with thick shell forms 3, 5, or 7. The 3D thick shell routine uses only 2 terms for the transformation and therefore needs unique initialization of the transformation data.
- Fixed thick shell types 3 and 5 when used in implicit solutions with *MATs 2, 21, 261, and 263. The material constitutive matrix for *MATs 2 and 21 was not rotated correctly causing wrong element stiffness. The constitutive matrix for *MATs 261 and 263 was not orthotropic. Also, for *MAT_021, type 5 thick shell needed some material terms defined to correct the assumed strain.
- Fixed thick shell forms 3 and 5 when used in implicit solutions with non-isotropic materials. The stiffness matrix was wrong due to incorrect transformations.
- Also, fixed the implicit stiffness of thin and thick shells when used with laminated shell theory by assumed strain (LAMSHT = 3,4,5 on *CONTROL_ACCURACY). Elements were either failing to converge or converging more slowly due to the failure to adjust the stiffness matrix to be consistent with the assumed strain.

- Added support for *ELEMENT_SHELL_SOURCE_SINK to form 2 elements with BWC = 1 on *CONTROL_SHELL.
- Fixed the s-axis and t-axis orientation of beam spot welds in the MPP version when those beam weld elements are defined with a 3rd node. The 3rd node was being discarded prior to initializing the beam orientation so the s and t-axes were being randomly assigned as if the 3rd node had not been assigned. The effect on solutions is likely fairly minimal since beam material is isotropic and failure typically is too, but may not be.
- Added Rayleigh damping (*DAMPING_PART_STIFFNESS) for thick shell formulations 1, 2, and 6. Previously, it was available for only the thick shells that call 3D stress updates, (forms 3 and 5), but now it is available for all thick shell formulations.
- Added new SCOOR options for discrete beam section 6 (*SECTION_BEAM). A flaw was found in how the discrete beam accounts for rigid body rotation when SCOOR = -3, -2, +2, and +3. A correction for this is made and introduced as new options, SCOOR = -13, -12, +12, and +13. A decision was made to leave the existing options SCOOR = -2, +2, -3 and +3 unchanged so that legacy data could run without changes.
- Enabled the ELFORM 18 linear DKT shell element to work with *PART_COMPOSITE and with an arbitrary number of through thickness integration points. It was limited to a single material and 10 Gauss points.
- Added the possibility to write *ELEMENT_SOLID_ORTHO into dynain file if requested. To activate this add OPTCARD to *INTERFACE_SPRINGBACK and set SLDO = 1.
- Refine characteristic length calculation for 27-node solid (ELFORM 24). This change may increase the time step substantially for badly distorted elements.
- Implement selective reduced integration for 27-node solid (ELFORM 24).
- Allow part sets to be used in *DEFORMABLE_TO_RIGID_AUTOMATIC. Either PID is defined negative or "PSET" is set in column 3 (D2R) or 2 (R2D).
- Add new option STRESS = 2 to *INCLUDE_STAMPED_PART: no stresses and no history variables are mapped with that setting.
- New keyword *PART_STACKED_ELEMENTS provides a method to define and to discretize layered shell-like structures by an arbitrary sequence of shell and/or solid elements over the thickness.
- The geometric stiffness matrix for the Belytschko beam element type 2 has been extended to include nonsymmetric terms arising from nonzero moments. Provides "almost" quadratic convergence, still some terms missing to be added in the future. Also support a strongly objective version activated by IACC on *CONTROL_ACCURACY.
- The geometric stiffness for the Hughes-Liu element type 1 is fixed.
- Fix parsing error in *SECTION_BEAM_AISC.

- **EM (Electromagnetic Solver)**

  - Add the new EM 2d axi solver in SMP and MPP for EM solver 1 (eddy current). It is coupled with the mechanics and thermal solvers.
  - The new EM 2d can be used with RLC circuits on helix/spiral geometries using *EM_CIRCUIT_CONNECT.
  - Add EM contact into new EM 2d axi, in SMP and MPP.
  - Add *EM_BOUNDARY support in new EM 2d axi solver.
  - Introduce scalar potential in new EM 2d axi. The 2d axi can also be coupled with imposed voltage.
  - Add new keyword *EM_CIRCUIT_CONNECT to impose linear constraints between circuits with imposed currents in 3d solvers. This allows for example to impose that the current in circuit 1 is equal to the current in circuit 2 even if the 2 corresponding parts are not physically connected.
  - Add *EM_VOLTAGE_DROP keyword to define a voltage drop between 2 segment sets. This voltage drop constraint is coupled to the contact constraint so that the contact (voltage drop = 0) has priority over the *EM_VOLTAGE_DROP constraint.
  - Add *EM_CONTROL_SWITCH_CONTACT keyword to turn the EM contact detection on and off.
  - NCYCLBEM/NCYCLFEM in *EM_SOLVER_... can now be different than 1 when EM_CONTACT detected.
  - Add RLC circuit for type 3 solver (resistive heating).
  - Add computation of mutuals/inductances in 2d axi for output to em_circuit.dat
  - Add criteria on autotimestep calculation when R,L,C circuit used to take into account R,L,C period.
  - Fix keyword counter in d3hsp.
  - Better and clearer output to terminal screen.
  - Support jobid for EM ascii file outputs.

- **Forming**

  - Improvements to trimming:

    - *DEFINE_CURVE_TRIM_NEW: if trim seed node is not defined, we will search a seed node based on nodes from the sheet blank and the inside/outside flag definition for the trimming curves.
    - Map strain tensors to triangular elements after trimming.

  - Add a new function to the trim of solid elements in normal (3-D) trimming case, related to *DEFINE_CURVE_TRIM_3D. If the trimming curve is close to the bottom side, set TDIR = -1. If the trimming curve is close to the upper side, set TDIR = 1.

- Add to *ELEMENT_LANCING. Allow parametric expression for variables END and NTIMES.
- A bug fix for *CONTROL_FORMING_AUTOPOSITION_PARAMETER_SET: Fix distance calculation error when the target mesh is too coarse.
- Improvements to springback compensations:
    - Output the new trimming curve with *DEFINE_CURVE_TRIM_3D (previously *DEFINE_CURVE_TRIM), so that it can be easily converted to IGES curve by LS-PrePost. or used in another trimming calculation.
    - Output each curve to IGES format in the following name format: newcurve_scp001.igs, newcurve_scp002.igs, newcurve_scp003.igs, etc.
    - Output change in file "geocur.trm". This update will allow change from *DEFINE_CURVE_TRIM(_3D,_NEW), whatever is used for input.
- Add a new keyword: *DEFINE_FORMING_CONTACT to facilitate the forming contact definitions.
- Add a new keyword *DEFINE_FORMING_CLAMP, to facilitate clamping simulation.
- A new feature in mesh fusion, which allows a moving box to control the fusion, only if the center of the elements is inside the box can the elements can be coarsened. Can be used in conjuntion with *DEFINE_BOX_ADAPTIVE.
- Add a new feature to *DEFINE_BOX_ADAPTIVE: Moving box in adaptivity, useful in roller hemming and incremental forming.
- In mesh coarsening, if the node is defined in a node set, the connected elements will be kept from being coarsened. Previously, only *SET_NODE_LIST was supported. Now option *SET_NODE_GENERAL is allowed.
- Add a new function: mesh refinement for sandwich part. The top and bottom layers are shell elements and the middle layer is solid elements. Set IFSAND to 1 in *CONTROL_ADAPTIVE.
    - Applies to both 8-noded and 6-noded solid elements.
    - Map stress and history variables to the new elements.
- New features related to blank size development *INTERFACE_BLANKSIZE_DEVELOPMENT:
    - Add *INTERFACE_BLANKSIZE_SYMMETRIC_PLANE to define symmetric plane in blank size development
    - Add *INTERFACE_BLANKSIZE_SCALE_FACTOR. For each trimming, different scale factors can be used to compensate the blanksize. This is especially useful when the inner holes are small. Includes an

option of offset the target curve which is useful if multiple target curves (e.g., holes) and formed curves are far from each other.

  - Allow target curve to be outside of the surface of the blank.
  - Add sorting to the mesh so the initial mesh and the formed mesh do not need to have the same sequence for the nodes.
  - Add a new variable ORIENT, set to "1" to activate the new algorithm to potentially reduce the number of iterations with the use of *INTERFACE_BLANKSIZE_SCALE_FACTOR (scale = 0.75 to 0.9).
  - Fix smooth problem along calculated outer boundary.
  - Automatically determine the curve running directions (IOPTION = 2 and -2 now both give the same results).
  - Accept parameteric expression.

- A bug fix for springback compensation: *INCLUDE_COMPENSATION_SYMMETRIC_LINES Fix reading problem of free format in the original coding.
- Add a new keyword *CONTROL_FORMING_BESTFIT. Purpose: This keyword rigidly moves two parts so that they maximally coincide. This feature can be used in sheet metal forming to translate and rotate a spring back part (source) to a scanned part (target) to assess spring back prediction accuracy. This keyword applies to shell elements only.
- Improvements to *CONTROL_FORMING_AUTOCHECK:

  - When IOFFSET = 1, rigid body thickness is automatically offset, based on the MST value defined in *CONTACT_FORMING_ONE_WAY_SURFACE.
  - Add new variable IOUTPUT that when set to 1 will output the offset rigid tool mesh, and the new output tool file is: rigid_offset.inc. After output the simulation stops. See R9.0 Manual for further details.
  - When both normal check and offset are used, small radius might cause problem for offsetting. The new modification will check the normal again after offsetting the tool
  - When outputting the rigid body mesh, output the bead nodes also.
  - Changes to *CONTROL_FORMING_AUTOCHECK when used together with SMOOTH option: check and fix rigid body bad elements before converting the master part ID to segment set id to be used by SMOOTH option.
  - Set IOUTPUT.eq. 3 to output rigid body mesh before and after offset.
  - Fix problems offseting a small radius to a even smaller radius.
  - Remove T-intersection.

- For *CONTROL_IMPLICIT_FORMING, fix output messages in d3hsp that incorrectly identified steps as implicit dynamic when they were actually implicit static.
- Improve *CONTROL_FORMING_UNFLANGING:

    - Automatically calculate CHARLEN, so user does not need to input it anymore.
    - Allow nonsmooth flange edge.
    - Instead of using preset value of 0.4 (which works fine for thin sheet metal), blank thickness is now used to offset the slave node (flanges) from the rigid body (die).

- ***FREQUENCY_DOMAIN**

  - *FREQUENCY_DOMAIN_RANDOM_VIBRATION: Fixed a bug in dumping d3psd binary database, when both stress and strain are included.
  - *FREQUENCY_DOMAIN_SSD_ERP: Implemented the Equivalent adiated Power (ERP) computation to MPP.
  - *FREQUENCY_DOMAIN_ACOUSTIC_BEM:

    - Enabled running dual collocation BEM based on Burton-Miller formulation (METHOD = 4) with vibration boundary conditions provided by Steady State Dynamic analysis (*FREQUENCY_DOMAIN_SSD).
    - Added exponential window function for FFT (FFTWIN = 5).
    - Implemented a new forward and backward mixed radix FFT.
    - Implemented acoustic computation restart from frequency domain boundary conditions, in addition to time domain boundary conditions (RESTRT = 1).
    - Enabled out-of-core velocity data storage, to solve large scale problems.
    - Implemented option HALF_SPACE to Rayleigh method (METHOD = 0) to consider acoustic wave reflection.
    - Added velocity interpolation to take care of mismatching between acoustic mesh and structural mesh (*BOUNDARY_ACOUSTIC_MAPPING), for the case that the boundary conditions are provided by Steady State Dynamic analysis.
    - Added weighted SPL output to acoustic computation (DBA = 1,2,3,4).
    - Implemented radiated sound power, and radiation efficiency computation to collocation BEMs (METHOD = 3,4). Added new ASCII xy-plot databases Press_Power and Press_radef to save the sound power and radiation efficiency results.
    - Enabled using both impedance and vibration (velocity) boundary conditions in acoustic simulation.

  - *FREQUENCY_DOMAIN_ACOUSTIC_FEM:

    - Added weighted SPL output to FEM acoustics (DBA = 1,2,3,4).
    - Implemented option EIGENVALUE to perform acoustic eigenvalue analysis; added ASCII database eigout_ac to save acoustic eigenvalue

results; added binary plot database d3eigv_ac to save acoustic eigenvectors.
    - Enabled consideration of nodal constraints in acoustic eigenvalue analysis.
    - Enabled FEM acoustic analysis with frequency dependent complex sound speed.
    - Implemented pressure and impedance boundary conditions.

  - *FREQUENCY_DOMAIN_ACOUSTIC_FRINGE_PLOT:

    - Added this keyword to 1) generate acoustic field points as a sphere or plate mesh (options SPHERE and PLATE), or 2) define acoustic field points mesh based on existing structure components (options PART, PART_SET and NODE_SET) so that user can get fringe plot of acoustic pressure and SPL. The results are saved in binary plot database d3acs (activated by keyword *DATABASE_FREQUENCY_BINARY_D3ACS).

  - *FREQUENCY_DOMAIN_RANDOM_VIBRATION:

    - Changed displacement rms output in d3rms to be the displacement itself, without adding the original nodal coordinates.
    - Implemented von mises stress PSD computation in beam elements.
    - Implemented fatigue analysis with beam elements.
    - Added strain output to binary plot databases d3psd and d3rms, and binout database elout_psd.
    - Added initial damage ratio from multiple loading cases (INFTG > 1).

  - *FREQUENCY_DOMAIN_SSD:

    - Implemented option ERP to compute Equivalent Radiated Power. It is a fast and simplified way to characterize acoustic behavior of vibrating structures. The results are saved in binary plot database d3erp (activated by keyword *DATABASE_FREQUENCY_BINARY_D3ERP), and ASCII xyplot files ERP_abs and ERP_dB.
    - Implemented fatigue analysis based on maximum principal stress and maximum shear stress.

- **ICFD (Incompressible Flow Solver)**

  - *ICFD_BOUNDARY_FSWAVE: Added a boundary condition for wave generation of 1st order stokes waves with free surfaces.
  - *ICFD_DATABASE_DRAG_VOL: For computing pressure forces on volumes ID (useful for forces in porous domains), output in icfdragivol.dat and icfdragivol.#VID.dat.

- *ICFD_CONTROL_DEM_COUPLING: Coupling the ICFD solver with DEM particles is now possible.
- *ICFD_CONTROL_MONOLITHIC: Added a monolithic solver (=1) which can be selected instead of the traditional fractional step solver (=0).
- *ICFD_CONTROL_POROUS: This keyword allows the user to choose between the Anisotropic Generalized Navier-Stokes model (=0) or the Anisotropic Darcy-Forchheimer model (=1) (for Low Reynolds number flows). The Monolithic solver is used by default for those creeping flows.
- *ICFD_CONTROL_TURBULENCE:

  - Modified existing standard k-epsilon.
  - Added Realizable k-epsilon turbulence model.
  - Added Standard 98 and 06 Wilcox and Menter SST 03 turbulence models.
  - Added Several laws of the wall.
  - Added Rugosity law when RANS turbulence model selected.

- *ICFD_MODEL_POROUS:

  - Added Porous model 5 for anistropic materials defined by P-V experimental curves.
  - Added porous model 6 for moving domain capabilities for Porous Media volumes using load curves for permeabilities directions.
  - Added porous model 7 for moving domain capabilities for Porous Media volumes using ICFD_DEFINE_POINT for permeabilities directions.
  - Added porous model 9 for a new Anisotropic Porous Media flow model (PM model ID = 9): It uses a variable permeability tensor field which is the result of solid dynamic problems. The model reads the solid mesh and the field state and maps elemental permeability tensor and solid displacements to the fluid mesh.

- *ICFD_MODEL_NONNEWT:

  - Added a few models for non newtonian materials and temperature dependant viscosity :
    - model 1 : power law non newtonian (now also temperature dependant)
    - model 2 : carreau fluid
    - model 3 : cross fluid
    - model 4 : herschel-bulkley
    - model 5 : cross fluid II
    - model 6 : temperature dependant visc (sutherland)
    - model 7 : temperature dependant visc (power law)

    - model 8 : load curve dependant visc, model 8 is especially interesting since a DEFINE_FUNCTION can be used (for solidification applications).

- *ICFD_SOLVER_TOL_MONOLITHIC: Used to define atol, rtol, dtol and maxits linear solver convergence controls of the monolithic NS time integration
- *MESH_BL: Added support for boundary layer mesh creation by specifying the thickness, number of layers, first node near the surface and the strategy to use to divide and separate the elements inside the BL adding.
- *ICFD_BOUNDARY_PRESCRIBED_VEL: Added the support of DEFINE_FUNCTION making the second line of the keyword obsolete.
- *ICFD_CONTROL_TIME: Min and Max timestep values can be set.
- *ICFD_DATABASE_DRAG:
    - Added frequency output.
    - Added option to output drag repartition percentage in the d3plots as a surface variable.
- *ICFD_CONTROL_IMPOSED_MOVE: This keyword now uses *ICFD_PART and *ICFD_PART_VOL instead of *MESH_VOL for ID. It is now possible to impose a rotation on a part using Euler angles.
- *ICFD_CONTROL_OUTPUT:
    - Field 4 now to output mesh in LSPP format and in format to be run by the icfd solver (icfd_fluidmesh.key and icfd_mesh.key)
    - icfd_mesh.key now divides the mesh in ten parts, from best quality element decile to worst.
    - A new mesh is now output at every remeshing.
    - Added support for parallel I/O for Paraview using the PVTU format.
- *ICFD_DEFINE_POINT: Points can now be made to rotate or translate.
- *ICFD_MAT:
    - Nonnewtonian models and Porous media models are now selected in the third line by using the new ICFD_MODEL keyword family.
    - HC and TC can now be made temperature dependent.
- *ICFD_CONTROL_DIVCLEAN: Added option 2 to use a potential flow solver to initialize the Navier Stokes solver.
- *ICFD_CONTROL_FSI: Field 5 provides a relaxation that starts after the birthtime.
- *ICFD_CONTROL_MESH: Field 3 added a new strategy to interpolate a mesh size during the node insertion. In some cases it speeds up the meshing process and produces less elements. Field 4 changes the meshing strategy in 2d.

- *ICFD_CONTROL_SURFMESH: Added support for dynamic re-meshing/adaptation of surface meshing.
- *ICFD_BOUNDARY_PRESCRIBED_VEL:VAD = 3 now works with DOF = 4.
- SF can be lower than 0.
- PID can be over 9999 in *ICFD_DATABASE_FLUX.
- Fixed d3hsp keyword counter.
- Clarified terminal output.
- Y+ and Shear now always output on walls rather than when a turbulence model was selected.
- Added coordinate of distorted element before remeshing occurs. Output on terminal and messag file
- Fixed bug in conjugate heat transfer cases. When an autotimestep was selected in *ICFD_CONTROL_TIME, it would always only take the thermal timestep.
- An estimation of the CFL number is now output in the d3plot files. This is not the value used for the autotimestep calculation.
- Turbulence intensity is now output in the d3plots.
- Jobid now supported for ICFD ASCII File outputs.
- Fixed communication of turbulent constants in MPP.
- Fixed the Near Velocity field output.
- Increasing the limit of number of parts for the model.
- Temperature added as a surface variable in output.
- Fixed non-linear conjugate heat solver.

- **Implicit**

  - Fixed Implicit for the case of Multi-step Linear (*CONTROL_IMPLICIT_GENERAL with NSOLVR = 1) with Intermittent Eigenvalue Computation (*CONTROL_IMPLICIT_EIGENVALUE with NEIG < 0).
  - Recent fix for resultant forces for Multi-step Linear cause segmentation fault when Intermittent Eigenvalue Computation was also active.
  - Fix possible issue related to constrained contacts in MPP implicit not initializing properly.
  - Fixed label at beginning of implicit step to be correct for the case of controlling implicit dynamics via a load curve (*CONTROL_IMPLICIT_DYNAMICS).
  - Corrected the computation of modal stresses with local coordinate terms and forsome shell elements (see MSTRES on *CONTROL_IMPLICIT_EIGENVALUE).
  - Corrected *CONTROL_IMPLICIT_INERTIA_RELIEF logic in MPP. In some cases the rigid body modes were lost.
  - Enhanced implicit's treatment of failing spotwelds (*CONSTRAINED_SPOTWELD).

- Added additional error checking of input data for *CONTROL_IMPLICIT_MODAL_DYNAMICS_DAMPING.
- Per user request we added the coupling of prescribed motion constraints for Modal Dynamics by using constraint modes. See *CONTROL_IMPLICIT_MODAL_DYNAMIC.
- Added reuse of the matrix reordering for MPP implicit execution. This will reduce the symbolic processing time which is noticable when using large numbers of MPP processes. Also added prediction of non tied contact connections for standard contact and mortar contact. This allows re-use of the ordering when contact interfaces are changing very slightly but can increase the cost of the numerical factorization. Useful only for MPP using large numbers of processes for large finite element models. This re-use checking happens automatically for MPP and is not required for SMP.
- Apply improvements to Metis memory requirements used in Implicit MPP.
- Enhanced Metis ordering software (ORDER = 2, the default, on *CONTROL_IMPLICIT_SOLVER).
- Added new keyword *CONTROL_IMPLICIT_ORDERING to control of features of the ordering methods for the linear algebra solver in MPP Implicit. Only should be used by expert users.
- The following 4 enhancements are applicable when IMFLAG > 1 on *CONTROL_IMPLICIT_GENERAL.

    - Implicit was modified to reset the time step used in contact when switching from implicit to explicit.
    - Adjusted implicit mechanical time step for the case of switching from explicit to implicit so as not to go past the end time.
    - Explicit with intermittent eigenvalue analysis was getting incorrect results after the eigenvalue analysis because an incorrect time step was used for the implicit computations. For this scenario implicit now uses the explicit time step.
    - The implicit time step is now reset for the dump file in addition to explicit's time.

- Implicit's treatment of prescribed motion constraints defined by a box had to be enhanced to properly handle potential switching to explicit.
- The following 6 enhancements are for matrix dumping (MTXDMP > 0 on *CONTROL_IMPLICIT_SOLVER) or for frequency response (*FREQUENCY_DOMAIN) computations.
- Corrected the collection of *DAMPING_PART_STIFFNESS terms for elements like triangles and 5, 6, and 7 node solid elements.
- Corrected Implicit's access of *DAMPING_PART_STIFFNES parameter when triangle and tet sorting is activated.
- Fixed Implicit's collecting of damping terms for beams that have reference nodes.

- There is an internal switch that turns off damping for beams if the run is implicit static. This switch needed to be turned off for explicit with intermittent eigenvalue analysis.
- Fixed collecting of stiffness damping terms for implicit. Corrected the loading of mass damping terms when collecting damping terms for post processing.
- Extend matrix dumping to include dumping the solution vector in addition to the matrix and right-hand-side.
- Adjusted Implicit's handling of sw1. and sw3. sense switches to properly handle dumping. If sw1. sense switch is issued when not at equilibrium, then reset time and geometry to that at the end of last implicit time step. If sw3. sense switch is issued, then wait until equilibrium is reached before dumping and continuing.
- Enable the use of intermittent eigenvalue computation for models using inertia relief and/or rotational dynamics. See NEIG < 0 on *CONTROL_IMPLICIT_EIGENVALUE and *CONTROL_INERTIA_RELIEF and *CONTROL_IMPLICIT_ROTATIONAL_DYNAMICS. Due to round-off, an implicit intermittent eigenvalue computation was occasionally skipped. A fudge factor of 1/1000 of the implicit time step was added to compensate for round-off error in the summation of the implicit time. See NEIG < 0 on *CONTROL_IMPLICIT_EIGENVALUE.
- Added support for *CONSTRAINED_LINEAR for 2D implicit problems. It was already supported for standard 3D problems.
- Added warning for implicit when the product of ILIMIT and MAXREF (two parameters on *CONTROL_IMPLICIT_SOLUTION) is too small. For the special case when the user changes the default of ILIMIT to 1 to choose Full Newton and does not change MAXREF then MAXREF is reset to 165 and a warning is generated. Reinstate the option of MAXREF < 0.
- Fixed the display of superelements in LS-PrePost. Enhanced reading of Nastran dmig files to allow for LS-DYNA-like comment lines starting with '$'. Fixed a problem with implicit initialization in MPP with 2 or more superelements. See *ELEMENT_DIRECT_MATRIX_INPUT.
- Turned off annoying warning messages associated with zero contact elemental stiffness matrices coming from mortar contact. See *CONTACT_..._MORTAR
- Fixed construction of d3mode file in MPP. Involves proper computation of the reduced stiffness matrix. See *CONTROL_IMPLICIT_MODES
- Fixed up *PART_MODES to correctly handle constraint modes.

  - removed rigid body modes
  - correct construction of reduced stiffness matrix

- Enhanced the error handling for input for *PART_MODES.
- Modified open statements for binary files used by implicit to allow for use of *CASE.

- Removed internal use files such as spooles.res when not required for debugging.
- Fixed implicit static condensation and implicit mode computation to properly deal with the *CASE environment. See *CONTROL_IMPLICIT_STATIC_CONDENSATION and *CONTROL_IMPLICIT_MODES. Sort node/dof sets for implicit_mode to get correct results. Properly handle cases with only solid elements.
- Add implicit implementation of the new "last location" feature for MPP error tracking.
- Fixed problem with implicit processing of rigid body data with deformable to rigid switching (*DEFORMABLE_TO_RIGID).
- Extended Implicit model debugging for LPRINT = 3 (*CONTROL_IMPLICIT_SOLVER) to isogeometric and other large elemental stiff matrices.
- Added beam rotary mass scaling to the modal effective mass computation. Enhanced implicit computation of modal effective mass that is output to file eigout with *CONTROL_IMPLICIT_EIGENVALUE. We had to account for boundary SPC constraints as well as beam reference nodes to get the accumulated percentage to add up to 100%.
- Fixed a problem reporting redundant constraints for MPP Implicit.
- Enhanced *CONTACT_AUTO_MOVE for implicit.
- Fixed Implicit handling of *CONSTRAINED_TIE-BREAK in MPP.
- Added support for implicit dynamics to *MAT_157 and *MAT_120.
- Skip frequency damping during implicit static dynamic relaxation.
- Added feature to simulate brake squeal. Transient and mode analysis can be combined to do the brake squeal study by intermittent eigenvalue analysis. Besides *CONTROL_IMPLICIT_ROTATIONAL_DYNAMICS, *CONTROL_IMPLICIT_SOLVER should also be used, setting LCPACK = 3 to enable unsymmetric stiffness matrix. In the non-symmetric stiffness matrix analysis such as brake squeal analysis, the damping ratio, defined as -2.0*RE(eigenvalue)/ABS(IMG(eigenvalue)), can be output to the eigout file and plotted in LS-PrePost. A negative damping ratio indicates an unstable mode.
- Add a warning message if the defined rotational speed is not the same as NOMEG in *CONTROL_IMPLICIT_ROTATIONAL_DYNAMICS.
- *CONTROL_IMPLICIT: Fixed a bug to initialize velocity correctly when using a displacement file in dynamic relaxation for implicit MPP.
- Nonlinear implicit solver 12 is made default implicit solver, which is aimed for enhanced robustness in particular relation to BFGS and line search.
- Parameter IACC available on *CONTROL_ACCURACY to invoke enhanced accuracy in selected elements, materials and tied contacts. Included is strong objectivity in the most common elements, strong objecitity and physical respons in most commont tied contacts and full iteration plasticity in *MATs 24 and 123. For more detailed information refer to the manual.

- Bathe composite time integration scheme implemented for increased stability and conservation of energy/momentum, see *CONTROL_IMPLICIT_DYNAMICS. Time integration parameter ALPHA on CONTROL_IMPLICIT_DYNAMICS is used for activation.
- For NLNORM.LT.0 all scalar products in implicit are with respect to all degrees of freedom, sum of translational and rotational (similar to NLNORM.EQ.4), just that the rotational dofs are scaled using ABS(NLNORM) as a characteristic length to appropriately deal with consistency of units.
- The message 'convergence prevented due to unfulfilled bc...' has annoyed users. Here this is loosened up a little and also accompanied with a check that the bc that prevents convergence is actually nonzero. Earlier this prevention has activated even for SPCs modelled as prescribed zero motion, which does not make sense.
- Implicit now writes out the last converged state to the d3plot database on error termination if not already written.
- Fixed bug for *CONTROL_IMPLICIT_MODAL_DYNAMIC if jobid is used.

- ***INITIAL**

  - Fix incorrect NPLANE and NTHICK for *INITIAL_STRESS_SHELL when output to dynain file for shell type 9.
  - Fix *INITIAL_STRAIN_SHELL output to dynain for shell types 12 to 15 in 2D analysis.
  - Write out strain at only 1 intg point if INTSTRN = 0 in *INTERFACE_SPRINGBACK_LSDYNA and all strains at all 4 intg points if INTSTRN = 1 and nip = 4 in *SECTION_SHELL.
  - *INITIAL_EOS_ALE: Allow initialization of internal energy density, relative volume, or pressure in ALE elements by part, part set, or element set.
  - *INITIAL_VOLUME_FRACTION_GEOMETRY: Add option (FAMMG < 0) to form pairs of groups in *SET_MULTI-MATERIAL_GROUP_LIST to replace the first group of the pair by the second one.
  - *INITIAL_STRESS_DEPTH can now work with parts that have an Equation of State (EOS types 1, 4, 6 only). Note however that *INITIAL_STRESS_DEPTH does not work with ALE.
  - Fix several instances of overwriting the initial velocities of any interface nodes read in from a linking file (SMP only).
  - *INITIAL_VOLUME_FRACTION_GEOMETRY: Add local coordinate system option for box.
  - The initial strain and energy is calculated for *INITIAL_FOAM_REFERENCE_GEOMETRY.
  - Add the option of defining the direction cosine using two nodes for *INITIAL_VELOCITY_GENERATION.

- ◦ Fix incorrect transformation of *DEFINE_BOX which results in incorrect initial velocities if the box is used in *INITIAL_VELOCITY.
- ◦ Fix incorrect initial velocity when using *INITIAL_VELOCITY with NX = -999.
- ◦ Fix seg fault when using *INITIAL_INTERNAL_DOF_SOLID_TYPE4 in dynain file.
- ◦ Do not transform the translational velocities in *INITIAL_VELOCITY or *INITIAL_VELOCITY_GENERATION if the local coordinate system ICID is defined.
- ◦ Fix uninitialized velocities when using *INITIAL_VELOCITY_ GENERATION with STYP = 2, i.e. part id, for *ELEMENT_SHELL_ COMPOSITE/*ELEMENT_TSHELL_COMPOSITE.
- ◦ Fix incorrect initialization of velocities if using *INITIAL_VELOCITY_ GENERATION with STYP = 1, i.e. part set for shells with formulation 23 & 24.
- ◦ Fix incorrect initial velocity and also mass output to d3hsp for shell types 23 & 24.
- ◦ Fix incorrect initial velocities when using *INITIAL_VELOCITY_ GENERATION with irigid = 1 and *PART_INERTIA with xc = yc = zc = 0 and nodeid > 0 with *DEFINE_TRANSFORMATION.
- ◦ Fix incorrect stress initialization of *MAT_057/MAT_LOW_DENSITY_ FOAM using dynain file with *INITIAL_STRESS_SOLID when NHISV is equal to the number of history variables for this mat 57.
- ◦ Fix seg fault when reading dynain.bin
- ◦ Fixed stress initialization (*INITIAL_STRESS_SECTION) for type 13 tetrahedral elements. The pressure smoothing was causing incorrect pressure values in the elements adjacent to the prescribed elements.
- ◦ Assign initial velocities (*INITIAL_VELOCITY) to beam nodes that are generated when release conditions are defined (RT1, RT2, RR1, RR2 on *ELEMENT_BEAM.)
- ◦ Added an option to retain bending stiffness in spot weld beams that have prescribed axial force. To use is, set KBEND = 1 on *INITIAL_AXIAL_ FORCE_BEAM.
- ◦ Fix for *INITIAL_STRESS_BEAM when used with spotweld beam type 9. It was possible that error/warning message INI+140 popped up even if number of integration points matched exactly.
- ◦ Fix for the combination of type 13 tet elements and *INITIAL_STRESS_ SOLID. The necessary nodal values for averaging (element volume, Jacobian) were not correctly initialized. Now the initial volume (IVEFLG) is used to compute the correct initial nodal volume.

- **Isogeometric Elements**

  - ◦ Enable spc boundary condition to be applied to extra nodes of nurbs shell, see *CONSTRAINED_NODES_TO_NURBS_SHELL

  - Fix a bug for isogeometric element contact, IGACTC = 1, that happens when more than one NURBS patches are used to model a part so that a interpolated elements have nodes belonging to different NURB patches.
  - *ELEMENT_SOLID_NURBS_PATCH:
    - Enable isogeometric analysis for solid elements, it is now able to do explicit and implicit analysis, such as contact and eigenvalue analysis, etc.
    - Add mode stress analysis for isogeometric solid and shell elements so that the isogeometric element is also able to do frequency domain analysis.
  - Add reduced, patch-wise integration rule for C1-continuous quadratic NURBS. This can be used by setting INT = 2 in *ELEMENT_SHELL_NURBS_PATCH.
  - Add trimmed NURBS capability. Define NL trimming loops to specify a trimmed NURBS patch. Use *DEFINE_CURVE (DATTYP = 6) to specify define trimming edges in the parametric space.
  - Fix bug in added mass report for *ELEMENT_SHELL_NURBS_PATCH in MPP.

- ***LOAD**
  - *LOAD_GRAVITY_PART and staged construction (*DEFINE_STAGED_CONSTRUCTION_PART) were ignoring non-structural mass MAREA (shells) and NSM (beams). Now fixed.
  - Fix for *INTERFACE_LINKING in MPP when used with adaptivity.
  - Updates for *INTERFACE_LINKING so that it can be used with adaptivity, provided the linked parts are adapting.
  - Fix for *INTERFACE_LINKING when used with LSDA based files generated by older versions of the code.
  - *DEFINE_CURVE_FUNCTION:
    - Functions "DELAY", PIDCTL" and "IF" of are revised.
    - Add sampling rate and saturation limit to PIDCTL of *DEFINE_CURVE_FUNCTION.
    - "DELAY" of *DEFINE_CURVE_FUNCTION can delay the value of a time-dependent curve by "-TDLY" time steps when TDLY < 0.
  - Add edge loading option to *LOAD_SEGMENT_SET_NONUNIFORM.
  - Fix insufficient memory error,SOL+659, when using *LOAD_ERODING_PART_SET with mpp.
  - Fix incorrect loading when using *LOAD_ERODING_PART_SET with BOXID defined.

  - Fix incorrect pressure applied if the directional cosines, V1/V2/V3, for *LOAD_SEGMENT_SET_NONUNIFORM do not correspond to a unit vector.
  - Add *DEFINE_FUNCTION capability to *LOAD_SEGMENT_SET for 2D analysis.
  - Fix incorrect behavior when using arrival time, AT, or box, BOXID, in *LOAD_ERODING_PART_SET.
  - Fix error when runing analysis with *LOAD_THERMAL_CONSTANT_ELEMENT_(OPTION) in MPP with ncpu > 1.
  - Fixed *LOAD_STEADY_STATE_ROLLING when used with shell form 2 when used with Belytschko-Wong-Chang warping stiffness (BWC = 1 *CONTROL_SHELL).
  - Add "TIMESTEP" as a code defined value available for *DEFINE_FUNCTION and *DEFINE_CURVE_FUNCTION. It holds the current simulation timestep.
  - Fixed issues involving *LOAD_THERMAL_D3PLOT.
  - Allow extraction of node numbers in loadsetud for all values of LTYPE in *USER_LOADING_SET. Comments included appropriately in the code. Argument list of loadsetud is changed accordingly.
  - Implemented SPF simulation (*LOAD_SUPERPLASTIC_FORMING) for 2d problems.
  - Added effective stress as target variable for SPF simulation.
  - Added box option for SPF simulation to limit target search regions.

- ***MAT**

  - Fix output to d3hsp for *MAT_HYPERELASTIC_RUBBER. Broken in r93028.
  - Error terminate with message, KEY+1115, if_STOCHASTIC option is invoked for *MATs 10,15,24,81,98, 123 but no *DEFINE_STOCHASTIC_VARIATION or *DEFINE_HAZ_PROPERTIES keyword is present in the input file.
  - Fix spurious error termination when using *DEFINE_HAZ_PROPERTIES with adaptivity.
  - Fixed *MATs 161 and 162 when run with MPP. The array that is used to share delamination data across processors had errors.
  - *MAT_261/*MAT_262: Fixed problem using *DAMPING_PART_STIFFNESS together with RYLEN = 2 in *CONTROL_ENERGY.
  - Added safety check for martensite phase kinetics in *MAT_244.
  - Fix for combination of *MAT_024_STOCHASTIC and shell elementstype 13, 14, and 15 (with 3d stress state).
  - Fix bug in *MATs 21 and 23 when used with *MAT_ADD_THERMAL_EXPANSION.
  - *MAT_ALE_VISCOUS: Implement a user defined routine in dyn21.F to compute the dynamic viscosity.

- Add histlist.txt to usermat package. This file lists the history variables by material.
- Bug in *MAT_089 fixed: The load curve LCSS specifies the relationship between "maximum equivalent strain" and the von Mises stress. The "maximum equivalent strain" includes both elastic and plastic components. The material model was not calculating this variable as intended, so was not following LCSS accurately. The error was likely to be more noticeable when elastic strains are a significant proportion of the total strain e.g. for small strains or low initial Youngs modulus.
- Fixed bug affecting *MAT_119: unpredictable unloading behaviour in local T-direction if there are curves only for the T-direction and not for the S-direction.
- Fixed bug in *MAT_172: Occured when ELFORM = 1 (Hughes-Liu shell formulation) was combined with Invariant Numbering (INN > 0 on *CONTROL_ACCURACY). In this case, the strain-softening in tension did not work: after cracking, the tensile strength remained constant.
- New option for *MAT_079: Load curve LCD defining hysteresis damping versus maximum strain to date. This overrides the default Masing behaviour.
- *MAT_172:
  - Added error termination if user inputs an illegal value for TYPEC. Previously, this condition could lead to abnormal terminations that were difficult to diagnose.
  - Fixed bug affecting ELFORM = 16 shells made of *MAT_172 – spurious strains could develop transverse to the crack opening direction.
- Fixed bug in *MAT_ARUP_ADHESIVE (*MAT_169). The displacement to failure in tension was not as implied by the inputs TENMAX and GCTEN. For typical structural adhesives with elastic stiffness of the order of 1000-10000 MPa, the error was very small. The error became large for lower stiffness materials.
- *MAT_SPR_JLR:
  - Modify output variables from *MAT_SPR_JLR (see Manual).
  - Fix bug that caused spurious results or unexpected element deletion if TELAS = 1.
- Fixed bug in *MAT_174 - the code could crash when input parameters EUR = 0 and FRACR = 0..
- Fix MPP problem when writing out aea_crack file for *MAT_WINFRITH.
- Include *MAT_196 as one that triggers spot weld thinning.
- *MAT_ADD_FATIGUE: Implemented multi slope SN curves to be used in random vibration fatigue (*FREQUENCY_DOMAIN_RANDOM_

VIBRATION_FATIGUE) and SSD fatigue (FREQUENCY_DOMAIN_SSD_FATIGUE).

- Guard against possible numerical round off that in some cases might result in unexpected airflow in *MAT_ADD_PORE_AIR.
- Added new material *MAT_115_O/*MAT_UNIFIED_CREEP_ORTHO.
- *MAT_274: Added support for 2D-solids. New flag (parameter 8 on card 2) is used to switch normal with in-plane axis.
- *MAT_255: Fixed bug in plasticity algorithm and changed from total strain rate to plastic strain rate for stability. Added VP option (parameter 5 on card 2) for backwards compatibility: VP = 0 invokes total strain rate used as before.
- Added new cohesive material *MAT_279/*MAT_COHESIVE_PAPER to be used in conjunction with *MAT_274/*MAT_PAPER.
- User materials: Added support for EOS with user materials for tshell formulations 3 and 5.
- Fixed bug in dyna.str when using EOS together with shells and orthotropic materials.
- *MAT_122: A new version of *MAT_HILL_3R_3D is available. It supports temperature dependent curves for the Young's/shear moduli, Possion ratios, and Hill's anisotropy parameters. It also supports 2D-tables of yield curves for different temperatures. Implicit dynamics is supported. The old version is run if parameter 5 on card 3 is set to 1.0.
- Added the phase change option to *MAT_216, *MAT_217, *MAT_218 to allow material properties to change as a function of location. This capability is designed to model materials that change their properties due to material processing that is otherwise not modeled. For example, increasing the mass and thickness due to the deposition of material by spraying. It is not used for modeling phase changes caused by pressure, thermal loading, or other mechanical processes modeled within LS-DYNA.
- Fix internal energy computation of *MAT_ELASTIC_VISCOPLASTIC_THERMAL/MAT_106.
- Fix incorrect results or seg fault for *MAT_FU_CHANG_FOAM/MAT_083 if KCON > 0.0 and TBID.ne.0.
- If SIGY = 0 and S = 0 in *MAT_DAMAGE_2/MAT_105, set S = EPS1/200, where EPS1 is the first point of yield stress input or the first ordinate point of the LCSS curve.
- Set xt = 1.0E+16 as default if user inputs 0.0 for *MAT_ENHANCED_COMPOSITE_DAMAGE/MAT_054. Otherwise, random failure of elements may occur. Implemented for thick shells and solids.
- Allow *MAT_ENHANCED_COMPOSITE_DAMAGE/MAT_054 failure mechanism to work together with *MAT_ADD_EROSION for shells.
- Fix incorrect erosion behavior if *MAT_ADD_EROSION is used with failure criteria defined for *MAT_123/MAT_MODIFIED_PIECEWISE_LINEAR_PLASTICITY.

- Fix non-failure of triangular elements type 4 using *MAT_ADD_EROSION with NUMFIP = -100.
- Implement scaling of failure strain for *MAT_MODIFIED_PIECEWISE_LINEAR_PLASTICITY_STOCHASTIC/MAT_123_STOCHASTIC for shells.
- Fix incorrect behavior for *MAT_LINEAR_ELASTIC_DISCRETE_BEAM/MAT_066 when using damping with implicit(statics) to explicit switching.
- Fix error due to convergence when using *MAT_CONCRETE_EC2/MAT_172 in implicit and when FRACRX = 1.0 or FRACRY = 1.0
- Fix incorrect fitting results for *MAT_OGDEN_RUBBER/MAT_077_O if the number of data points specifed in LCID is > 100.
- Fix incorrect fitting results for *MAT_MOONEY-RIVLIN-RUBBER/MAT_027 if the number of data points specifed in LCID is > 100.
- Fix incorrect forces/moments when preloads are used for *MAT_067/NONLINEAR_ELASTIC_DISCRETE_BEAM and the strains changes sign.
- Implement *MAT_188/MAT_THERMO_ELASTO_VISCOPLASTIC_CREEP for 2D implicit analysis.
- Support implicit for *MAT_121/MAT_GENERAL_NONLINEAR_1DOF_DISCRETE_BEAM.
- Fix seg fault when using *DEFINE_HAZ_TAILOR_WELDED_BLANK with *DEFINE_HAZ_PROPERTIES.
- Fix ineffective *MAT_ADD_EROSION if the MID is defined using a alpha-numeric label.
- Fix seg fault when using *MAT_PIECEWISE_LINEAR_PLASTIC_THERMAL/MAT_255 for solids.
- Zero the pressure for *MAT_JOHNSON_HOLMQUIST_JH1/MAT_241 after it completely fractures, i.e. D>=1.0, under tensile load.
- Fix incorrect element failure when using EPSTHIN and VP = 0 for *MAT_123/MODIFIED_PIECEWISE_LINEAR_PLASTICITY.
- Fix error termination when using adaptive remeshing for 2D analysis with *MAT_015/JOHNSON_COOK and NIP = 4 in *SECTION_SHELL and ELFORM = 15.
- Fix erosion due to damage, max shear & critical temperature in elastic state for *MAT_MODIFIED_JOHNSON_COOK/MAT_107 for solids.
- Check diagonal elements of C-matrix of *MAT_002/MAT_{OPTION}TROPIC_ELASTIC and error terminate with message, STR+1306, if any of them are negative.
- Fix plastic strain tensor update for *MAT_082/*MAT_PLASTICITY_WITH_DAMAGE.
- Fix error when using *MAT_144/MAT_PITZER_CRUSHABLE_FOAM with solid tetahedron type 10.
- Fix out-of-range forces after dynamic relaxation when using VP = 1 for *MAT_PIECEWISE_LINEAR_PLASTICITY and non-zero strain rate pa-

rameters, C & P, and the part goes into plastic deformation during dynamic relaxation.

- Fixed unit transformation for GAMAB1 and GAMAB2 on *MAT_DRY_FABRIC. We were incorrectly transforming them as stress.
- Fixed implicit solutions with shell elements that use *MAT_040 and laminated shell theory.
- Fixed the stress calculation in the thermal version of *MAT_077.
- Corrected the AOPT = 0 option of ortho/anisotropic materials when use with skewed solid elements. Previously, the material direction was initialized to be equivalent to the local coordinate system direction. This is not consistent with the manual for skewed elements which states that the material a-axis is in the 1-2 directions for AOPT = 0. This is now fixed and the manual is correct.
- Fixed the AOPT = 0 option of ortho/anisotropic materials for tetrahedral element forms 10, 13, and 44.
- Fixed *MAT_082 for solid elements. An error in the history data was causing possible energy growth or loss of partially damaged elements.
- Modified *MAT_FABRIC/*MAT_034 FORM = 24 so that Poisson's effects occur in tension only.
- Modified *MAT_221/*MAT_ORTHOTROPIC_SIMPLIFIED_DAMAGE to correct the damage behavior. Prior to this fix, damage was applied to new increments of stress, but not the stress history, so material softening was not possible.
- Fixed *MAT_106 when used with curves to define the Young's modulus and Poisson's ratio and when used with thick shell form 5 or 6. The assumed strain field was unreasonable which caused implicit convergence to fail.
- Added 2 new erosion criteria for *MAT_221/*MAT_ORTHOTROPC_SIMPLIFIED_DAMAGE. The new options are NERODE = 10: a or b directions failure (tensile or compressive) plus out of plane failure bc or ca. NERODE = 11: a or b directions failure (tensile only) plus out of plane failure bc or ca.
- Added a new option for shell *MAT_022/*MAT_COMPOSITE_DAMAGE. When ATRACK = 1, the material directions will follow not only element rotation, but also deformation. This option is useful for modeling layered composites, that have material a-directions that vary by layer, by allowing each layer to rotate independently of the others. Within each layer, the b-direction is always orthogonal to the a-direction.
- Fixed the TRUE_T option on *MAT_100 and *MAT_100_DA. If the weld connects shells with different thickness and therefore different bending stiffness, the scheme used by TRUE_T to reduce the calculated moment could behave somewhat unpredictably. With the fix, TRUE_T behaves much better, both for single brick welds and brick assemblies.

- Added a warning message and automatically switch DMGOPT > 0 to DMGOPT = 0 on *MAT_FABRIC when RS < EFAIL or RS = EFAIL. This prevents a problem where weld assemblies did not fail at all when RS = 0.
- *MATs 9, 10, 11, 15, 88, and 224 are now available for thick shells, however only *MATs 15, 88, and 224 are available for the 2D tshell forms 1,2, and 6.
- Added thick shell support for the STOCHASTIC option of *MATs 10, 15, 24, 81, and 98.
- Added support for *MAT_096 for several solid element types including ELFORMs 3, 4, 15, 18, and 23.
- Added a MIDFAIL keyword option for *MAT_024, (MAT_PIECEWISE_LINEAR_PLASTICITY). With this option, element failure does not occur until the failure strain is reached in the mid plane layer. If an even number of layers is used, then the failure occurs when the 2 closest points reach the failure strain.
- Enabled *MATs 26 and 126 (HONEYCOMB) to be used with thick shell forms 3, 5, and 7. These was initialized incorrectly causing a zero stress.
- Enabled *MAD_ADD_EROSION to be used with beams that have user defined integration. Memory allocation was fixed to prevent memory errors.
- Enabled OPT = -1 on *MAT_SPOTWELD for solid elements.
- Enabled thick shells to use *MATs 103 and 104 in an implicit solution. These materials were lacking some data initialization so they would not converge.
- Enabled solid elements with user-defined orthotropic materials to work with the INTOUT and NODOUT options on *DATABASE_EXTENT_BINARY. The transformation matrix was stored in the wrong place causing strain and stress transformations to fail.
- Enabled *MAT_017 to run with thick shell forms 3 and 5. Neither element was initialized correctly to run materials with equations of state.
- Add degradation factors and strain rate dependent strength possibility for *MAT_054/*MAT_ENHANCED_COMPOSITE_DAMAGE solids.
- Fixed bug in *MAT_058/*MAT_LAMINATED_COMPOSITE_FABRIC when used with strain-rate dependent tables for stiffnesses EA, EB and GAB and LAMSHT = 3.
- Add strain rate dependency of ERODS in *MAT_058.
- Add possibility to use *DEFINE_FUNCTION for *MAT_SPOTWELD_DAMAGE_FAILURE (*MAT_100), OPT = -1/0. If FVAL = FunctionID, then a *DEFINE_FUNCTION expression is used to determine the weld failure criterion using the following arguments: func (N_rr, N_rs, N_rt, M_rr, M_ss, M_tt).
- Store tangential and normal separation (delta_II & delta_I) as history variables 1&2 of *MAT_138/*MAT_COHESIVE_MIXED_MODE.
- Add second normalized traction-separation load curve (TSLC2) for Mode II in *MAT_186/*MAT_COHESIVE_GENERAL.

- Fixed bug in using *MAT_157/*MAT_ANISOTROPIC_ELASTIC_PLASTIC with IHIS.gt.0 for shells. Thickness strain update d3 was not correct and plasticity algorithm failed due to typo.
- Fixed bug in *MAT_157 for solids: This affected the correct stress transformation for post-processing using CMPFLG = 1 in *DATABASE_EXTENT_BINARY.
- Fixed bug in *MAT_225 (*MAT_VISCOPLASTIC_MIXED_HARDENING) when using Table-Definition together with kinematic hardening.
- Add load curves for rate dependent strengths (XC, XT, YC, YT, SC) in *MAT_261/*MAT_LAMINATED_FRACTURE_DAIMLER_PINHO (shells only).
- Add table definition for LCSS for rate dependency in *MAT_261 (shells only).
- Add load curves for rate dependent strengths (XC, XCO, XT, XTO, YC, YT, SC) in *MAT_262/*MAT_LAMINATED_FRACTURE_DAIMLER_CAMANHO (shells only).
- Fixed bug when using *MAT_261 or *MAT_262 solids (ELFORM = 2).
- Add load curves for SIGY and ETAN for rate dependency of *MAT_262 (shells only)
- *MAT_021_OPTION

    - Fixed a bug for defining different orientation angles through the thickness of TSHELL elements (formulations 2 and 3)
    - Added new option CURING:
    - Two additional cards are read to define parameters for curing kinetics. Formulation is based on Kamal's model and considers one ODE for the state of cure.
    - State of cure does not affect the mechanical parameters of the material.
    - CTE's for othotropic thermal expansione can be defined in a table with respect to state of cure and temperature.
    - An orthotropic chemical shrinkage is accounted for.

- *MAT_REINFORCED_THERMOPLASTICS_OPTION (*MAT_249_OPTION):

    - Fiber shear locking can be defined wrt to the fiber angle or shear angle.
    - Output of fiber angle to history variables.
    - Simplified input: Instead of always reading 8 lines, now the user only has to specify data for NFIB fibers.
    - Added fiber elongation to history variables in *MAT_249 for posprocessing.
    - New Option UDFIBER (based on a user defined material by BMW):

        - Transversely isotropic hyperelastic formulation for each fiber family (see Bonet&Burton,1998).
        - Anisotropic bending behavior based on modified transverse shear stiffnesses.
        - Best suited for dry NCF's.

- *MAT_GENERALIZED_PHASE_CHANGE (*MAT_254):

    - New material that is a generalized version of *MAT_244 with application to a wider range of metals.
    - Up to 24 different phases can be included.
    - Between each of the phases, the phase transformation can be defined based on a list of generic transformation laws. For heating JMAK and Oddy are implemented. For cooling Koistinen-Marburger, JMAK and Kirkaldy can be chosen.
    - Constant parameters for the transformations are given as 2d tables, parameters depending on temperature (rate) or phase concentration employ 3d tables.
    - Plasticity model (temperature and strain rate dependent) similar to MAT_244.
    - Transformation induced strains.
    - TRIP algorithm included.
    - Temperature dependent mixture rules.
    - Parameter 'dTmax' that defines the maximum temperature increment within a cycle. If the temperature difference at a certain integration point is too high, local subcycling is performed.
    - Implemented for explicit/implicit analysis and for 2d/3d solid elements.

- *MAT_ADHESIVE_CURING_VISCOELASTIC (*MAT_277):

    - New material implementation including a temperature dependent curing process of epoxy resin based on the Kamal-Sourour-model.
    - Material formulation is based on *MAT_GENERAL_VISCOELASTIC.
    - Viscoelastic properties defined by the Prony series, coefficients as functions of state of cure.
    - Chemical and thermal shrinkage considered (differential or secant formulations).
    - Available for shell and solid elements.
    - Can be used in combination with *MAT_ADD_COHESIVE.
    - Implemented for explicit and implicit analysis.
    - An incremental and a total stress calculation procedure available.

- Enable *MAT_ADD_EROSION to be safely used with material models that have more than 69 history variables, for now the new limit is 119.

- Use correct element ID for output of failed solid elements when GISSMO (*MAT_ADD_EROSION) is used with *CONTROL_DEBUG.
- Improve performance of GISSMO (*MAT_ADD_EROSION with IDAM = 1), especially when used with *MAT_024, no other failure criteria, shell elements, and DMGEXP = 1 or 2. Allows speed-up of 10 to 20 percent.
- Add new keyword *MAT_ADD_GENERALIZED_DAMAGE. It provides a very flexible approach to add non-isotropic (tensorial) damage to standard materials in a modular fashion. Solely works with shell elements at the moment.
- Correct the computation of effective strain for options ERODS < 0 in *MAT_058 (*MAT_LAMINATED_COMPOSITE_FABRIC) and EFS < 0 in *MAT_261 and *MAT_262 (*MAT_LAMINATED_FRACTURE_ DAIMLER_...). The shear strain term was twice the size as it should have been.
- Adjust stiffness for time step calculation in *MAT_076 and subsequent models (*MAT_176, *MAT_276, ...) to prevent rarely observed instabilities.
- Add output of original and fitted curves to messag and separate file (curveplot_<MID>) for *MAT_103.
- In *MAT_104 (*MAT_DAMAGE_1), stress-strain curve LCSS can now be used directly with all FLAG options (-1,0,1,10,11), no fitting.
- Correct strain calculation for anisotropic damage in *MAT_104 (*MAT_ DAMAGE_1) with FLAG = -1.
- Initialize stress triaxiality of *MAT_107 (*MAT_MODIFIED_JOHNSON_ COOK) to zero instead of 1/3.
- Avoid negative damage in *MAT_107 (*MAT_MODIFIED_JOHNSON_ COOK) with FLAG2 = 0 for solid elements.
- Rectify the characteristic element length in *MAT_138 (*MAT_COHESIVE_ MIXED_MODE) for solids type 21 and 22 (cohesive pentas) and shell type 29 (cohesive shell) for "curve" options T < 0 and S < 0.
- Correct/improve material tangent for *MAT_181 with PR > 0 (foam option).
- Add possibility to define logarithmically defined strain rate table LCID-T in material *MAT_187 (*MAT_SAMP-1).
- Fix missing offset when using *DEFINE_TRANSFORMATION with load curve LCID-P in *MAT_187 (*MAT_SAMP-1).
- Add reasonable limit for biaxial strength in *MAT_187 with RBCFAC > 0.5 to avoid concave yield surface.
- Improve performance of *MAT_187 to reach speed-up of 10 to 40 percent, depending on which options are used.
- Add new option for *MAT_224 (*MAT_TABULATED_JOHNSON_COOK). With BETA < 0 not only a load curve but now also a table can be referred to. The table contains strain rate dependent curves, each for a different temperature.

- Fix for implicit version of *MAT_224 (*MAT_TABULATED_JOHNSON_COOK). Computations with shell elements should converge faster now.
- *MAT_224 (*MAT_TABULATED_JOHNSON_COOK) can now be used in implicit even with temperature dependent Young's modulus (parameter E < 0).
- Always store the Lode parameter as history variable #10 in *MAT_224 (*MAT_TABULATED_JOHNSON_COOK), not just for LCF being a table.
- Variable LCI of *MAT_224 / *MAT_224_GYS can now refer to a *DEFINE_TABLE_3D. That means the plastic failure strain can now be a function of Lode parameter (TABLE_3D), triaxiality (TABLE), and element size (CURVE).
- For thick shells type 1 and 2, the element size in *MAT_224 is now correct.
- Add new option for definition of parameters FG1 and FG2 in *MAT_240 (*MAT_COHESIVE_MIXED_MODE_ELASTOPLASTIC_RATE).
- Add new option to *MAT_240: new load curves LCGIC and LCGIIC define fracture energies GIC and GIIC as functions of cohesive element thickness. GIC_0, GIC_INF, GIIC_0, and GIIC_INF are ignored in that case.
- Add new feature to *MAT_248 (*MAT_PHS_BMW). Estimated Hocket-Sherby parameters are written to history variables based on input functions and phase fractions.
- Add new option ISLC = 2 to *MAT_248 (*MAT_PHS_BMW) which allows to define load curves (cooling rate dependent values) for QR2, QR3, QR4, and all parameters on Cards 10 and 11.
- Add new option LCSS to *MAT_252 (*MAT_TOUGHENED_ADHESIVE_POLYMER): A load curve, table or 3d table can now be used to define rate and temperature dependent stress-strain behavior (yield curve).
- Fix for *MAT_255, evaluation of 2d tables LCIDC and LCDIT. Negative temperatures were interpreted as logarithmic rates.
- Add new material model *MAT_280 (*MAT_GLASS) for shell elements. It is a smeared fixed crack model with a selection of different brittle, stress-state dependent failure criteria and crack closure effects.
- *DEFINE_FABRIC_ASSEMBLIES: Assemblies of *MAT_FABRIC part sets can be specified to properly treat bending of t-intersecting fabrics that are stitched or sewn together. See ECOAT, TCOAT and SCOAT on *MAT_FABRIC_... Bending can only occur within an assembly, aka a part set.
- *MAT_USER_DEFINED_MATERIAL_MODELS: In user defined material models, a logical parameter 'reject' can be set to .true. to indicate to the implicit solver that equilibrium iterations should be aborted. The criterion is the choise of the implementor, but it could be if plastic strain increases by more than say 5% in one step or damage increases too much, whatever that might render an inaccurate prediction and bad results. Setting this parameter for explicit won't do anything.
- IHYPER = 3 for user shell materials now supports thickness train update, see *MAT_USER_DEFINED_MATERIAL_MODELS.

- *MAT_SIMPLIFIED_RUBBER/FOAM: AVGOPT < 0 is now supported for the FOAM option, which activates a time averaged strain rate scheme to avoid noisy response.
- MAT_181 is now supported for 2D implicit simulations.
- *MAT_ADD_EROSION:

  - A number of extensions and improvements to the DIEM damage model were made, IDAM < 0.
  - General efficiency, it was slow, now it's GOT to be faster.
  - NCS can be used as a plastic strain increment to only evaluate criteria in quantifications of plastic strain.
  - NUMFIP < 0 is employing the GISSMO approach, number of layers for erosion.
  - A new ductile damage criterion based on principal stress added (DMITYP = 4).
  - MSFLD and FLD can be evaluated in mid or outer layers to separate membrane and bending instability (P2).
  - MSFLD and FLD can use an incremental or direct update of instability parameter (P3).
  - Output of integration point failure information made optional (Q2).
  - Specifying DCTYP = -1 on the damage evolution card will not couple damage to stress but the damage variable is only calculated and stored.

- *MAT_SMOOTH_VISCOELASTIC_VISCOPLASTIC, *MAT_275: An elastic-plastic model with smooth transition between elastic and plastic mode is available. It incorporates viscoelasticity and viscoplasticity and is based on hyper-elastoplasticity so it is valid for arbitrarily large deformations and rotations. A sophisticated parameter estimation is required to match test data, it is available for implicit and explicit analyisis but perhaps mostly suited for implicit.
- *MAT_FABRIC_MAP: Stress map material 34 is equipped with bending properties identical to that of the form 14 and form -14 version of the fabric. Coating properties are set in terms of stiffness, thickness and yield. The material is supported in implicit, including optional accounting for the nonsymmetric tangent. Should be used with bending stiffness on, and convergence is improved dramatically if geometric stiffness is turned on.
- *MAT_084 with predefined units (CONM < 0) is now transformed correctly with INCLUDE_TRANSFORM.
- If LCIDTE = 0 in *MAT_121, then LS-DYNA was crashing on some platforms, including Windows. This is fixed.
- Fix initialization issues so that PML models can be run with *CASE commands.
- *MAT_027 is revised to avoid accuracy issues for single precision executables.

  - The nearly imcompressible condition is enhanced for *MAT_027 shell elements.
  - Add a new material model as a option for *MAT_165. *MAT_PLASTIC_NONLINEAR_KINEMATIC_B is a mixed hardening material model, and can be used for fatigue analysis.
  - Output local z-stress in *MAT_037, when *LOAD_STRESS_SURFACE is used. This was previously calculated and saved as another history variable.
  - Add a new material model *MAT_260 (2 forms).

    - Uses non-associated flow rule and Hill's yield surface; including strain rate effect and temperate effect. MIT failure criteria is also implemented.
    - Implemented for solids and shells.
    - Strain rate sensitivity for solids.
    - Option to directly input the Pij and Gij values.
    - Separate the material model *MAT_260 into *MAT_260A and *MAT_260B:
      - MAT260A=*MAT_STOUGHTON_NON_ASSOCIATED_FLOW
      - MAT260B=*MAT_MOHR_NON_ASSOCIATED_FLOW
    - Incorporates FLD into the fracture strain, so as to consider the mesh size effect.
    - Calculates the characteristic length of the element for *MAT_260B, so that an size-dependent failure criterial can be used.
    - When failure happens for half of the integration points through the thickness, the element is deleted.

  - Add Formablitiy Index to *MAT_036, *MAT_037, *MAT_226.
  - Add new history variables for Formability Index, affecting *MAT_036, *MAT_037, *MAT_125, *MAT_226. Those new history variables are FI, beta, effective strain. These comes after the 4 regular history variables.
  - *MAT_036, *MAT_125: New option_NLP is added to evaluate formability under non-linear strain paths. User inputs a forming limit diagram (FLD), and Formablitiy Index (F.I.) will be automatically converted to effective stain vs. beta based space.

- **MPP**

  - Fix problem of MPP pre-decomposition that can occur if the local directory specified in the pfile has very different lengths in the initial run vs the actual run The difference resulted in a line count difference in the size of the structured files created, throwing off the reading of the file in the actual run.
  - Straighten out some silist/sidist issues in MPP decomp:

    - silist and sidist outside of a "region" in the pfile are no longer supported, and an error message is issued which suggests the use of "region { silist" instead.
    - They have been undocumented for several years (since "region" was introduced), and had other issues.

- Fix the keywords, CONTROL_MPP_DECOMPOSITION_CONTACT_DISTRIBUTE and CONTROL_MPP_DECOMPOSITION_CONTACT_ISOLATE, which were not treating each contact interface individually (as the manual states), but collectively.
- Fix for MPP decomp of part sets.
- Fixed *CONTROL_MPP_PFILE (when used inside an include file) so that it honors ID offsets from *INCLUDE_TRANSFORM for parts, part sets, and contact ids referenced in "decomp { region {" specifications. Furthermore, such a region can contain a "local" designation, in which case the decomposition of that region will be done in the coordinate system local to the include file, not the global system. For example:
  *CONTROL_MPP_PFILE
  decomp { region {partset 12 local c2r 30 0 -30 0 1 0 1 0 0}}
  would apply the c2r transformation in the coordinate system of the include file, which wasn't previously possible. The local option can be useful even if there are no such transformations, as the "cubes" the decomposition uses will be oriented in the coordinate system of the include file, not the global system. Furthermore, the following decomposition related keywords now have a_LOCAL option, which has the same effect:

    - *CONTROL_MPP_DECOMPOSITION_PARTS_DISTRIBUTE_LOCAL
    - *CONTROL_MPP_DECOMPOSITION_PARTSET_DISTRIBUTE_LOCAL
    - *CONTROL_MPP_DECOMPOSITION_ARRANGE_PARTS_LOCAL
    - *CONTROL_MPP_DECOMPOSITION_CONTACT_DISTRIBUTE_LOCAL

- Revert revision 86884, which was:

    - "MPP: change to the decomposition behavior of *CONTROL_MPP_DECOMPOSITION_PARTS_DISTRIBUTE
      *CONTROL_MPP_DECOMPOSITION_PARTSET_DISTRIBUTE
      *CONTROL_MPP_DECOMPOSITION_ARRANGE_PARTS
      in the case where a decomposition transformation is also used. Previously, any such regions were distributed without the transformation being applied. This has been fixed so that any given transformation applies to these regions also. So now the transformations will NOT

apply to these keywords. Really, the "region" syntax should be used together with *CONTROL_MPP_PFILE as it is more specific.

- Modify behavior of DECOMPOSITION_AUTOMATIC so that if the initial velocity used is subject to *INCLUDE_TRANSFORM, the transformed velocities are used.
- Fix MPP decomposition issue with "decomp { automatic }" which was not honored when in the pfile.
- Save hex weld creation orientation to the pre-decomposition file so that the subsequent run generates the welds in the same way.
- Fix for MPP not handling element deletion properly in some cases at decomposition boundaries.
- Add new pfile option "contact { keep_acnodes }" which does NOT exclude slave nodes of adaptive constraints from contact, which is the default behavior. (MPP only.)
- MPP Performance-Related Improvements:
    - Allow user input of *LOAD_SEGMENT_FILE through familied files.
    - Bug fix for *LOAD_SEGMENT_FILE to get correct time history data for pressure interpolation.
    - Output two csv files for user to check MPP performance:
        - load_profile.csv: general load balance
        - cont_profile.csv: contact load balance
    - Allow user to control decomp/distribution of multiple airbags using *CONTROL_MPP_DECOMPOSITION_ARRANGE_PARTS
    - memory2 = option on *KEYWORD line
    - Disable unreferenced curves after decomposition using *CONTROL_MPP_DECOMPOSITION_DISABLE_UNREF_CURVES. This applies to the curves used in the following options to speed up the execution several times.
        - *BOUNDARY_PRESCRIBED_MOTION_NODE
        - *LOAD_NODE
        - *LOAD_SHELL_ELEMENT
        - *LOAD_THERMAL_VARIABLE_NODE
- Bug fix for *CONTROL_MPP_DECOMPOSITION_SHOW with *AIRBAG_PARTICLE.
- Fix cpu dependent results when using function RCFORC() in *DEFINE_CURVE_FUNCTION. This affects MPP only.
- Fix hang up when using *DEFINE_CURVE_FUNCTION with element function BEAM(id,jflag,comp,rm) and running MPP with np > 1.
- *CONTROL_MPP_DECOMPOSITION: The cpu cost for solid elements -1 and -2 are accounted for in the mpp domain decomposition.
- Fix bug in *CONTROL_MPP_IO (Windows platform only) related to insufficient administrative privileges for writing tmp file on root drive.

  - Revise l2a utility on Windows platform to create identical node output format as Linux.

- **Output**

  - Fix for MPP external work when bndout is output and there are *BOUNDARY_PRESCRIBED_MOTION_RIGID commands in the input.
  - Fixed the output of forces and associated energy due to *LOAD_RIGID_BODY for both explicit and implicit (*DATABASE_BNDOUT).
  - Fixed stress and strain output of thick shells when the composite material flag is set on *DATABASE_EXTENT_BINARY. The transformation was backwards.
  - If the size of a single plot state was larger than the d3plot size defined by x=<factor> on the execution line, the d3plot database may not be readable by LS-PrePost. This issue is now fixed.
  - *DATABASE_PROFILE: Output data profiles for beams (TYPE = 5) and add density as DATA = 20.
  - New option HYDRO = 4 on *DATABASE_EXTENT_BINARY. Outputs 7 additional variables: the same 5 as HYDRO = 2 plus volumetric strain (defined as Relative Volume - 1.0) and hourglass energy per unit initial volume.
  - Fix for binout output of swforc file which can get the data vs. ids out of sync when some solid spotwelds fail.
  - Fix for d3plot output of very large data sets in single precision.
  - Fix for output of bndout data for joints in MPP, which was writing out incorrect data in some cases.
  - Added new option *INTERFACE_SPRINGBACK_EXCLUDE to exclude selected portions from the generated dynain file.
  - Add a new option to *INTERFACE_COMPONENT_FILE to output only 3 degrees of freedom to the file, even if the current model has 6.
  - Minor change to how pressure is computed for triangles in the INTFOR output.
  - Fix MPP output issue with intfor file.
  - Fixes for writing and reading of dynain data in LSDA format.
  - Corrected the summation of rigid body moments for output to bndout for some special cases in MPP.
  - Corrected the output to d3iter when 10 node tets are present (D3ITCTL on *CONTROL_IMPLICIT_SOLUTION).
  - Enhanced implicit collection of moments for the rcforc file.
  - For implicit, convert spc constraint resultant forces to local coordinate system for output. Also corrected Implicit's gathering of resultant forces due to certain SPC constraints.
  - Fixed the gathering of resultant forces in implicit for prescribed motion on nodes of a constrained rigid body for output to bndout.

- Added output of modal dynamics modal variables to a new file moddynout. Output is controlled by *CONTROL_IMPLICIT_MODAL_DYNAMICS.
- Corrected the output of resultant forces for Implicit Linear analysis. Corrected the output of resultant forces for MPP executions. These enhancements affect a number of ASCII files including bndout.
- The following 4 enhancements are to the eigensolvers, including that used for *CONTROL_IMPLICIT_EIGENVALUE.

  - Standardized and enhance the warning/error messages for Implicit eigensolution for the case where zero eigenmodes are computed and returned in eigout and d3eigv.
  - Added nonsymmetric terms to the stiffness matrix for the implicit rotational dynamics eigenanalysis. This allows brake squeal analysis with the contact nonsymmetric terms from mortar contact now included in the analysis.
  - Updated implicit eigensolution for problems with unsymmetric stiffness matrices. Fixed Rotational Dynamics eigensolution to work correctly when first order matrix (W) is null. (See *CONTROL_IMPLICIT_ROTATIONAL_DYNAMICS).
  - Added the eigensolution for problems with stiffess (symmetric or unsymmetric), mass, and damping.

- Improve Implicit's treatment of constrained joints to account for rounding errors. Applicable to *CONSTRAINED_JOINT with *CONTROL_IMPLICIT_GENERAL.
- For implicit springback, zero out the forces being reported to rcforc for those contact interfaces disabled at the time of springback. Also enhance the removal of contact interfaces for springback computations. For *INTERFACE_SPRINGBACK.
- *DATABASE_RECOVER_NODE is available to recover nodal stress.
- Fix a bug for detailed stress output, eloutdet, for SOLID type 18.
- Support new format of interface force files for ALE, DEM, and CPM. LS-PrePost can display the correct label for each output component.
- Added *DATABASE_NCFORC_FILTER option to allow the NCFORC data to be filtered using either single pass or double pass Butterworth filtering to smooth the output. Added the same filtering capability to *DATABASE_BINARY_D3PLOT. This capability is specified on the additional card for the D3PLOT option and does not require "_FILTER" in the keyword input.
- Fix incorrect mass properties for solids in SSSTAT file when using *DATABASE_SSSTAT_MASS_PROPERTIES.
- Fix seg fault during writing of dynain file if INSTRN = 1 in *INTERFACE_SPRINGBACK and STRFLG.ne.0 in *DATABASE_EXTENT_BINARY and

the *DATABASE_EXTENT_BINARY comes after *INTERFACE_SPRINGBACK. Also output warning message, KEY+1104.

- Fix zero strain values output to curvout for *DEFINE_CURVE_FUNCTION using function, ELHIST, for solid elements.
- Fix missing parts in d3part when MSSCL = 1 or 2 in *DATABASE_EXTENT_BINARY.
- Fix incorrect damping energy computation for glstat.
- Fix incorrect part mass in d3plot for shells, beams & thick shells.
- Fix incorrect curvout values when using BEAM(id,jflag,comp,rm) for *DEFINE_CURVE_FUNCTION and if the beam formulation is type 3, i.e. truss.
- Fix incorrect output to curvout file if using ELHIST in *DEFINE_CURVE_FUNCTION for shells.
- Output stresses for all 4 intg points to eloutdet for cohesive element types 19 & 20.
- Fix incorrect rotational displacement to nodout when REF = 2 in *DATABASE_HISTORY_NODE_LOCAL. Affects MPP only.
- Fix incorrect strains output to elout for shell type 5 and when NIP > 1.
- Fix incorrect acceleration output to nodout file when IACCOP = 1 in *CONTROL_OUTPUT and IGRAV = 1 in *ELEMENT_SEATBELT_ACCELEROMETER.
- Fix corrupted d3plot when RESPLT = 1 in *DATABASE_EXTENT_BINARY and idrflg.ge.5 in *CONTROL_DYNAMIC_RELAXATION.
- Fix missing element connectivities in nastin file when using *INTERFACE_SPRINGBACK_NASTRAN_NOTHICKNESS.
- Fix seg fault when using *DATABASE_BINARY_D3PART with *CONTACT_TIED_SHELL_EDGE_TO_SURFACE. This affects SMP only.
- Fix incorrect output to bndout when using multiple *LOAD_NODE_POINT for the same node and running MPP with ncpu > 1.
- Fix incorrect dyna.inc file when using *MAT_FU_CHANG_FOAM/MAT_83, *DEFINE_COORDINATE_NODES, and *CONSTRAINED_JOINT_STIFFNESS_GENERALIZED with *INCLUDE_TRANSFORM.
- Fix IEVERP in *DATABASE_EXTENT_D3PART which was not honored in writing out d3part files.
- Fix incorrect stresses written out to dynain for thick shells with formulations 1,2 and 4.
- Fix incorrect output to disbout data for discrete beams.
- Fix incorrect output to binary format of disbout. Affects SMP only.
- Fix error when writing initial stresses for thick shells to dynain. Affects MPP only.
- Fix thick shells strain output to dynain.
- Fix incorrect writing of material data to dyna.str for *MAT_SEATBELT when using long = s.
- Fix coordinate/disp output to d3plot of *CONSTRAINED_NODAL_RIGID_BODY's pnode.

- Fixed the initial d3plot state in SMP runs when tied contact is used with theCNTCO parameter on *CONTROL_SHELL. The geometry was wrong in that state.
- Add cross section forces output (*DATABASE_SECFORC) for cohesive elements ELFORM type 19, 20, 21, and 22.
- Slight increase of precision for values in nodout file.
- Add new option FSPLIT to *INTERFACE_SPRINGBACK_LSDYNA to split the dynain file into two files (geometry and initial values).
- *DEFINE_MATERIAL_HISTORIES: New keyword for organizing material history outputs, currently only for solids, shells and beams and the d3plot output but to be extended to tshells and ascii/binout. The purpose is to customize the history variables that otherwise are output via NEIPS/NEIPH/NEIPB on *DATABASE_EXTENT_BINARY, to avoid variable conflict and large d3plots and thus facilitate post-processing of these variables. Currently available in small scale but to be continuously extended.
- Fixed bug affecting IBINARY = 1 (32 bit ieee format) in *DATABASE_FORMAT. This option was not working.
- Fixed incorrect printout of node ID for *ELEMENT_INERTIA.
- Increased the header length to 80 for the following files in binout: matsum, nodout, spcforc, ncforc
- Fixed bug in which d3msg was not written for SMP.
- The d3plot output for rigid surface contact was incorrect for MPP.
- Fixed bugs when when using curve LCDT to control d3plot output.
- Fixed abnormal increase in d3plot size caused by outputting velocity and acceleration when data compression is on.
- Added new variable GEOM in *CONTROL_OUTPUT for chosing geometry or displacement in d3plot, d3part, and d3drlf.
- Added command line option "msg=" to output warning/error descriptions. See MSGFLG in *CONTROL_OUTPUT for alternate method of requesting such output. Accepted values for "msg=" are message# or all.

  - message#, e.g., KEY+101 or 10101. This option will print the error/warning message to the screen.
  - all. this option will print all error/warning messages to d3msg file.

- Fixed bug for *DATABASE_BINARY_D3PROP file if adaptivity used. The error caused blank d3prop output.
- *DATABASE_HISTORY_SHELL_SET combined with *CONTROL_ADAPTIVITY caused error 20211. The error involves the BOX option being used for shell history output.
- Added *INTEGRATION... data to d3prop.

- **Restarts**

  - Fix bug when deleted uniform pressure (UP) airbag during simple restart.
  - Fix for index error that could cause problems for accelerometers during full deck restart in MPP.
  - Fix for MPP output of LSDA interface linking file when restarting from a dump file.
  - Fix incorrect strains in d3plot after restart when STRLG > 1.
  - Fix incorrect velocity initialization for SMP full deck restart when using
  - *INITIAL_VELOCITY_GENERATION and *INITIAL_VELOCITY_GENERATION_START_TIME.
  - Fix incorrect behavior of *CONTACT_ENTITY in full deck restart.
  - Fix incorrect full deck restart analysis if initial run was implicit and the full deck restart run is explicit.
  - Fix ineffective boundary condition for *MAT_RIGID when using *CHANGE_RIGID_BODY_CONSTRAINT with *RIGID_DEFORMABLE_R2D for small deck restart.
  - Fix initialization of velocities of *MAT_RIGID_DISCRETE nodes after restart using *CHANGE_VELOCITY_GENERATION.
  - Fix internal energy oscillation after full deck restart when using *CONTACT_TIED_SURFACE_TO_SURFACE_OFFSET with TIEDID = 1 in optional card D. This affects SMP only.
  - Corrected bug affecting full restart that included any change to node/element IDs. This bug has existed since version R6.
  - Fixed bug affecting d3plot times following fulldeck restart with curve in SMP.
  - Fixed bug in simple restart: *INTERFACE_COMPONENT_FILE forgets the filename and writes to infmak instead.

- ***SENSOR**

  - Enable full restart for *SENSOR.
  - Add optional filter ID to SENSORD of *DEFINE_CURVE_FUNCTION.
  - Enable LOCAL option of *CONSTRAINED_JOINT to be used with *SENSOR_DEFINE_FORCE.
  - Fix a MPP bug that happens when *SENSOR_DEFINE_NODE has a defined N2.
  - *SENSOR_CONTROL:

    - Fix a bug for TYPE = JOINTSTIF
    - Fix a MPP bug for TYPE = PRESC-MOT when the node subject to prescribed motion is part of a rigid body
    - Add TYPE = BELTSLIP to control the lockup of *ELEMENT_SEATBELT_SLIPRING.
    - Add TYPE = DISC-ELES to delete a set of discrete elements.

  - Add FTYPE = CONTACT2D to to *SENSOR_DEFINE_FORCE to track the force from *CONTACT_2D.
  - Add the variable SETOPT for *SENSOR_DEFINE_NODE_SET and *SENSOR_DEFINE_ELEMENT_SET to sense and process data from a node set or element set, resp., resulting in a single reported value.
  - *SENSOR can be used to control *CONTACT_GUIDED_CABLE.
  - Fix a bug related to *SENSOR_DEFINE_FUNCTION triggered by more than 10 sensor definitions.

- **SPG (Smooth Particle Galerkin)**

  - *SECTION_SOLID_SPG (KERNEL = 1): The dilation parameters (DX,DY,DZ) of SPG Eulerian kernel are automatically adjusted according to the local material deformation to prevent tensile instability.

- **SPH (Smooth Particle Hydrodynamics)**

  - Retain user IDs of SPH particles in order to ensure consistent results when changing the order of include files.
  - Add feature to inject SPH particles, *DEFINE_SPH_INJECTION.
  - Added support of various material models for 2D and 3D SPH particles:

    - *MAT_098 (*MAT_SIMPLIFIED_JOHNSON_COOK)
    - *MAT_181 (*MAT_SIMPLIFIED_RUBBER)
    - *MAT_275 (*MAT_SMOOTH_VISCOELASTIC_VISCOPLASTIC)

  - Added support of *DEFINE_ADAPTIVE_SOLID_TO_SPH for 2D shell elements and 2D axisymmetric shell elements.
  - When using *DEFINE_ADAPTIVE_SOLID_TO_SPH, eliminated duplicate kinetic energy calculation for SPH hybrid elements (both SPH particles and solid elements contributed kinetic energy into global kinetic energy).
  - Added support of second order stress update (OSU = 1 in *CONTROL_ ACCURACY keyword) for 2D and 3D SPH particles. This is necessary for simulation of spinning parts.
  - Added ISYMP option in *CONTROL_SPH to define as a percentage of original SPH particles the amount of memory allocated for generation of SPH ghost nodes used in *BOUNDARY_SPH_SYMMETRY_PLANE.
  - Fixed unsupported part and part set option in *BOUNDARY_SPH_FLOW.
  - Fixed unsupported ICONT option from *CONTROL_SPH when combined with *BOUNDARY_SPH_FLOW.
  - *DEFINE_SPH_TO_SPH_COUPLING: Output contact forces between two SPH parts (x,y,z and resultant forces) into sphout. The forces can be plotted by LS-PrePost.
  - *CONTACT_2D_NODE_TO_SOLID: Added bucket sort searching algorithm to speed up the process of finding contact pairs between SPH particles and solid segments.

- **Thermal**

  - Corrected a long standing bug in MPP thermal associated with spotwelds (*CONSTRAINED_SPOTWELD) using thermal linear solver option 11 or greater. The spotweld loads were not being loaded correctly due to an indexing issue in MPP.
  - Fix for thermal with *CASE.
  - Fix MPP support for thermal friction in SOFT = 4 contact.
  - Fixed bug where thermal solver gives a non-zero residual even though no loads are present.
  - Added SOLVER = 17 (GMRES solver) to *CONTROL_THERMAL_SOLVER for the conjugate heat transfer problem. The GMRES solver has been developed as an alternative to the direct solvers in cases where the structural thermal problem is coupled with the fluid thermal problem in a monolithic approach using the ICFD solver. A significant savings of calculation time can be observed when the problem reaches 1M elements. This solver is implemented for both SMP and MPP.
  - *CONTACT_(option)_THERMAL (3D contact only): Add variable FRTOHT to specify fraction of frictional energy applied to slave surface. It follows that 1.-FRTOHT is applied to master surface. Default is 0.5 which gives a 50% - 50% split between the slave and master surfaces which was hardwired in prior releases.
  - First release of AUTOMATIC_SURFACE_TO_SURFACE_TIED_WELD_THERMAL. This will only work when used with BOUNDARY_THERMAL_WELD. This combination of keywords will activate a condition where sliding contact will become tied contact on cooldown when the temperature of the segments in contact go above an input specified temperature limit during welding.
  - *LOAD_THERMAL_D3PLOT: The d3plot data base was changed such that the 1st family member contains control words, geometry, and other control entities. Time state data begins in the 2nd family member. This change allows the new d3plot data structure to be read in by LS-DYNA when using the *LOAD_THERMAL_D3PLOT keyword. This change is not backward compatible. The old d3plot data structure will no longer be read correctly by LS-DYNA.
  - Synchronize data in TPRINT for SMP and MPP:

    - Fixed output to tprint/binout for thermal contact.
    - Fixed part IDs for part energies.
    - Fixed format of TPRINT file generated by l2a.

  - Fixed handling of start time defined with *CONTROL_START for thermal solver.
  - Change the maximum number of *LOAD_HEAT_CONTROLLER definitions from 10 to 20.

  - Added a third parameter to the TIED_WELD contact option. The parameter specifies heat transfer coefficient h_contweld for the welded contact. Before welding, the parameter from the standard card of the thermal contact is used.
  - Parameter FRCENG supported for mortar contact to yield heat in coupled thermomechanical problems.

- **XFEM (eXtended Finite Element Method)**

  - Added ductile failure to XFEM using critical effective plastic strain as failure criterion.

- **Miscellaneous**

  - Support *SET_NODE_GENERAL PART with SPH and DES.
  - *DEFINE_POROUS_...: Compute the coefficients A and B with a user defined routine in dyn21.F.
  - Fixed bugs in Staged Construction (*DEFINE_STAGED_CONSTRUCTION_PART):

    - Staged construction not working on SMP parallel. Symptoms could include wrong elements being deleted.
    - Staged construction with beam elements of ELFORM = 2: when these beams are dormant, they could still control the time step.
    - Staged construction with *PART_COMPOSITE. The bug occurred when different material types were used for different layers within the same part, and that part becomes active during the analysis. The symptom of the bug was that stresses and/or history variables were not set to zero when the part becomes active.

  - Bugs fixed in *DAMPING_FREQUENCY_RANGE_DEFORM:

    - Incorrect results when large rigid body rotations occur.
    - If RYLEN on *CONTROL_ENERGY = 2, the energy associated with this damping should be included in the Internal Energy for the relevant part(s). This energy was being calculated only if there was also *DAMPING_PART_STIFFNESS in the model. Now fixed - the damping energy will be included in the internal energy whenever RYLEN = 2.

  - Fixed NID option of *DEFINE_COORDINATE_VECTOR (bug occurred in MPP only).
  - Fix lsda open mode to require only minimal permissions to avoid unnecessary errors, for example if using an interface linking file that is read only.
  - Fix for DPART processing (*SET_..._GENERAL) for solid and thick shell elements.

- Fix for JOBID > 63 characters.
- Fix input processing problem (hang) that could happen in some unusual cases if encrypted *INCLUDE files are used.
- Fix interaction of *CASE with jobid = on command line, so the jobid on the command line is combined with the generated case ids instead of being ignored.
- *INCLUDE_NASTRAN:
  - Integration defaults to Lobatto for Nastran translator.
  - The default number of integration points is set to 5 for Nastran translator.
- Issue error message and terminate when illegal *DEFINE_ TRANSFORMATION is specified.
- Add OPTION = POS6N to *DEFINE_TRANSFORMATION to define transformation with 3 reference nodes and 3 target nodes.
- Add OPTION = MIRROR to *DEFINE_TRANSFORMATION.
- Fix a bug that could occur when adapted elements are defined in a file included by *INCLUDE_TRANSFORM.
- Fix a bug that could occur when *BOUNDARY_SPC_SYMMETRIC_ PLANE is used together with *INCLUDE_TRANSFORM.
- Fix a bug that occurs when *DEFINE_BOX is included by *INCLUDE_ TRANSFORM.
- Make *SET_NODE_COLLECT work together with *NODE_SET_MERGE.
- Fix incorrect shell set generated when using *SET_SHELL_GENERAL with OPTION = PART.
- Add error trap for *SET_PART_LIST_GENERATE_COLLECT to catch missing part IDs.
- Fixed bug in *INCLUDE_TRANSFORM for adaptive case if JOBID is used.
- Fixed bug in memory allocation for *DEFINE_CURVE if total number of points in curve is more than 100.
- Fixed bug with *INCLUDE_TRANSFORM and *CONTROL_ADAPTIVITY due to an *INCLUDE inside *INCLUDE_TRANSFORM file. Added new files: adapt.inc# for *INCLUDE_TRANSFORM file. The *NODE, *ELEMENT_SHELL and *ELEMENT_SOLID are removed from include file.
- Fixed bug for DPART option in *SET_SEGMENT_GENERAL. DPART option was treated as PART option before.
- Fixed failure of *PARAMETER definition in long format.
- Fixed error in reading solid id for *SET_SOLID_GENERAL.
- Ignore any nonexistant part set IDs in *SET_PART_ADD.
- Fix bug in which sense switches sw2 and sw4 don't work when the output interval for glstat is small.
- Fixed bug if *DEFINE_CURVE is used to define adaptivity level.

- Three new keywords are implemented in support of user defined subroutines: *MODULE_PATH[_RELATIVE], MODULE_LOAD, MODULE_USE.

    - The MODULE feature allows users to compile user subroutines into dynamic libraries without linking to the LS-DYNA main executable.
    - The dynamic libraries are independent from the main executable and do not need to be recompiled or linked if the main executable is updated.
    - This feature loads multiple dynamic libraries on demand as specified in the keywords.
    - Without the MODULE feature, only one version of each umat (such as umat41) can be implemented. With the MODULE feature, most umat subroutines can be have multiple versions in multiple dynamic libraries, and used simultaneously.
    - The MODULE feature supports all user subroutines.
    - The LS-DYNA main executable may also run without any dynamic libraries if no user subroutines are required.

Capabilities added to create LS-DYNA R10.0:

See release notes (published separately) for further details.

- ***AIRBAG**

    - Enhance the robustness of *AIRBAG_INTERACTION to help avoid instability in MPP when the interaction involves more than two bags.
    - *AIRBAG_PARTICLE:

        - Adjust dm_out calculation of vent hole to avoid truncation error.
        - Fix bug in chamber output when there are multiple airbags and multiple chambers not in sequential order.
        - Bug fix for closed volume of airbag/chamber with intersecting tubes.
        - Add new feature to allow user to define local coordinates of jetting of particles through internal vents.
        - Support *SENSOR_CONTROL for CPM airbag.
        - CPM is not supported for dynamic relaxation. Disable CPM airbag feature during DR and reactivate airbag following DR.
        - Allow solid parts in definition of internal part set. The solid volume will be excluded from the airbag volume.
        - Allow additional internal part set for shells. The shell part should form a closed volume and its volume will be excluded from the airbag volume.

- ***ALE**

- *LOAD_BLAST_SEGMENT: Automatically generate the ALE ambient elements attached to a segment or segment set.
- *BOUNDARY_AMBIENT_EOS: implement *DEFINE_CURVE_FUNCTION for the internal energy and relative volume curves.
- *CONTROL_ALE, *CONSTRAINED_LAGRANGE_IN_SOLID and *ALE_REFERENCE_SYSTEM: If NBKT < 0 in *CONTROL_ALE, call *DEFINE_CURVE to load a curve defining the number of cycles between bucket sorting in function of time. If NBKT > 0, the bucket sorting is activated if the mesh rotations and deformations are large.
- *ALE_FSI_TO_LOAD_NODE: Implement a mapping of the FSI accelerations (penalty forces/masses) computed by *CONSTRAINED_LAGRANGE_IN_SOLID (ctype = 4) between different meshes.
- DATABASE_FSI, *DATABASE_BINARY_FSIFOR and *DATABASE_BINARY_FSILNK: Add a parameter CID to output fsi forces in a local coordinate system.
- Structured ALE (S-ALE) solver:
  - ALE models using rectilinear mesh can be directly converted to S-ALE models and run using S-ALE solver by assigning CPIDX = -1 in *ALE_STRUCTURED_MESH.
  - S-ALE progressive mesh generation via RATIO in *ALE_STRUCTURED_MESH_CONTROL_POINTS.
- Recode ALE Donor Cell/Van Leer advection routines and restructure *CONSTRAINED_LAGRANGE_IN_SOLID communication algorithm. These give 30% improvement in run time.

- ***BOUNDARY**
  - *BOUNDARY_PWP can now accept a *DEFINE_FUNCTION instead of a load curve. The input arguments are the same as for *LOAD_SEGMENT: (time, x, y, z, x0, y0, z0).
  - Add option of "toffset" for *BOUNDARY_PRESCRIBED_ORIENTATION_RIGID to offset the curves by the birth time.
  - MPP now supports MCOL coupling, *BOUNDARY_MCOL.
  - Fix bug of there being fully constrained motion of a rigid part when prescribing more than one translational dof with *BOUNDARY_PRESCRIBED_MOTION_RIGID while con2 = 7 in *MAT_RIGID, i.e., all rotational dof are constrained.
  - Instead of error terminating with warning message, STR+1371, when *BOUNDARY_PRESCRIBED_MOTION and *BOUNDARY_SPC is applied

to same node and dof, issue warning message, KEY+1106, and release the conflicting SPC.

- Fix erroneous results if SET_BOX option is used for *BOUNDARY_PRESCRIBED_MOTION.
- Fix *BOUNDARY_PRESCRIBED_ACCELEROMETER_RIGID for MPP. It may error terminate or give wrong results if more than one of this keyword is used.
- Fix segmentation fault when using *BOUNDARY_PRESCRIBED_ORIENTATION with vad = 2, i.e. cubic spline interpolation.
- Added instruction *BOUNDARY_ACOUSTIC_IMPEDANCE for explicit calculations that applies an impedance boundary condition to the boundary of *MAT_ACOUSTIC element faces. This is a generalization of the non-reflecting boundary condition. Both *LOAD and *BOUNDARY_ACOUSTIC_IMPEDANCE may be used on the same faces, in which case the boundary acts like both and entrant and exit boundary.
- Fixed a problem with non-reflecting boundaries redefining the bulk modulus which caused contact to change behavior.
- Added support for acoustic materials ith non-reflective boundaries.
- Fix the single precision version so that *INCLUDE_UNITCELL now has no problem to identify pairs of nodes in periodic boundaries.
- When using *INCLUDE_UNITCELL to generate Periodic Boundary Constraints (PBC) for an existing mesh, a new include file with PBCs is generated instead of changing the original mesh input file. For example, if users include a file named "mesh.k" through *INCLUDE_UNITCELL (INPT = 0), a new include file named "uc_mesh.k" is generated where all PBCs are defined automatically following the original model information in mesh.k.
- *INCLUDE_UNITCELL now supports long input format in defining the element IDs.
- Include SPC boundary conditions as part of H8TOH20 solid element conversion.
- Add a new option SET_LINE to *BOUNDARY_PRESCRIBED_MOTION: This option allows a node set to be generated including existing nodes and new nodes created from h-adaptive mesh refinement along the straight line connecting two specified nodes to be included in prescribed boundary conditions.

- **BLAST**

  - *PARTICLE_BLAST and DES:

    - Consider eroding of shell and solid in particle_blast.
    - Support interface force file output for gas particle-structure coupling.
    - Bug fix for wet DES coupled with beam.
    - Support *SET_NODE_GENERAL PART with SPH or DES.

    - MPP now uses async communication for DES coupling to improve general performance.
    - Support for solid element when modeling irregular shaped charge with HECTYPE = 0/1 in *PARTICLE_BLAST.
    - Output adaptive generated DES and NODE to a keyword file.

  - Fix inadvertent detonation of HE part when there are more than one HE part and even though the HE part is not defined with *INITIAL_DETONATION.
  - Fixed explicit *BOUNDARY_USA_COUPLING to support *INITIAL_STRESS and *INITIAL_STRAIN_ usage, typically from a dynain file.
  - Fixed explicit *BOUNDARY_USA_COUPLING to support *CONTROL_DYNAMIC_RELAXATION IDRFLF = 5, so a static implicit calculation can be used to initialize/preload a model before conducting an explicit transient calculation. If inertia relief is used during the static phase, then it must be disabled with *CONTROL_IMPLICIT_INERTIA_RELIEF for the explilcit phase.
  - Support imperial unit system for *PARTICLE_BLAST. mass = lbf-s^2/in, length = inch, time = second, force = lbf, pressure = psi.
  - Add option to define detonation point using a node for *PARTICLE_BLAST.
  - Add interface force file output for *PARTICLE_BLAST with keyword *DATABASE_BINARY_PBMFOR and command line option "pbm=". This output of forces for gas-particle-structure coupling.
  - For *PARTICLE_BLAST, add built-in smoothing function for particle structure interaction.
  - For *PARTICLE_BLAST, when coupling with DEM, the DEM nodes that are inside HE domain are automatically deactivated.
  - Add support for solid elements when modeling irregular shaped charge with HECTYPE = 0/1 for *PARTICLE_BLAST. The original approach only supports shell elements and the initial coordinates of HE particle are at shell surface. The model had to relax several hundred time step to let particle fill in the interior space, which was not convenient. Using new approach, the initial positions of HE particles are randomly distributed inside the container by using solid element geometry. Both hex and Tet solids are supported.
  - For particle blast method (PBM), consider reflecting plane as infinite.
  - Change the name of keyword *DEFINE_PBLAST_GEOMETRY to *DEFINE_PBLAST_HEGEO.

- ***CESE (Compressible Fluid Solver)**

  - CESE time steps:

  - Modified the blast wave boundary condition treatment to make it more stable in blast wave calculations (with *LOAD_BLAST_ENHANCED).
  - The flow field calculation will be skipped if the structural time-step is much smaller than the fluid time step, until both time-steps reach the same order. This will save CPU time in some fluid/structure interaction (FSI) problem calculations.
  - In addition to depending upon the local CFL number, the fluid time step 'dt' calculation has been modified to also adjust dynamically to extreme flow conditions. This makes stiff flow problems more stable especially in 3D fluid problem calculations when the mesh quality is poor.

- Moving mesh solvers:

  - Corrected several aspects of the implicit ball-vertex (BV) mesh motion solver for the following keywords:
    *ICFD_CONTROL_MESH_MOV
    *CESE_CONTROL_MESH_MOV.
  - The absolute tolerance argument is no longer used by the BV solver. As an example, the following is all that is needed for CESE moving mesh problems:

    ```
    *CESE_CONTROL_MESH_MOV
     $    ialg  numiter   reltol
             1       500  1.0e-4
    ```

  - Also corrected the CESE moving mesh solvers for a special case involving a wedge element. Also, fixed the d3plot output of wedge element connectivities for the CESE moving mesh solvers.

- CESE d3plot output:

  - Added real 2D CESE output, and this is confirmed to work with LSPP4.3 and later versions. This also works for d3plot output with the 2D CESE axisymmetric solver.
  - For all immersed-boundary CESE solvers, corrected the plotting of the Schlieren number and the chemical species mass fractions.
  - The following new CESE input cards are related to surface d3plot output:
    *CESE_SURFACE_MECHSSID_D3PLOT
    *CESE_SURFACE_MECHVARS_D3PLOT
    In conjunction with the above, new FSI and conjugate heat transfer output on solid (volume) mesh outside boundaries is now supported.

- CESE immersed-boundary method (IBM) FSI solvers:

    - *CESE_FSI_EXCLUDE is a new keyword for use with the CESE immersed boundary method FSI solvers. With it, unnecessary structural parts that are not actively participating in the FSI in the CESE IBM-FSI solver can now be excluded from the CESE FSI calculation. This is also supported for the case when some of the mechanics parts involve element erosion.

  - CESE chemistry solvers:

    - In R10, we also updated several things in the FSI solver with chemistry called FSIC. In chemical reacting flow, a delta time between iterations is extremely important for code stabilization and thus, to get reasonable results. To this end, we optimized such an iterative delta time, which is based on the CFL number. This optimization is based on the gradient of the local pressure, which we think will dominate control of the CFL number.
    - Next, the total number of species are increased up to 60 species in chemical reacting flow, so that the reduced Ethylene (24~53 species) and Methane (20~60 species) combustion are possible with this version.
    - We will update more practical examples about FSIC problems including precise experimental validations.
    - Note that we can provide some related examples upon user request.
    - Other corrections of note include the following:
      Brought in enthalpy-related corrections to the CESE chemistry solvers.
      Fixed the conjugate heat transfer boundary condition for the 2D and 3D CESE fixed mesh chemistry solvers.
      Corrected the initialization of fluid pressure for CESE IBM chemistry solvers.
      Enabled output of the timing information for the CESE chemistry solvers.
      Added restart capability to the CESE chemistry solvers.

- ***CHEMISTRY**

  - New inflator models of Pyrotechnic and Hybrid type are updated. It is important to note that these are basically 0-dimensional models via the following two main keywords,
    *CHEMISTRY CONTROL_INFLATOR
    *CHEMISTRY_INFLATOR_PROPERTIES
  - By using the *CHEMISTRY CONTROL_INFLATOR keyword, the user can select the type of the solver, output mode, running time, delta t, and time interval for output of time history data.
    For example, if we have a keyword set up as,

```
*CHEMISTRY CONTROL_INFLATOR,
 $ isolver  ioutput  runtime    delt   p_time
         1         0        0.1  1.0e-6    5.0e-4
```

with "isolver set to 1", the user can simulate a conventional Pyrotechnic inflator mode, while with "isolver" set to 2 or 3, Hybrid inflator simulation is possible.

- In addition, to continue an airbag simulation via an ALE or CPM method, the user can save the corresponding input data file by using "ioutput" option. For more details about airbag simulations using a saved data file, refer to the keyword manual.
- Also, note that the updated version has two options for the Hybrid models:

  - isolver = 2 => Hybrid model for the cold flow
  - isolver = 3 => Hybrid model for the heated flow.

- In the *CHEMISTRY_INFLATOR_PROPERTIES keyword, there are several cards to set up the required properties of an inflator model. The first two cards are for the propellant properties involved in inflator combustion. For example,

```
 $card1: propellants
 $ comp_id    p_dia  p_height   p_mass   p_tmass
          10     0.003    0.0013      2.0e-5  5.425e-3
 $card2: control parameters
 $ t_flame   pindex      A0    trise   rconst
       2473.        0.4  4.45e-5     0.0     0.037
```

  In the first card, the user can specify the total amount of propellant particles and their shape.
  Using the second card, the user can also specify the thermodynamics of the propellant and its burning rate.
  To support the options in card2, especially the second option, pindex, and the third, A0, we provide a standalone program upon request for the propellant equilibrium simulation.
  The remaining cards are for the combustion chamber, gas chamber, and airbag, respectively.

- ***CONTACT**

  - *CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_MORTAR_TIED_WELD for modeling welding has been added. Surfaces are tied based on meeting temperature and proximity criteria. Non-MORTAR version of this contact was introduced at R9.0.1.
  - Fix issue setting contact thickness for rigid shells in ERODING contact.
  - Add MPP support for *CONTACT_AUTOMATIC_GENERAL with adaptivity.

- Change "Interface Pressure" report in intfor file from abs (force/area) to -force/area, which gives the proper sign in case of a tied interface in tension.
- Rework input processing so that more than one *CONTACT_INTERIOR may be used, and there can be multiple part sets in each one.
- Minor change to how pressure is computed for triangles in the intfor database.
- Fix 2 bugs for contact involving high order shell elements:
  - When high order shell elements are generated by SHL4_TO_SHL8.
  - When using a large part id like 100000001.
- Implement a split-pinball based contact option for neighbor elements in segment-based contact. Invoke this option by setting |SFNBR|>=1000. The new algorithm is more compatible with DEPTH = 45 so that there is no longer a need to split quads.
- The effect of shell reference system offsets on contact surface location is now properly considered when running MPP. The shell offset may be specified using NLOC in *SECTION_SHELL or in *PART_COMPOSITE, or by using the OFFSET option of *ELEMENT_SHELL. This effect on contact is only considered when CNTCO is set to 1 or 2 in *CONTROL_SHELL.
- Fix bug of zero forces in rcforc at time = 0.0 for *CONTACT_AUTOMATIC_SURFACE_TO_SURFACE after dynamic relaxation when consistency is on in SMP.
- Fix input error when using many *RIGIDWALL_GEOMETRIC_... with _DISPLAY option.
- Fix input error when *CONTACT_ENTITY is attached to a beam part, PID.
- Fix error termination due to negative volume, SOL+509, even when *CONTACT_ERODING... is set. This affects MPP only.
- Check whether a slave/master node belongs to a shell before updating the nodal thickness when ISTUPD > 0.0 in *CONTROL_SHELL and SST/MST.ne.0.0 and in SSFT/SMFT = 0.0 card 3 of *CONTACT_..... For SMP only.
- Fix penetrating nodes when using *CONTACT_ERODING_NODES_TO_SURFACE with SOFT = 1 in *MAT_142/*MAT_
- Fix seg fault when using *CONTACT_AUTOMATIC_SINGLE_SURFACE_TIED with consistency mode, .i.e. ncpu < 0, for SMP.
- Fix corrupted intfor when using parts/part sets in *CONTACT_AUTOMATIC_....This affects SMP only.
- Implement incremental update of normal option, invoked by TIEDID = 1, for *CONTACT_TIED_NODES_TO_SURFACE_CONSTRAINED_OFFSET for SMP.
- Fix unconstrained nodes when using *CONTACT_TIED_SURFACE_TO_SURFACE_CONSTRAINED_OFFSET resulting in warning message, SOL+540. This affects SMP only.

- Fix spurious repositioning of nodes when using *CONTACT_SURFACE_TO_SURFACE for SMP.
- Added support to segment based contact for the SRNDE parameter on optional card E. This option allows round edge extensions that do not extend beyond shell edges and also square edges. The latter overlaps with the SHLEDG parameter on card D.
- Fixed a potential memory error that could occur during segment based contact input.
- Fixed an error that could cause an MPP job to hang in phase 3. The error could occur when SOFT = 2 contact is used with the periodic intersection check and process 0 does not participate in the contact.
- Modified SOFT = 2 contact friction when used with *PART_CONTACT to define friction coefficients, and the two parts in contact have different coefficient values. With this change, the mu values used for contact will be the average of the values that are calculated for each part. Prior to this change, mu was calculated for only the part that is judged to be the master. This change makes the behavior more predictable and also makes it behave like the other contacts with SOFT = 0 and SOFT = 1.
- Added a warning message (STR+1392) for when trying to use the ORTHO_FRICTION contact option with SOFT = 2 contact, because that option is not available. The contact type is switched to SOFT = 1.
- Fixed serious error in MPP *CONTACT_2D_AUTOMATIC_SURFACE_TO_SURFACE when used with node sets to define the contact surfaces. The master side was likely to trigger a spurious error about missing nodes that terminated the job.
- Switched segment based (SOFT = 2) non-eroding contact to prevent it from adding any new segments when brick element faces are exposed when other elements are deleted. There were two problems. The first is that the interface force file could not support NFAIL = 1 on *DATABASE_EXTENT_INTFOR because the intfor file does not expect new segments to replace the old, so it just undeletes the old segments instead of adding the new. The second problem is that when non-eroding contact is used, we only have enough memory in fixed length arrays for the segments that exist at t = 0. When segments are deleted, I was using the space that they vacated to create new segments, but it was very likely that some segments could not be created when the number of open spaces was less than the number of new segments that are needed. In this case, some segments would not be created and there would be surfaces that could be penetrated with no resistance. This behavior is impossible to predict, so it seems better to prevent any new segments from being created unless eroding contact is used.
- Fixed rcforc output for MPP 2D automatic contact. The forces across processors were missed.

- Fixed a segment based contact error in checking airbag segments. This affects only airbags that are defined by control volumes, that is defined by *AIRBAG. The symptom was a segmentation fault.
- Fixed SMP eroding segment based (SOFT = 2) contact which was not activating the negative volume checking of brick elements. The MPP contact and the other SMP contacts were doing this but not SMP SOFT = 2.
- Fixed support for CNTCO on *CONTROL_SHELL by segment based (SOFT = 2) contact. It was adjusting the contact surface only half of what it should have done.
- Fixed eroding segment based contact when used with the CNTCO > 0 on *CONTROL_CONTACT. A segmentation fault was occurring.
- Modified MPP segment based (soft = 2) contact to use R8 buffers to pass nodal coordinates. This should reduce MPP scatter when decomposition changes.
- Added support for using a box to limit the contact segments to those initially in the box when using eroding segment based contact. The box option has not been available for any eroding contact up until now. (SOFT = 2 and SBOXID, MBOXID on *CONTACT_ERODING_...).
- Fixed force transducers with MPP segment based contact when segments are involved with multiple, 2-surface force transducers. The symptom was that some forces were missed for contact between segments on different partitions.
- Added support for *ELEMENT_SOURCE_SINK used with segment based contact. With this update, inactive elements are no longer checked for contact.
- Fixed an MPP problem in segment based contact that cased a divide by zero during the bucket sort. During an iteration of the bucket sort, all active segments were somehow in one plane which was far from the origin such that a dimension rounded to zero. The fix for this should affect only this rare case and have no effect on most models.
- Modified segment based (SOFT = 2) contact to make SMP and hybrid faster, particularly for larger numbers of processors.
- Fixed thermal MPP segment based contact. The message passing of thermal energy due to friction was being skipped unless peak force data was written to the intfor file.
- Fixed likely memory errors in MPP problems with 2D automatic contact when friction is used.
- Support the VC parameter (coefficient for viscous friction) in the case of segment based contact, which has previously been unsupported. This option will work best with FNLSCL > 0, DNLSCL = 0 on optional card D. The card D option causes the contact force to be proportional to the overlap area which causes even pressure distribution.
- Enabled segment based contact (SMP and MPP) to work with type 24 (27-node) brick elements.

- Fixed MPP segment based contact for implicit solutions. During a line search, some data was not restored correctly when the solver goes back to the last converged state. This caused possible memory errors.
- Fixed friction for MPP segment based contact in the implicit solver. The sliding velocity was calculated incorrectly using the explicit time step rather than the implicit step.
- Fixed a bug in MPP *CONTACT_2D_AUTOMATIC..., where a flaw in code used during MPP initialization could cause segments to fail to detect penetration.
- Fixed the thick beam checking of *CONTACT_2D_AUTOMATIC_SINGLE_SURFACE in the MPP version. There was a memory error that could occur if thick beams were in the model.
- New values for user friction element history (*USER_INTERFACE_FRICTION): material directions, relative velocity components and yield stress.
- Add new user-defined interface for tiebreak contact invoked by *CONTACT_AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE_TIEBREAK_USER.
- MORTAR CONTACT

    - PENMAX and SLDTHK has taken over the meanings of SST and TKSLS in R9 and earlier, although in a different way. Now PENMAX corresponds to the maximum penetration depth for solid elements (if nonzero, otherwise it is a characteristic length). SLDTHK is used to offset the contact surface from the physical surface of the solid element, instead of playing with SST and TKSLS, which was rather awkward. This update also saves the pain of having to treat shells and solids in separate interfaces if these features are wanted. This changes the behavior in some inputs that did have SST turned on for solids, but a necessary measure to make the contact decent for future versions.
    - The characteristic length for solid elements has been revised to not result in too small sizes that would lead to high contact stiffnesses and less margin for maximum penetration.
    - SFS on CONTACT_..._MORTAR can be input as negative, then contact pressure is the -SFS load curve value vs penetration.
    - Smooth roundoffs of sharp edges in MORTAR contact has been extended to high order segments, meaning that edge contact is valid even in this case.
    - The MORTAR contact now honors the NLOC parameter for shells, see *SECTION_SHELL, adjusting the contact geometry accordingly. Note that CNTCO on *CONTROL_SHELL applies as if always active, meaning that if NLOC is on, then CNTCO will also be "on" for MORTAR contacts.

        - Output of contact gaps to the intfor file is now supported for MORTAR contact, see *DATABASE_EXTENT_INTFOR.
        - Transducer contacts, *CONTACT_..._FORCE_TRANSDUCER, are supported for MORTAR contact in SMP and MPP. A disclaimer is that the slave and master sets in the transducer have to be defined through parts or part sets. Warnings are issued if this is violated.
        - Option 2 is now supported for tiebreak MORTAR contact, *CONTACT_..._MORTAR_TIEBREAK, but only for small sliding. Options 4 and 7 are supported in the MORTAR tiebreak contact for any type of sliding.
        - For explicit analysis, the bucket sort frequency for MORTAR contact is 100, but can be changed through parameter BSORT on the CONTACT_..._MORTAR card or NSBCS on CONTROL_CONTACT. Note that the MPP bucket sort parameter does not apply. This assumes to improve the efficiency of MORTAR explicit contact significantly compared to R9 and earlier versions.
        - Dynamic friction is supported in MORTAR contact for explicit and implicit dynamic analysis. See FD and DC on *CONTACT_... card.
        - Wear calculations are supported for the MORTAR contact. See CONTACT_ADD_WEAR.
        - Triangular shell form 24 is supported with MORTAR forming contact and accounts for high order shape functions.
        - Automatic MORTAR contact now supports contact with end faces of beam elements and not just the lateral surfaces.
        - Mortar contact is available in 2D plane strain and axisymmetric simulations, but only for SMP implicit. See CONTACT_2D_...MORTAR.

    - Wear computed from *CONTACT_ADD_WEAR can optionally be output to dynain on optional card of *INTERFACE_SPRINGBACK_LSDYNA. This will generate *INITIAL_CONTACT_WEAR cards for subsequent wear simulations, and LS-DYNA will apply this wear and modify geometry accordingly. Restrictions as described in the manual apply.
    - Improve SOFT = 6 under *CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE to allow users to define part ID and a node set is automatically generated.

- ***CONSTRAINED**

    - Add frictional energy calculation for constraint-based rigid walls.
    - *CONSTRAINED_BEAM_IN_SOLID:

        - Works with r-adaptivity now.
        - Can now constrain beams in tshells as well as solids.

  - Fix a bug for *CONSTRAINED_LOCAL that might mistakenly constrain z-translation when RC = 0.
  - The following options do not support MEMORY = auto properly. The MEMORY = auto option will be turned off in this section and report an error if additional memory allocation is needed.
    *CONSTRAINED_LINEAR_OPTION
    *CONSTRAINED_MULTIPLE_GLOBAL
  - Switched translational joints with stiffness to use double precision storage for the displacement value so that the calculated forces are more accurate. This prevents round-off error that can become significant.
  - Fixed *CONSTRAINED TIED_NODES_FAILURE when used with MPP single surface segment based contact. Non-physical contact between segments that share tied constraints was being penalized leading to failure of the constraints.
  - The SPR models (*CONSTRAINED_SPR2, *CONSTRAINED_INTERPOLATION_SPOTWELD) now support the SPOTDEL option of *CONTROL_CONTACT. That means if shell elements involved in the SPR domain fail, the SPR gets deactivated.

- ***CONTROL**

  - Fix possible error termination with single precision MPP when PSFAIL.ne.0 in *CONTROL_SOLID and using solid formulation 10/13/44.
  - Fix spurious deletion of elements when using TSMIN.ne.0.0 in *CONTROL_TERMINATION, erode = 1 in *CONTROL_TIMESTEP and initialized implicitly in dynamic relaxation.
  - Added keyword *CONTROL_ACOUSTIC to calculate the nodal motions of *MAT_ACOUSTIC nodes for use in d3plot and time history files. Without this option the *MAT_ACOUSTIC mesh propagates pressure but does not deform because it uses a linear Eulerian solution method. The structural response is unaffected by this calculation; it is only for visualization and will roughly double the time spent computing acoustic element response.
  - When IACC = 1 on *CONTROL_ACCURACY and for shell type 16/-16 in nonlinear implicit, shell thickness change due to membrane strain when ISTUPD > 0 in *CONTROL_SHELL is now included in the solution process and will render continuity in forces between implicit time steps. The output contact forces will reflect the equilibrated state rather than the state prior or after the thickness update.
  - Fix bug when RBSMS in *CONTROL_RIGID, affecting mass scaled solutions, is used in conjunction with *ELEMENT_INERTIA and/or *PART_INERTIA, specifically with choices of IFLAG on *CONSTRAINED_RIGID_BODIES and *CONSTRAINED_EXTRA_NODES.
  - Tshells added to the subcycling scheme (*CONTROL_SUBCYCLE).

- Tshells and spotweld beams are supported in selective mass scaling. See IMSCL in *CONTROL_TIMESTEP.
- Add a new keyword: *CONTROL_FORMING_SHELL_TO_TSHELL to convert shell elements to tshell elements.

    - If a parent node has SPCs, the same SPC constraints will be applied to the corresponding tshell nodes.
    - If adaptivity is invoked, *BOUNDARY_SPC_SET is automatically updated to include newly generated nodes.
    - Allows the normal of the segment set to be changed.
    - Can offset the generated tshells from the mid-surface of the parent shells.
    - Automatically generate segment sets for the top and bottom surfaces, which can be used for contact.

- **DISCRETE ELEMENT METHOD**

    - Implement generalized law of erosion for *DEFINE_DE_TO_SURFACE_COUPLING based on the following article in the journal "Wear": Magnee, A., Generalized law of erosion: application to various alloys and intermetallics, Wear, Vol. 181, 500, 1995.
    - Modify tangential force calculation to get better rigid body rotation behavior for *DEFINE_DE_BOND
    - Support restart feature for DEM interface force file and DATABASE output.
    - Instead of using bulk modulus, use mass and time step to estimate contact stiffness for SPH-DEM coupling. This should be better if DEM material is quite different from SPH material.
    - Fix *DEFINE_DE_MASSFLOW_PLANE bug if DE injection is defined.
    - Add CID_RCF to *DEFINE_DE_TO_SURFACE_COUPLING for force output in local coordinates to 'demrcf' file.
    - Update the *DEFINE_DE_BY_PART card so that it matches the capabilities of the *CONTROL_DISCRETE_ELEMENT card.
    - Add penalty stiffness scale factor, thickness scale factor, birth time and death time to *DEFINE_DE_TO_SURFACE_COUPLING.
    - Add dynamic coefficient of friction to *CONTROL_DISCRETE_ELEMENT.
    - Implement Finnie's wear law and user defined wear model to *DEFINE_DE_TO_SURFACE_COUPLING.
    - Implement user-defined curve for DEM frictional coefficient as function of time.
    - Implement user-defined curve for contact force calculation for *CONTROL_DISCRETE_ELEMENT.
    - Fix inconsistent results between *DEFINE_DE_BY_PART and *CONTROL_DISCRETE_ELEMENT.

- *ELEMENT
  - Fixed bug affecting output from beam elements ELFORM = 2 when certain uncommon inputs are present. Forces and moments in the output files could be wrongly rotated about the beam axis. This affected the output files only, not the solution inside LS-DYNA. The error could occur under two circumstances: (a) if IST on *SECTION_BEAM is non-zero, the output forces and moments are supposed to be rotated into the beam's principal axis system, but this rotation could be applied to the wrong beam elements; and (b) when no ELFORM = 2 elements have IST, but the model also contains beams with ELFORM = 6 and RRCON = 1 on the SECTION_BEAM card, some of the ELFORM = 2 elements can have their output forces and moments rotated by 1 radian.
  - Fix a bug affecting 2d seatbelt with time-dependent slipring friction.
  - Fix erroneous 1d seatbelt slipring message.
  - Fix seatbelt consistency issue in SMP (ncpu < 0).
  - Add error message when 2d seatbelt part doesn't have shell formulation of 5 and *MAT_SEATBELT.
  - Fix a bug for 2d seatbelt that could occur when a model has both 1d and 2d belts, and a 1d pretensioner of type 2, 3 or 9.
  - Fix an MPP seatbelt bug that could occur when using a type 9 pretensioner.
  - Allows shell formulation 9 to be used for 2d seatbelt. It was reset to formulation 5 by LS-DYNA, no matter what formulation was input. Now, only formulation 5 and 9 are accepted as input. Other formulations will incur error message.
  - MPP now supports *ELEMENT_MASS_MATRIX_NODE_(SET).
  - Added cohesive shell formulation -29. This formulation uses a cohesive midlayer where local direction q1 coincides with the average of the surrounding shell normals. This formulation is better suited for simulating normal shear.
  - Cohesive shell formulation +/-29: Fixed absence of part mass in d3hsp.
  - Make *TERMINATION_DELETED_SOLIDS work with hex spot weld failures.
  - Fix incorrect load curve used if large value is used for FC < 0 and/or FCS < 0 in *ELEMENT_SEATBELT_SLIPRING.
  - Fix incorrect velocity on accelerometer if
    - velocity is prescribed on the rigid body that the accelerometer is attached to, and
    - INTOPT = 1 in *ELEMENT_SEATBELT_ACCELEROMETER, and
    - *INITIAL_VELOCITY_GENERATION_START_TIME is used.
  - Fix incorrect discrete spring behavior when used with adaptivity.

- Fix input error when using *DEFINE_ELEMENT_DEATH with BOXID > 0 for MPP.
- Modify tolerances on error messages SOL+865 and SOL+866 to prevent unnecessary error terminations when translational or rotational mass of a discrete beam was close to zero.
- Made the solid element negative volume warning SOL+630 for penta formulatgion 15 consistent with the volume calculation in the element. With this change, elements are deleted rather than the job terminating with error SOL+509.
- Fixed the default hourglass control for shell form 16. It was defaulting to type 5 hourglass control rather than 8.
- Fixed default hourglass control when the *HOURGLASS control card is used but no HG type is specified. We were setting to type 1 instead of 2. Also, fixed the default HG types to match the user's manual for implicit and explicit.
- Fixed the fully integrated membrane element (shell ELFORM = 9) when used with NFAIL4 = 1 on *CONTROL_SHELL and there are triangular elements in the mesh. Triangular elements were being deleted by the distorted element check.
- Fixed a divide by zero error that occurred with *SECTION_BEAM, ELFORM = 6, SCOOR = 12 or -12, and node 3 was omitted on *ELEMENT_BEAM, and nodes 1 and 2 are along the global y-direction or z-direction.
- Fixed laminated shell theory for type 6 and 7 shell elements when made active by LAMSHT = 3 or 5 on *CONTROL_SHELL.
- Added an int.pt. variable for *PART_COMPOSITE_LONG and *PART_COMPOSITE_TSHELL_LONG called SHRFAC which is a scale factor for the out-of-plane shear stress that allows the user to choose the stress distribution through thickness. This was motivated by test data that shows that for large differences is layer shear stiffness, the parabolic assumption is poor.
- Fixed implicit hourglass stiffness in viscoelastic materials when used with tshell forms 5 or 6. The stiffness was much too small.
- Modified tshell type 5 to use the tangent stiffness for calculating the Poisson's affects and hourglass control for *MAT_024. This makes the behavior softer during buckling which is much more realistic.
- Fixed a significant bug in segment based contact when SHLEDG = 1 and SBOPT = 3 or 5 and DEPTH < 45, and shell segments in contact have different thicknesses. A penetration check was using incorrect thicknesses causing contact to be detected too late, particularly for edge to surface contact.
- Improved the time step calculation for triangular tshell elements. The time step was too conservative for elements with significant thickness. This fix does not affect tshell type 7.

- Fixed all tshells to work with anisotropic thermal strains which can be defined by *MAT_ADD_THERMAL. Also, this now works by layer for layered composites.
- Enabled tshell form 5 to recalculate shear stiffness scale factors when plasticity material models 3, 18, 24, 123, or 165 are included in a composite section. Prior to this change the scale factors were based on elastic properties so after yielding, the stress distribution was not what was expected. This new capability supports the constant stress option, the parabolic option, and the SHRFAC option on *PART_COMPOSITE_TSHELL_LONG.
- Improved tshell 5 when used with mixed materials in the layers. A failure to use the correct Poisson's ratio was causing a less accurate stress tensor.
- Modified the time step calculation for tshell forms 3 and 5. A dependence on volumetric strain rate was removed in order to prevent oscillations in the time step which caused stability problems, particularly for tshell 5.
- Fixed tshell constant shear stress option (TSHEAR = 1 on *SECTION_TSELL or *PART_COMPOSITE). It was producing a not very constant stress distribution.
- Fixed stress and strain output of tshells when the composite material flag CMPFLG is set on *DATABASE_EXTENT_BINARY. The transformation was backwards.
- Fixed mass of parts reported to d3hsp when *ELEMENT_SHELL_SOURCE_SINK is used. The mass of inactive elements was being included.
- Enabled *MAT_026 and *MAT_126 (HONEYCOMB) to be used with tshell forms 3, 5, and 7. It was initialized incorrectly causing a zero stress.
- Added a missing internal energy calculation for tshell form 6.
- Enabled tshell forms 1, 2, and 6 to work with material types 54, 55, and 56.
- Modified the z-strain distribution in tshell forms 5 and 6 when used in composites with mixed materials that are isotropic. The existing assumed strain scheme was doing a poor job of creating a constant z-stress through the thickness.
- Increased the explicit solution time step for thin shell composite elements. The existing method calculated a sound speed using the stiffness from the stiffest layer and dividing it by the average density of all layers. This could be overly conservative for composites with soft layers of low density. The new method uses the average stiffness divided by average density. This is still conservative, but less so.
- Corrected rotational inertia of thin shells when layers have mixed density and the outer layers are denser than inner layers. The fix will mostly affect elements that are very thick relative to edge length.
- Added support for *ELEMENT_SHELL_SOURCE_SINK to type 2 shells with BWC = 1 on *CONTROL_SHELL.
- Prevent inactive shell elements (from *ELEMENT_SHELL_SOURCE_SINK) from controlling the solution time step.

- Fixed *LOAD_STEADY_STATE_ROLLING when used with shell form 2 and Belytschko- Wong-Chang warping stiffness (BWC = 1 *CONTROL_SHELL). The load was not being applied.
- Improved the brick element volume calculation that is used by the option erode elements (ERODE = 1 on *CONTROL_TIMESTEP or PSFAIL.ne.0 on *CONTROL_SOLID). It was not consistent with the element calculation which caused an error termination.
- Fixed all tshell forms to work with anisotropic thermal strains which can be defined by *MAT_ADD_THERMAL. Also, this now works by layer for layered composites.
- Reworked shell output so that we can correctly output stress in triangular shells when triangle sorting is active, that is when ESORT = 1 or 2 on *CONTROL_SHELL.
- *ELEMENT_T/SHELL_COMPOSITE(_LONG) and *PART_COMPOSITE_T/SHELL_(LONG): Permit the definition of zero thickness layers in the stacking sequence. This allows the number of integration points to remain constant even as the number of physical plies varies and eases post-processing since a particular integration point corresponds to a physical ply. Such a capability is important when plies are not continuous across a composite structure.

  To represent a missing ply, set THK to 0.0 for the corresponding integration point and additionally, either set MID = -1 or set PLYID to any nonzero value. Obviously, the PLYID option applies only to the keywords containing LONG.
- Implemented sum factorization for 27-node quadratic solid that may increase speed by a factor of 2 or 3.
- Support second order solid elements (formulations 23,24,25,26) for *SET_NODE_GENERAL.
- Invoke consistent mass matrix of 27-node hex element for implicit dynamics and eigenvalues.
- Reorder node numbering when assembling global stiffness matrix for 27-node hex. This fixes a bug in which it was reported than the implicit 27-node element didn't work
- Automatically transfer nodal boundary conditions for newly generated nodes if H8TOH27 option is used in *ELEMENT_SOLID.
- Modify initialization of material directions for solid elements. If there are only zeros for all the 6 values in *INITIAL_STRESS_SOLID, then the values from the other input (e.g. *ELEMENT_SOLID_ORTHO) are kept.
- Enable *PART_STACKED_ELEMENTS to pile up shell element layers. Before, it was necessary that solid element layers were placed between shell element layers. Now, shell element layers can follow each other directly. Contact definitions have to be done separately.
- Allow *PART_STACKED_ELEMENTS to be used in adaptive refinement simulations.

  - Add alternative mass calculation for critical time step estimate of cohesive elements. This hopefully resolves rarely occurring instability issues. Option ICOH on *CONTROL_SOLID is used for that.
  - Correct the strain calculation for tet formulation 13. This did not affect the stress response, only output of strains. Nodal averaging was not accounted for.
  - User defined elements (ELFORM = 101 to 105 on *SECTION_SHELL/SOLID) can now be used together with *MAT_ADD_EROSION.
  - Add option to define a pull-out force vs. time curve in *ELEMENT_BEAM_SOURCE by defining a negative variable FPULL. |FPULL| can refer to *DEFINE_CURVE or *DEFINE_CURVE_FUNCTION.
  - Solid tet form 13 supported for all materials in implicit, including a presumable consistency improvement for the future.
  - The Hughes-Liu beam is supported in *INTEGRATION_BEAM such each integration point may refer to a different part ID and thus have a different coef. Of thermal expansion. See *MAT_ADD_THERMAL_EXPANSION.
  - Shell types 2 and 16 that combines thermal expansion and thick thermal shells, see *MAT_ADD_THERMAL_EXPANSION and TSHELL on *CONTROL_SHELL, now correctly treat temperature gradient through the thickness to create bending moments. All shell types are to be supported in due time.
  - *SECTION_BEAM_AISC now provides predefined length conversion factors for specific unit systems.
  - 3D tet r-adaptivity now supports *DEFINE_BOX_ADAPTIVE.

    - For every adaptive part, users can define multiple boxes where different BRMIN & BRMAX (corresponding to RMIN & RMAX in *CONTROL_REMESHING) can be specified for 3D tet remesher to adjust the mesh size.
    - Current implementation does not support LOCAL option.

  - Fix bug in 3D adaptivity so that users can now have both non-adaptive tshell parts and 3D adaptive parts in one analysis.
  - Fix the bug in 3D adaptivity so that users can now have both dummy nodes and 3D adaptive parts in one analysis.

- ***EM (Electromagnetic Solver)**

  - Randles Circuits for Battery Modeling

    - A Randles circuit is an equivalent electrical circuit that consists of an active electrolyte resistance r0 in series with the parallel combination of the capacitance c10 and an impedance r10. The idea of the distributed Randles model is to use a certain number of Randles circuits between corresponding nodes on the two current collectors of a battery unit cell. These Randles circuits model the electrochemistry that happens in the electrodes and separator between the current collectors. The EM solver can then solve for the EM fields in the current collectors, and the connections between them.

- Added analysis of distributed Randles circuits to MPP.
- Added d3plot output for distributed Randles circuits:
  D3PL_RAND_r0_EM,
  D3PL_RAND_r10_EM,
  D3PL_RAND_c10_EM,
  D3PL_RAND_soc_EM,
  D3PL_RAND_i_EM,
  D3PL_RAND_u_EM,
  D3PL_RAND_v_EM,
  D3PL_RAND_vc_EM,
  D3PL_RAND_temperature_EM,
  D3PL_RAND_P_JHR_EM,
  D3PL_RAND_P_dudt_EM,
  D3PL_RAND_i_vector_EM
  This output can be visualized in LS-PrePost versions 4.3 and 4.5 on the separator part of the battery cell using Post/FriComp/Extend/EM node.
- Added tshells for EM analysis for use in battery modeling.
- Added new capability for modeling Randles short, based on *DEFINE_FUNCTION so that the user has a lot of freedom to define where and when the short happens as well as the short resistance.
- Added a new capability for battery exothermal reactions also based on *DEFINE_FUNCTION. The new keyword *RANDLE_EXOTHERMAL_REACTION makes it possible to complement the heating of a short circuit created by a short by exothermal reactions if, for example, the temperature becomes higher than a threshold value.

- **FORMING ANALYSIS**

  - Extend *INCLUDE_AUTO_OFFSET to solid and beam elements (draw beads).
  - Add a new keyword for springback compensation:*INTERFACE_COMPENSATION_NEW_REFINE_RIGID to refine and break rigid tool mesh along the user supplied trim curves so compensated tool mesh follows exactly the blank mesh (file "disp.tmp"). This needs to be done only once in the beginning of the springback compensation (ITER0).
  - *CONTROL_FORMING_ONESTEP:

    - Change the default element formulation option for onestep method to QUAD2.
    - Add a new option QUAD to allow quadrilateral elements to be considered.
    - Limit the maximum thickening by using a new variable TSCLMAX for the sheet blank.
    - Set the value of OPTION to a negative value to output the file 'onestepresult' in large format (E20.0).
    - Calculate and add the damage factor and output to the 6th history variable in the output file "onestepresult". Add the variable for a curve ID to define the fracture strain vs. triaxility. Add another variable DMGEXP (damage parameter), as used in GISSMO model.
    - Keep the original coordinates for the onestep output "onestepresults".

- Add a new option VECTOR to *CONTROL_FORMING_BESTFIT to output deviation vector (in the format of: NODEID, xdelta, ydelta, zdelta) for each node to its closest target element. The deviation vectors are output under the keyword *NODE_TO_TARGET_VECTOR.
- *CONTROL_FORMING_OUTPUT:

    - Output will skip any negative abscissa (Y1) value.
    - When CIDT < 0, the positive value defines the time dependent load curve.

- Add a warning in springback compensation *INTERFACE_SPRINGBACK_COMPENSATION to identify which input file (typically the blank with adaptive mesh not output directly by LS-DYNA) has the wrong adaptive constraints.
- *INTERFACE_COMPENSATION_3D: turn off the output of nikin file.
- *ELEMENT_LANCING:

    - Allow some unused lancing curves to be included in the input.
    - When the gap between the two ends of a lancing curve is not zero, but small enough, then this curve is automatically closed.
    - Allow several parts to be cut during lancing; the parts can be grouped in *SET_PART_LIST, and defined using a negative value IDPT.
    - Specify the distance to bottom dead center as AT and ENDT when the new variable CIVD is defined.
    - Set IREFINE = 1 (default) in lancing, to refine blank mesh automatically along the lancing curves.
    - Re-set the adaptive level to be 1 to prevent those elements along the lancing route to be further refined.

    - When IREFINE = 1, elements along the lancing curve will be refined to make sure that no adapted nodes exist in the neighborhood. This helps get improved lancing boundary.
    - Change of tolerance for lancing to merge the small elements into bigger ones.

- Add a new keyword to perform trimming after lancing (shell elements only): *DEFINE_LANCE_SEED_POINT_COORDINATES. Maximum of two seed nodes can be defined.
- Extend *CONTROL_FORMING_TOLERANC to *MAT_036, *MAT_037, *MAT_125, and *MAT_226. When beta is less than -0.5, there is no necking and no calculation of FI. When beta is greater than 1.0, beta = 1.0/beta. This keyword adds a smoothing method to calculate the strain ratios for a better formability index.
- Sandwiched parts (*CONTROL_ADAPTIVE, *DEFINE_CURVE_TRIM):

    - Disable *CONTROL_ADAPTIVE_CURVEs for sandwich parts, since refinement along the curve is automatically done during trimming.
    - Refine the elements along the trimming curve to make sure no slave nodes are be cut by trimming curves.
    - Allow mesh adaptivity.
    - Allow multi-layers of solids.
    - Add a check to the variable IFSAND in *CONTROL_ADAPTIVE for sandwich part to be refined to exclude solid elements.

- Solid element trimming (*DEFINE_CURVE_TRIM):

    - Refine those elements along the trimming curve.
    - Improve solid trimmig to allow the trimming of one panel into two panels with two seed nodes.

- Add a new keyword *CONTROL_FORMING_REMOVE_ADAPTIVE_CONSTRAINTS to remove adaptive constraints on a formed, adapted blank, and replaced them with triangular elements.
- *DEFINE_CURVE_TRIM_NEW: Allow trimming of tshells.
- Add a new keyword:*INTERFACE_WELDLINE_DEVELOPMENT to obtain initial weld line from the final part and the final weld line position.

    - When Ioption = -1, convert weld line from its initial position to the final part.
    - Output the element nodes that intersect the weld line in the final part, and the output file is: affectednd_f.ibo
    - Output the element nodes that intersect the weld line in the initial part, and the output file is: affectednd_i.ibo

- Add a new variable DT0 to *CONTROL_IMPLICIT_FORMING so there is no need to use *CONTROL_IMPLICIT_GENERAL to specify DT0.
- *INTERFACE_BLANKSIZE:
    - Add a new feature DEVELOPMENT option. When ORIENT = 2, then a reference mesh file for the formed part should be included. The calculated and compensated boundary will be based on the reference mesh.
    - Add a new option SCALE_FACTOR that allows the target curve to be moved. This is useful when multiple target curves (e.g. holes) and formed curves are far away from each other.

- ***FREQUENCY_DOMAIN**
    - Added new keyword *CONTROL_FREQUENCY_DOMAIN to define global control parameters for frequency domain analysis. Currently two parameters are defined:
        - REFGEO: flag for reference geometry in acoustic eigenvalue analysis (either the original geometry at t = 0, or the deformed geometry at the end of transient analysis).
        - MPN: large mass added per node, to be used in large mass method for enforced motion.
    - *FREQUENCY_DOMAIN_ACOUSTIC_BEM:
        - Enabled using incident wave (*FREQUENCY_DOMAIN_ACOUSTIC_INCIDENT_WAVE) in Rayleigh method (METHOD = 0).
        - Enabled acoustic pressure fringe plot (*FREQUENCY_DOMAIN_ACOUSTIC_FRINGE_PLOT) in Rayleigh method (METHOD = 0).
        - Fixed bug in running acoustic analysis with multiple boundary conditions in MPP.
        - Fixed running MATV (Modal Acoustic Transfer Vector) approach in MPP (*FREQUENCY_DOMAIN_ACOUSTIC_BEM_MATV).
        - Added treatment for triangular elements used in Rayleigh method (METHOD = 0).
        - Added output of acoustic intensity to binary database D3ACS (defined by *DATABASE_FREQUENCY_BINARY_D3ACS).
        - Fixed bug in acoustic pressure fringe plot for collocation BEM (METHOD = 3) and dual BEM based on Burton-Miller formulation (METHOD = 4).
    - *FREQUENCY_DOMAIN_ACOUSTIC_FEM:

    - Fixed bug in acoustic analysis by FEM, when dimensions of mass and k (stiffness) matrices are mismatched.

- *FREQUENCY_DOMAIN_ACOUSTIC_FRINGE_PLOT:

    - Implemented acoustic fringe plot for MPP for the options PART, PART_SET, and NODE_SET.

- *FREQUENCY_DOMAIN_FRF:

    - Added new loading types:
      VAD1 = 5: enforced velocity by large mass method
      = 6: enforced acceleration by large mass method
      = 7: enforced displacement by large mass method
      = 8: torque
      = 9: base angular velocity
      = 10: base angular acceleration
      = 11: base angular displacement
    - Added rotational dof output for FRF.

- *FREQUENCY_DOMAIN_MODE:

    - Added option _EXCLUDE to exclude some eigenmodes in modal superposition in frequency domain analysis.

- *FREQUENCY_DOMAIN_RANDOM_VIBRATION:

    - Fixed bug in running random vibration with random pressure wave load (VAFLAG = 2) in MPP.
    - Improved random vibration analysis by allowing using complex variable cross PSD functions. Previously cross PSD was defined as real variables thus the phase difference was ignored.
    - Added PSD and RMS computation for Von Mises stress in beam elements.

- *FREQUENCY_DOMAIN_RESPONSE_SPECTRUM:

    - Added Von Mises stress output for beam elements in database D3SPCM.
    - Corrected computation of response spectrum at an intermediate damping value by interpolating spectra at two adjacent damping values. Now the algorithm is based on ASCE 4-98 standard.

- *FREQUENCY_DOMAIN_SSD:

    - Added new loading types:
      VAD = 5: enforced velocity by large mass method

= 6: enforced acceleration by large mass method
= 7: enforced displacement by large mass method
= 8: torque

    - Fix for running SSD fatigue in MPP (affected keyword: *FREQUENCY_DOMAIN_SSD_FATIGUE).
    - Updated ssd computation with local damping, and enabled the restart feature by reading damping matrix.
    - Implemented ERP (Equivalent Radiated Power, keyword *FREQUENCY_DOMAIN_SSD_ERP) for MPP.

  - *DATABASE_FREQUENCY_ASCII:

    - Added keyword *DATABASE_FREQUENCY_ASCII_{OPTION} to define the frequency range for writing frequency domain ASCII databases NODOUT_SSD, ELOUT_SSD, NODOUT_PSD and ELOUT_PSD.

- ***ICFD (Incompressible Fluid Solver)**

  - New ICFD features and major modifications

    - Simple restart is now supported for ICFD.
    - Added wave damping capabilities. See *ICFD_DEFINE_WAVE_DAMPING.
    - Added steady state solver. See *ICFD_CONTROL_GENERAL and *ICFD_CONTROL_STEADY.
    - Added steady state potential flow solver. See *ICFD_CONTROL_GENERAL.
    - Weak thermal coupling for conjugate heat transfer is now possible in addition to the classic monolithic approach. See *ICFD_CONTROL_CONJ.
    - Windkessel boundary conditions are now available for blood flow. See *ICFD_BOUNDARY_WINDKESSEL.
    - It is now possible to output the heat transfer coefficient as a surface variable in LSPP or in ASCII format on segment sets for a subsequent solid-thermal only analysis. See *ICFD_DATABASE_HTC.
    - Two way coupling is now possible with DEM particles. See *ICFD_CONTROL_DEM_COUPLING.
    - Modifications introduced in the SUPG stabilization term used in thermal and conjugate heat transfer problems for improved accuracy and speed.

  - Additions and modifications to existing ICFD keywords

    - *ICFD_BOUNDARY_FSWAVE:

Added a boundary condition for wave generation of 2nd order stokes waves with free surfaces

- *ICFD_CONTROL_FSI:
  Added a flag which, when turned on will project the nodes of the CFD domain that are at the FSI interface onto the structural mesh. This is recommended for cases with rotation.
- *ICFD_CONTROL_MESH:
  Added a flag to allow the user control over whether there will be re-mesh or not. If there is no re-mesh then we can free space used to backup the mesh and lower memory consumption.
- *ICFD_CONTROL_MESH_MOVE:
  Added option to force the solver to turn off any mesh displacements. This can be useful in cases where the mesh is static to save a little bit of calculation time.
- *ICFD_CONTROL_OUTPUT:
  Added option to support output in Fieldview format, binary and ASCII.
  When output of the fluid volume mesh is requested, the mesh will be divided into ten distinct parts, grouping elements in ten deciles based on the mesh quality (Part 1 has the best quality elements, part 10 the worst).
- *ICFD_CONTROL_POROUS:
  Improvements for RTM problems.
- *ICFD_CONTROL_TIME:
  Added an option to define an initial timestep.
  Added an option to shut off the calculation of Navier Stokes after a certain time leaving only the heat equation. This can be useful to save calculation times in conjugate heat transfer cases where the fluid often reaches steady state before the thermal problem.
- *ICFD_DATABASE_DRAG:
  It is now possible to output the force on segment sets in a FSI run directly in LS-DYNA compatible format. This can be useful for a subsequent linear FSI analysis running only the solid mechanics part.
  Added flag to output drag as a surface variable in LSPP.
- *ICFD_DATABASE_FLUX:
  Added option to change output frequency
- *ICFD_DATABASE_NODOUT:
  The user node IDs are now required rather than the internal node IDs
- *ICFD_CONTROL_IMPOSED_MOVE:
  Added the option to choose between imposing the displacements or the velocity.
- *ICFD_CONTROL_TRANSIENT:
  Choose implicit time integration scheme for NS.
- *ICFD_CONTROL_DEM_COUPLING:

Added a scale factor for the sphere radius in the computation of the DEM force.

- *ICFD_MODEL_POROUS:
  Added a scale factor option on the permeability for model 1 and 2. A *DEFINE_FUNCTION can also be used.
- *MESH_BL:
  Added option to generate boundary layer mesh using a growth factor.

- ICFD bug fixes and minor improvements

  - Fixed bug when multiple *DEFINE_FUNCTIONs were used in an ICFD problem. Only the last one was taken into account.
  - LES turbulence model: fixed van Driest damping issue in the boundary layer. LES models can use wall functions.
  - RANS turbulence models: Standard k-epsilon, realizable k-epsilon, Wilcox k-omega uses HRN laws of the wall by default while SST and Spalart Allmaras use LRN. Improvements on the convergence of all those models.
  - The DEM particle volume is now taken into account in free surface problems.
  - Average shear is now output as a surface variable in the d3plots.
  - *ICFD_CONTROL_MONOLITHIC is obsolete (replaced by *ICFD_CONTROL_GENERAL).
  - Added more output for the mesh generation indicating the stage of the meshing process and the amount of elements that are being generated as a multiple of 10000. Added progress % for the extrusion of the mesh during the BL mesh generation.
  - Improvements on the element assemble speed in MPP.
  - Fixed synchronization problem for the last timestep in an FSI problem.
  - More options have been added to the timer output.
  - Correction of the calculation of the flux in *ICFD_DATABASE_FLUX in free surface cases.
  - Boundary layer mesh can go through free-surfaces or mesh size interfaces.
  - The Center of Gravity of the fluid is output in the icfd_lsvol.dat ASCII file in free surface problems

- **Implicit (Mechanical) Solver**

  - Enhanced termination of MPP eigensolver when non eigenmodes are found.

- Implicit was enforcing birth and death times on *BOUNDARY_SPC during dynamic relaxation contrary to the User's Manual. These times are now ignored by implicit during dynamic relaxation.
- Corrected output of eigenvalues and frequencies to file eigout for the asymmetric eigenvalue problem.
- Enhanced logic that determines when to write out the last state to d3plot for implicit.
- Improved error message for reading d3eigv file for *PART_MODES for the case when the user inputs a d3eigv file from a different model than intended.
- Corrected the reporting of kinetic and internal energy in file glstat for implicit.
- Applied corrections to tied contact in implicit (MPP). This affects slave nodes coming from other processes.
- Corrected output to file d3iter (implicit nonlinear search vectors) for restart.
- Enhanced termination process when the implicit solver determined an early termination.
- When implicit springback was following an explicit transient step, the implicit keywords with the _SPR were not properly handled. This is now corrected.
- Added a warning about the combined use of *CONSTRAINED_RIGID_BODY_STOPPERS and the Lagrange multiplier formulation for joints (*CONTROL_RIGID) for explicit. The warning recommends switching to the penalty joint formulation.
- Applied numerous bug fixes to the implicit solver associated with *CONSTRAINED_INTERPOLATION where there are lots of independent degrees-of-freedom.
- Corrected initialization of MPP tied contact with implicit mechanics when the implicit phase follows explicit dynamic relaxation.
- Fixed an implicit problem where a linear implicit analysis follows inertia relief computation.
- Added gathering of damping terms from discrete elements from implicit especially for FRF computations and matrix dumping.
- Fixed Implicit for the case of multi-step Linear (NSOLVR = 1) with Intermittent Eigenvalue Computation.
- Corrected the output to d3iter when 10-noded tets are present.
- Keypoints specified in *CONTROL_IMPLICIT_AUTO are now enforced at the initial time step and on restart from explicit.
- Skip frequency damping during implicit static dynamic relaxation, i.e. IDRFLAG > 5.
- *CONTROL_IMPLICIT_ROTATIONAL_DYNAMICS:

    - The VID of the rotating axis can now be defined by both *DEFINE_VECTOR and *DEFINE_VECTOR_NODES. It enables the

        movement of the rotating axis. Previously, only *DEFINE_VECTOR could be used to define the VID.
    - The rotational dynamics now work in MPP.

- Shell forms 23 and 24 (high order shells), 1D seatbelts, Hughes-Liu and spotweld beams (types 1 and 9) are now supported with the implicit accuracy option (IACC = 1 in *CONTROL_ACCURACY) to render strong objectivity for large rigid body rotations. Also, shell type 16 is supported with implicit accuracy option, resulting in forms 16 and -16 giving the same solution.
- Translational and generalized stiffness joints are now strongly objective for implicit analysis. See CONSTRAINED_JOINT_STIFFNESS....
- In implicit it may happen that the initial loads are zero, for instance in forming problems. In addition, the goal is to move a tool in contact with a workpiece, and the way line search and convergence works, it is hard to get things going. We now attempt to handle this situation by automatically associating an augmented load to the prescribed motion simply to get off the ground.
- New tolerances on maximum norms are introduced for convergence in implicit: ratio of max displacement/energy/residual, and absolute values of nodal and rigid body translation/rotational residual can be specified. See DNORM.LT.0 on *CONTROL_IMPLICIT_SOLUTION for defining an additional card for these parameters DMTOL, EMTOL and RMTOL. Furthermore, maximum absolute tolerances on individual nodal or rigid body parameters can be set on NTTOL, NRTOL, RTTOL and RRTOL on the same card.
- If ALPHA < 0 on first *CONTROL_IMPLICIT_DYNAMICS card, the HHT implicit time integration scheme is activated.

- ***INITIAL**

    - Fix *INITIAL_VELOCITY_GENERATION when used with *INCLUDE_TRANSFORM, which was broken due to misplaced conditionals in r100504.
    - Fix 3 bugs for *INITIAL_VELOCITY_GENERATION involving omega > 0 and icid > 0:

        - When nx = -999. Now the directional cosine defined by node NY to node NZ will be the final direction to rotate about. In other words, the direction from node NY to node NZ will not be projected along icid any more.
        - When nx != -999, (xc,yc,zc) should not be rotated along icid, since (xc,yc,zc) are global coordinates.
        - When *INITIAL_VELOCITY_GENERATION is included by *INCLUDE_TRANSFORM, (xc,yc,zc) is transformed.

- ○ Add the option of ramping time steps, ndtrrg, for *INITIAL_FOAM_REFERENCE_GEOMETRY. The solid elements with reference geometry and ndtrrg > 0 will restore its reference geometry in ndtrrg time steps.
- ○ Fix incorrect initial velocity when ICID.ne.0 in *INITIAL_VELOCITY_GENERATION, and rotational velocity, omega, is not zero and *PART_INERTIA is also present.
- ○ Add variable IZSHEAR in *INITIAL_STRESS_SECTION to initialize shear stress.
- ○ Fix incorrect initial velocity for *INITIAL_VELOCITY if IRIGID = -2 and ICID > 0.
- ○ Fix incorrect NPLANE and NTHICK for *INITIAL_STRESS_SHELL when writing dynain for shell type 9.
- ○ Fix *INITIAL_STRAIN_SHELL output to dynain for shell types 12 to 15 in 2D analysis. Write out strain at only 1 intg point if INTSTRN = 0 in *INTERFACE_SPRINGBACK_LSDYNA and all strains at all 4 intg points if INTSTRN = 1 and nip = 4 in *SECTION_SHELL.
- ○ Skip transformation of the initial velocities if ICID > 0 and *INCLUDE_TRANSFORM is used to transform the keyword input file with the *INITIAL_VELOCITY.... keyword. Also echo warning message, KEY+1109, that the transformation will be skipped since icid is specified.
- ○ Fix incorrect transformation of *DEFINE_BOX which results in incorrect initial velocities if the box is used in *INITIAL_VELOCITY.
- ○ Fixed *INITIAL_STRESS_DEPTH when used with 2D plane strain and axisymmetric elements. The prestress was being zeroed.
- ○ Improved the precision of the initial deformation calculation for *INITIAL_FOAM_REFERENCE_GEOMETRY in the single precision version.
- ○ Fixed stress initialization (*INITIAL_STRESS_SECTION) for type 13 tet elements. The pressure smoothing was causing incorrect pressure values in the elements adjacent to the prescribed elements.
- ○ Add _SET option to *INITIAL_STRESS_SOLID for element sets.
- ○ Fix bug in 3D adaptivity so that users can now define *INITIAL_TEMPERATURE for adaptive parts.

- **Isogeometric Elements**

  - ○ The stability of the trimmed NURBS shell patches has been improved.
  - ○ Add *LOAD_NURBS_SHELL to apply traction type loading directly on the surface of NURBS shell.
  - ○ Users can use the PART option of *SET_SEGMENT_GENERAL to define segment set of a NURBS patch. The segment set will contain all segments of interpolated null shell elements.
  - ○ *ELEMENT_SOLID_NURBS_PATCH:

        - Isogeometric solid analysis implemented for MPP.
        - Isogeometric solid analysis implemented for SMP with multiple CPUs, including consistency (ncpu < 0).
        - Activate user-defined materials for isogeometric solid.

    - *ELEMENT_SHELL_NURBS_PATCH:

        - Isogeometric shell analysis now implemented for SMP with multiple CPUs, including consistency (ncpu < 0).
        - Add a power iteration method to get the maximum eigen-frequency for each isogeometric element. This will be used to set a reasonable time step for trimmed elements.

    - *ELEMENT_SHELL_NURBS_PATCH:

        - Changed the way of projecting the results from isogeometric (NURBS) elements to the interpolation elements. Now a background mesh, spanned over the locations of the integration points of the isogeometric (NURBS) elements serves as basis to interpolate results from the integration points to the centroid of the interpolation elements. This change may lead to slightly different post-processing results in the interpolation elements.

    - Add support for trimmed NURBS to work in single precision. Anyway, it is still recommended to use double precision versions for trimmed NURBS patches.
    - Add post-processing of strains and thickness for interpolation shells.

- ***LOAD**

    - Fixed bugs affecting discrete beam elements (ELFORM = 6) when used with staged construction. Here, "dormant" refers to elements that have not yet become active as defined on *DEFINE_STAGED_CONSTRUCTION_PART.

        - Dormant discrete beams could still control the timestep and attract mass-scaling, when they should not do so.
        - Dormant discrete beams reaching a failure criterion defined on the *MAT card were deleted, when they should not be.
        - The displacements output (see *DATABASE_DISBOUT) included displacements occurring while the elements were dormant. Now, the output displacements are reset to zero at the moment the element becomes active.

  - Fixed bug in Staged Construction: if FACT on *CONTROL_STAGED_CONSTRUCTION had been left blank, and Dynamic Relaxation was active, an error termination occurred.
  - Fixed bug: *LOAD_GRAVITY_PART (and also gravity loading applied by *DEFINE_STAGED_CONSTRUCTION_PART) was failing to account for non-structural mass when calculating gravity load: NSM on *SECTION_BEAM and MAREA on *SECTION_SHELL.
  - Fixed bug in *LOAD_VOLUME_LOSS: inconsistent results when run in SMP parallel.
  - Fix bugs affecting *LOAD_SEGMENT_FILE:
    - Remove LOAD_SEGMENT_FILE file size limit (It used to be 200M).
    - Apply correct pressure on the shared boundary between processors.
  - Fix GRAV = 1 in *PART which was not were not working correctly with *LOAD_DENSITY_DEPTH. Make *LOAD_DENSITY_DEPTH work for Lagrangian 2D elements.
  - Fix insufficient memory error,SOL+659, when using *LOAD_ERODING_PART_SET with MPP.
  - Fix incorrect loading when using *LOAD_ERODING_PART_SET with BOXID defined.
  - Added *LOAD_SUPERPLASTIC_FORMING for implicit analysis.
  - *LOAD_SUPERPLASTIC_FORMING box option now works in MPP.

- ***MAT and *EOS**
  - *MAT_197 (*MAT_SEISMIC_ISOLATOR) could become unstable when the parameter DAMP was left at its default value. A workaround was to input DAMP as a small value such as 0.05. The timestep for *MAT_197 is now smaller than previously, irrespective of the DAMP setting, and the behavior is now stable even if DAMP is left at the default.
  - Fixed bug: Timestep calculation was wrong for *MAT_089 solid elements. Response could be unstable especially for higher values of Poisson's ratio, e.g. 0.4.
  - Fixed bug: An error trap was wrongly preventing ELFORM = 15 for *MAT_169 (*MAT_ARUP_ADHESIVE). Wedge elements with ELFORM = 15 are now permitted.
  - *MAT_172 (*MAT_CONCRETE_EC2):
    Note that items (1) and (2) below can lead to different results compared to previous versions of LS-DYNA.
    - (1) The number of potential cracks in MAT_172 shell elements has been increased from 2 to 4. MAT_172 uses a fixed crack model: once the first crack forms, it remains at the same fixed angle relative to the element axes. Further cracks can then form only at pre-defined an-

gles to the first crack. Previously, only one further crack could form, at 90 degrees to the first crack. Thus, if the loading direction subsequently changed so that the principal tension is at 45 degrees to the first crack, that stress could exceed the user-defined tensile strength by a considerable margin. Now, further cracks may form at 90, +45 and -45 degrees to the first crack. Although the maximum principal stress can still exceed the user-defined tensile strength, the "error" is much reduced. There is an option to revert to the 2-crack model as in R9 (to do this, add 100 to TYPEC).

- (2) Add element erosion to MAT_172. This change may lead to different results compared to previous versions, because erosion strain limits are now added by default. Elements are now deleted when crack-opening strain becomes very large, or the material is crushed beyond the spalling limit. Plastic strain in the rebar is considered too. Previously, these elements that have passed the point of being able to generate any stress to resist further deformation would remain in the calculation, and sometimes showed very large non-physical deformations and could even cause error terminations. Such elements would now be deleted automatically. Default values are present for the erosion strains but these can be overridden in the input data, see new input fields ERODET, ERODEC, ERODER.
- (3) New history variables 10,11,12 (maximum value so far of through-thickness shear stress). This is useful for checking results because MAT_172 cracks only in response to in-plane stress; before cracking occurs, the through-thickness shear capacity is unlimited. The data components are:
  Ex History Variable 10 - maximum out of 11 and 12
  Ex History Variable 11 - maximum absolute value of YZ shear stress
  Ex History Variable 12 - maximum absolute value of ZX shear stress
  These are in the element local axis system. Note that these variables are written only if TYPESC is zero or omitted. TYPESC is a pre-existing capability that requests a different type of shear check.
- (4) Fixed bug. Elastic stiffness for MAT_172 beams was not as described in the manual, and the axial response could sometimes become unstable. The bug did not affect shell elements, only beams.
- (5) *MAT_172 can now handle models with temperatures defined in Kelvin (necessary if the model also has heat transfer by radiation). *MAT_172 has thermally-sensitive material properties hard-wired to assume temperatures in Centigrade. A new input TMPOFF in *MAT_172 offsets the model temperatures before calculating the material properties.
- (6) When the input parameter AGGSZ is defined, the maximum shear stress that can be transferred across closed cracks is calculated from a formula that has tensile strength and compressive stress as inputs. In MAT_172, the tensile strength of concrete is reduced when compres-

sive damage has occurred (see description of UNLFAC). Up to now, compressive damage was therefore influencing the maximum shear across cracks. However, the Norwegian standard from which the shear formula is taken treats the tensile strength as a constant. Therefore, for the purpose of calculating the maximum shear stress across closed cracks only, the compressive damage effect is now ignored.

- (7) Added capability for water pressure in cracks, for offshore applications. The water pressure is calculated from the depth of the element below the water surface (calculated from the z-coordinate). The water pressure is applied as a compressive stress perpendicular to the plane of any crack in the element. See new input fields WRO_G and ZSURF.

- *MAT_119 (*MAT__GENERAL_NONLINEAR_6DOF_DISCRETE_BEAM): Fixed bug in UNLOAD option 2. The bug occurs if an unloading curve has been left zero (e.g. LCIDTUR) while the corresponding loading curve was non-zero (e.g. LCIDTR), and UNLOAD = 2. Depending on the computer system, the symptoms could be harmless or the code could crash. Now, if the unloading curve is left blank, it is assumed to be the same as the loading curve i.e. load and unload up and down the same curve. That behavior was already implemented for UNLOAD = 1.

- Added Equation Of State 19 (*EOS_MURNAGHAN). Used extensively for fluid modeling in SPH through Weakly-Compressible formulation, in conjunction with SPH formulations 15 (fluid form) and 16 (normalized fluid form).
- *MAT_ADD_FATIGUE: Added a new form of Basquin equation to define material's SN curve: LCID = -3: S = a*N^b, where a and b are material constants.
- Add the option of A0REF for *MAT_FABRIC. That allows the option of using reference geometry to calculate A0 for the purpose of porosity leakage calculation.
- Add optional parameter DVMIN for *MAT_ADD_PORE_AIR to define the min volume ratio change to trigger pore air flow analysis.
- *DEFINE_HAZ_PROPERTIES:
- Distance of shell from the weld center is treated consistently under MPP and the shell material's yield stress is scaled properly.
- *MAT_168 and *MAT_279: Fixed support for element erosion.
- *MAT_092: Improved of implicit convergence for shells.
- *MAT_224: Fixed bug where wrong shear modulus was used in EOS.
- *MAT_270: Increased stability for thickness strain iterations for shells.
- *MAT_240: Added support for cohesive shell formulation +/-29.
- Scale load curve, LCSRS, of *MAT_ADD_EROSION when used with *INCLUDE_TRANSFORM.

- Fix incorrect results when using *MAT_TABULATED_JOHNSON_COOK/*MAT_224 with table LCKT defined and the first abscissa value, temperature, is negative.
- Fix spurious element deletion when using table for LCF in *MAT_TABULATED_JOHNSON_COOK/*MAT_224 and *MAT_TABULATED_JOHNSON_COOK_GYS/*MAT_224GYS.
- Error terminate with message, KEY+1142, if *MAT_ADD_EROSION is applied to resultant materials 28,116,117,118,130,139,166,170 and 98(with 1 intg point).
- Increase robustness of *MAT_033/*MAT_BARLAT_ANISOTROPIC_PLASTICITY for solids.
- Fix input error when using *MAT_ELASTIC_WITH_VISCOSITY_CURVE/*MAT_060c when LCID = 0.
- Fix seg fault when using shell type 15, axisymmetric volume weighted, with *MAT_ADD_EROSION and also materials with equation-of-states.
- Store computed yield strength as history variable #6 for *MAT_255.
- Fix inconsistency for *MAT_MODIFIED_PIECEWISE_LINEAR_PLASTICITY/*MAT_123 when ncpu < 0.
- Include original volume output to dynain file for 2D analysis when materials with an equation-of-state are used. This is needed to compute the deformation gradient when initializing a run using the dynain file.
- Fix improper stress initialization using *INITIAL_STRESS_SHELL via dynain for *MAT_018/*MAT_POWER_LAW_PLASTICITY with VP = 1.0.
- Make AOPT < 0 work for *MAT_170/*MAT_RESULTANT_ANISOTROPIC, i.e. with material coordinate system using *DEFINE_COORDINATE_(OPTION).
- Fix incorrect operation of TDEL for *MAT_MODIFIED_PIECEWISE_LINEAR_PLASTICITY/*MAT_124 and *MAT_PLASTICITY_WITH_DAMAGE/*MAT_081/*MAT_082.
- Fix incorrect damping when using *DAMPING_PART_STIFFNESS for *MAT_16/*MAT_PSEUDO_TENSOR and *EOS_TABULATED_COMPACTION.
- Fix incorrect computation of bulk modulus which caused complex sound speed error when using *EOS_TABULATED/EOS_09 with tabulated input.
- Fix moving part with *MAT_220 during dynamic relaxation when velocities are initialized.
- Fix convergence issue for *MAT_065/*MAT_MODIFIED_ZERILLI_ARMSTRONG for shells when VP = 1.
- Error terminate with message, KEY+1115, if _STOCHASTIC option is invoked for materials 10,15,24,81,98, or 123 but no

*DEFINE_STOCHASTIC_VARIATION or *DEFINE_HAZ_PROPERTIES keyword is present in the input file.

- Fix spurious error termination when using *DEFINE_HAZ_PROPERTIES with adpativity.
- Fix incorrect results or seg fault for *MAT_FU_CHANG_FOAM/*MAT_083 if KCON > 0.0 and TBID.ne.0.
- If SIGY = 0 and S = 0 in *MAT_DAMAGE_2/*MAT_105, set S = EPS1/200, where EPS1 is the first point of yield stress input or the first ordinate point of the LCSS curve.
- Allow *MAT_ENHANCED_COMPOSITE_DAMAGE/*MAT_054 failure mechanism to work together with *MAT_ADD_EROSION for shells.
- Fix incorrect erosion behavior if *MAT_ADD_EROSION is used with failure criteria defined for *MAT_123/*MAT_MODIFIED_PIECEWISE_LINEAR_PLASTICITY.
- Implement *MAT_FHWA_SOIL/*MAT_147 for 2D analysis, shell types 13, 14 and 15.
- Implement scaling of failure strain for *MAT_MODIFIED_PIECEWISE_LINEAR_PLASTICITY_STOCHASTIC/*MAT_123_STOCHASTIC for shells.
- Fix incorrect behavior for *MAT_LINEAR_ELASTIC_DISCRETE_BEAM/*MAT_066 when using damping with implicit (static) to explicit switching.
- Fixed *MAT_FABRIC/*MAT_034 with the negative unloading curve option. When searching for the intersection point of the load and unload curves, and extrapolation of one of the curves was needed to find the intersection point, the extrapolated stress was calculated incorrectly causing unpredictable behavior.
- Fixed fabric material forms 0 and 1 when used with a reference geometry. There were two problems, both occurring when there are mixed quad and triangular elements in the same block. A flaw in the strain calculation was leading to possible NaN forces in the elements. When a reference geometry was not used, the forces from triangular elements in mixed element blocks were 2 times too high.
- Added a new option for *MAT_SPOTWELD called FMODE. The FMODE option is available for DMGOPT = 10, 11, and 12. When the failure function is reached, and when FMODE > 0.0 and < 1.0, the value of FMODE will determine if a weld will fail immediately, or will have damage initiated. The failure function may include axial, shear, bending and torsion terms. If the sum of the squares of the shear and torsion terms divided by the sum of the square of all terms is greater than FMODE, then the weld will fail immediately. Otherwise, damage will be initiated.
- Enabled OPT = -1 on *MAT_SPOTWELD for brick elements which had not worked previously. Also, fixed TRUE_T when used with brick element forms 0, 1, and -1.

- Fixed spotwelds with DMGOPT = 12 by removing warning STR+1327 which made it impossible to set a small value of RS without triggering this warning, or without setting EFAIL smaller. Setting EFAIL small however could lead to damage initiation by plastic strain when the user wanted only initiation by the failure function.
- If DMGOPT = 10, 11, or 12 and EFAIL = 0, on *MAT_SPOTWELD, damage will now initiate only by the failure function. If EFAIL > 0, then damage will initiate be either then failure function or when plastic strain exceeds EFAIL. Prior to this version, damage could initiate when plastic strain exceeds zero if the user set EFAIL = 0. This behavior is still true for DMGOPT = 0, 1, or 2, but no longer for DMGOPT = 10, 11, or 12.
- Allow solid spot welds and solid spot weld assemblies to have up to 300 points in the running average that is used to smooth the failure function. In other words, up to NF = 300 is possible.
- Fixed a problem with brick spot weld assemblies when OPT = 0 failure is used without defining any weld resultant values. Welds were being immediately deleted.
- Added new PID option for *DEFINE_SPOTWELD_FAILURE (applies to *MAT_SPOTWELD, OPT = 10). Changes the Card 3 input for static strength values to use part set ID’s rather than material ID’s.
- Modified shell *MAT_214/*MAT_DRY_FABRIC to calculate fiber strains based on the current distance between the points where the fibers intersect with the element edges. Previously, they were calculated from the rate-of-deformation, but this was not as accurate as the new total strain measure.
- Fixed unit scaling for GAMAB1 and GAMAB2 on *MAT_DRY_FABRIC. We were incorrectly transforming them as stress.
- Reworked the plastic stress update in *MAT_225/*MAT_VISCOPLASITC_MIXED_HARDENING to prevent a divide by zero.
- Enabled *MAT_ADD_EROSION to be used with beams that have user defined integration. Memory allocation was fixed to prevent memory errors.
- Fixed *MAT_106 when used with tshell form 5 or 6. The elastic constants used in the assumed strain field were not reasonable.
- Fix issue that could have led to problems using *MAT_054 (or *MAT_058 or *MAT_158) in combination with TFAIL/TSIZE.gt.0.0 and damping.
- *MAT_054 - *MAT_ENHANCED_COMPOSITE_DAMAGE:

    - Add possibility to use failure criterion in *MAT_054 for solids in a transversal isotropic manner. It is assumed that the material 1-direction is the main axis and that the behavior in the 2-3 plane is isotropic. This feature is invoked by setting TI = 1 in *MAT_054.

- *MAT_058 - *MAT_LAMINATED_COMPOSITE_FABRIC:

    - Bugfix for shear stiffness behavior in *MAT_058 when using a table definition for GAB and only providing stress-strain-curves for positive shear.
    - Bugfix for strain-rate dependent stiffness behavior in *MAT_058 when using a table definition for EA, EB or GAB under compressive loading.
    - Add default values for strengths (XT,XC,YT,YC,SC) 1.e+16 for *MAT_058. If no values for the strengths were defined, unpredictable things could have happened.

- *MAT_138 - *MAT_COHESIVE_MIXED_MODE:

    - Store total mixed-mode and normal separation (delta_II & delta_I) on history variables 1&2 for *MAT_COHESIVE_MIXED_MODE (*MAT_138). This is only for post-processing and should not lead to any changes in the results.

- *MAT_157 - *MAT_ANISOTROPIC_ELASTIC_PLASTIC:

    - Add Tsai-Hill failure criterion (EXTRA = 2).
    - Allow strain-rate dependent strength values (XT,XC,YT,YC,ZT,ZC,SXY,SYZ,SZX) using *DEFINE_CURVE. This is available for Tsai-Wu (EXTRA = 1) and Tsai-Hill.
    - Fixed bug in using *MAT_157 with IHIS.gt.0 for shells. Thickness strain update d3 was not correct and plasticity algorithm may have failed.
    - Add additional option to IHIS in *MAT_157 for SHELLs.
    - Now also the strength values (XT,XC,YT,YC,SXY) may be initialized via *INITIAL_STRESS_SHELL. See variable IHIS and remarks in the User's Manual for details of initializing various blocks of material parameters.

- *MAT_215 - *MAT_4A_MICROMEC:

    - Add new material *MAT_215 that is a micromechanical material model that distinguishes between a fiber/inclusion and a matrix material. The material is intended for anisotropic composite materials, especially for short (SFRT) and long fiber thermoplastics (LFRT). This model is available for shells, tshells and solids.

- *MAT_225 - *MAT_VISCOPLASTIC_MIXED_HARDENING:

    - Fixed bug in *MAT_225 (*MAT_VISCOPLASTIC_MIXED_HARDENING) when using a table for LCSS together with kinematic hardening.

- *MAT_261 - *MAT_LAMINATED_FRACTURE_DAIMLER_PINHO:
  *MAT_262 - *MAT_LAMINATED_FRACTURE_DAIMLER_CAMANHO:

  - Allow table input for fracture toughness values for mats 261/262.Table represents fracture toughness vs. element length vs. strain rate (shells, tshells, solids)
  - Fixed bug in mats 261/262 when using *DAMPING_PART_STIFFNESS together with RYLEN = 2 in *CONTROL_ENERGY.
  - Correct shear failure behavior in *MAT_262. This will most probably have no effect to any real application, but could be seen in very special 1-element tests.

- Changed storage of history variables for *MAT_249 (*MAT_REINFORCED_THERMOPLASTIC). A new variable POSTV controls which variables are written and at what history variable location in d3plot.
- *MAT_254 (*MAT_GENERALIZED_PHASE_CHANGE) can now be used with shell elements and thermal thick shells.
- Added flag 'EZDEF' to *MAT_249_UDFIBER. In this case the last row of the deformation gradient is replaced by 0-0-1.
- Add opt. damage limitation curve/table LCDLIM for *MAT_ADD_GENERALIZED_DAMAGE.
- Add pre-defined damage tensors option PDDT to *MAT_ADD_GENERALIZED_DAMAGE.
- *MAT_ADD_GENERALIZED_DAMAGE now works for solid elements (only shells in R9).
- Add optional failure criterion FFCAP to *MAT_100 with OPT = -1 or 0.
- Enable *MAT_ADD_COHESIVE to be used in implicit analysis.
- Add alternative version of *MAT_280 invoked by new flag on 1st card. It is a physically based damage model with 4 new parameters.
- Enable *DEFINE_CONNECTION_PROPERTIES' option PROPRUL>=2 to be used with spotweld clusters, i.e. not only 1 hex element but several (via *DEFINE_HEX_SPOTWELD_ASSEMBLY or RPBHX > 1 on *CONTROL_SPOTWELD_BEAM).
- Enable *MAT_ADD_EROSION to be safely used with material models that have more than 119 history variables, for now the new limit is 169 (e.g. necessary for *MAT_157 with IHIS = 7).
- Add Tsai-Wu failure criterion to *MAT_157 for solid and shell elements invoked by EXTRA = 1 on card 6 and corresponding parameters on cards 8 and 9.
- Add viscoelastic option to *MAT_187 (SAMP-1). Rate dependent Young's modulus and associated settings can be defined on new optional card 5.
- Add new option IRNG for *DEFINE_STOCHASTIC_VARIATION to govern random number generation (deterministic or true random).

- Add option to define element size dependent parameters EN and SN for *MAT_120 and *MAT_120_JC by setting them to negative values (curves).
- Minor improvements for *MAT_252: Optional output of damage initiation information and more post-processing history variables.
- If the first abscissa value of *MAT_224's failure strain curve LCG is negative, it is assumed that all abscissa values are natural logarithms of a strain rate.
- Put *MAT_100_DA's "failure function" value to history variable 18.
- Add optional in-plane failure strain to *MAT_169 (ARUP_ADHESIVE): new input parameter FSIP.
- *MAT_USER_DEFINED_MATERIAL_MODELS now provides a few more variables for cohesive elements, i.e. additional arguments in subroutines umatXXc: temperature, element size, implicit rejection flag, integration point identifier, and total number of integration points.
- A modified version of the 3-parameter Barlat model (*MAT_036) is introduced as *MAT_EXTENDED_3-PARAMETER_BARLAT. In this model, hardening in 00, 45, 90, biaxial and shear can be specified as load curves. Furthermore, r-values in 00, 45, 90, biaxial and shear can be specified in terms of load curves vs plastic strain or constants. This is an extension of hardening law 7 of the original 3-parameter Barlat model.
- Improve implicit version of *MAT_098/*MAT_SIMPLIFIED_JOHNSON_COOK.
- *MAT_181/*MAT_SIMPLIFIED_RUBBER/FOAM is now supported for 2D implicit simulations.
- Fixed issue in which *MAT_WINFRITH_CONCRETE wrote d3crack data too frequently.
- *EOS_JWL now has an AFTERBURN option. This adds afterburn energy to the EOS, where the energy can be added at a constant or linear rate, or can be added according to Miller's extension.
-
- *MAT_084 (*MAT_WINFRITH_CONCRETE) with predefined units (CONM < 0) is now transformed correctly with *INCLUDE_TRANSFORM.
- User-defined materials for Hughes-Liu beams can now be used with implicit analysis by defining the appropriate tangent modulus in the supplied routine urtanb.
- User-defined cohesive materials can now be used with implicit analysis by defining the appropriate tangent stiffness.
- *MODULE for user-defined materials and other user-defined capabilities:

    - A new command line option "module = filename" is added to load one module file without changing the input deck. It provides back compatibility to input deck without the MODULE keywords.
    - The system paths defined in LD_LIBRARY_PATH are also included for searching module files for those filenames start with "+".

- Add shell implementation to *MAT_277 (*MAT_ADHESIVE_CURING_VISCOELASTIC).
- Add *MAT_278 for carbon fiber prepreg compression forming simulation. This material model is available for both solid and shell formulations.
- Add *MAT_293 non-orthogonal material model for carbon fiber prepreg forming simulation. This material model is only available for shell formulations.
- *MAT_260A:
  - Extend *MAT_260A to include solid elements.
  - Add a new option XUE for Xue's fracture criteria/theory for *MAT_M260A (solid elements only).
- *MAT_260B:
  - Set default values for P's and G's in *MAT_260B.
  - Add a length scale to the fracture limit. The fracture limit strongly depends on the length scale in the measurement.
  - Add a new fracture criterion to *MAT_260B (Xue and Wierzbicki, Int. J. solids and Structures 46 (2009) 1423-1435). When the option XUE is activated, an additional card is needed, for example:

    ```
    $    ef0     plim       q    gama      m
         0.70    925.7  0.970     0.296    2.04
    ```
- *MAT_037:
  - Improve *MAT_037 with negative R value in implicit calculation. The modification will allow the implicit method stress calculation to be more accurate.
  - Add a new option NLP2 to calculate formability index in *MAT_037. The previous method (option NLP_FAILURE) was based on the effective strain method, which assumes that necking happens at one instant. In fact, it might happen over a longer process. The new method calculates the damage accumulation.
- Add *MAT_165B (*MAT_PLASTIC_NONLINEAR_KINEMATIC_B) for shells and solids.

- **MPP**
  - Fix the report of decomp balance (shown as "Normalized element costs assigned during decomposition" in the d3hsp file), which was broken in r109760
  - MPP decomposition has not been properly balanced since r112652 due to a bug in that revision

- Fix MPP SYNC error due to inconsistent summation in *CONTACT_SLIDING_ONLY_PENALTY.
- Allow real values as the scale multipliers for "memory=" on the command line. For example, "memory = 2.5G memory2 = 1.1G" and the like.
- MPP: fix support for nlq setting in *CONTROL_SOLUTION which was not being honored on processors other than 0.
- Significant improvements in MPP groupable routines for FORMING contact.
- MPP: increase contact release distance for SINGLE_SURFACE contacts in the case of a node coming into contact with a solid element. The previous interpretation was releasing when the contact penetration was 0.5*solid thickness, but now when the node passes below the solid surface by 0.5*solid thickness (which is different by the half thickness of the slave material, in the case of a shell slave node).
- MPP: fix for viscous damping in automatic tiebreak contact.
- Implement new bucket sort based extent testing for MPP single surface contact.
- Added MPP support for *CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_LUBRICATION.
- Fixed *CONTROL_MPP_PFILE so that it honors ID offsets from *INCLUDE_TRANSFORM for parts, part sets, and contact IDs referenced in "decomp { region {" specifications.
- Furthermore, such a region can contain a "local" designation, in which case the decomposition of that region will be done in the coordinate system local to the include file, not the global system. For example:
- *CONTROL_MPP_PFILE decomp { region { partset 12 local c2r 30 0 -30 0 1 0 1 0 0 } } would apply the c2r transformation in the coordinate system of the include file, which wasn't previously possible. The local option can be useful even if there are no such transformations, as the "cubes" that the decomposition uses will be oriented in the coordinate system of the include file, not the global system.
- Furthermore, the following decomposition related keywords now have a _LOCAL option, which has the same effect:

    - *CONTROL_MPP_DECOMPOSITION_PARTS_DISTRIBUTE_LOCAL
    - *CONTROL_MPP_DECOMPOSITION_PARTSET_DISTRIBUTE_LOCAL
    - *CONTROL_MPP_DECOMPOSITION_ARRANGE_PARTS_LOCAL
    - *CONTROL_MPP_DECOMPOSITION_CONTACT_DISTRIBUTE_LOCAL

- MPP job performance profiles are output to both .csv and .xy files.

- **OUTPUT**

- Fix for writing d3plot file when individual output states exceed 8GB in single precision
- Added new option *INTERFACE_SPRINGBACK_EXCLUDE to exclude selected portions from the generated dynain file.
- Add a new option to *INTERFACE_COMPONENT_FILE to output only 3 degrees of freedom to the file even if the current model has 6.
- Fix missing plastic strain tensors in d3plot when STRFLG in *DATABASE_EXTENT_BINARY is set and INTSTRN = 1 in *INTERFACE_SPRINGBACK.
- Fix no output to bndout when run with q = remap even though the keyword *DATABASE_BNDOUT was present in the remap run but was not present in the initial run.
- Fix d3plot output frequency which was different from the dt specified in *DATABASE_BINARY_D3PLOT when *CONTACT_AUTO_MOVE is used.
- Fix stress output to elout for solid elements which was in the global coordinates instead of local coordinates when CMPFLG = 1 in *DATABASE_EXTENT_BINARY and OPTION1 > 0 in *DATABASE_ELOUT.
- Fix incorrect mass properties for solids in ssstat file when using *DATABASE_SSSTAT_MASS_PROPERTY.
- Fix seg fault during writing of dynain file if INSTRN = 1 in *INTERFACE_SPRINGBACK and STRFLG.ne.0 in *DATABASE_EXTENT_BINARY and the *DATABASE_EXTENT_BINARY comes after *INTERFACE_SPRINGBACK.
- If HYDRO is nonzero in *DATABASE_EXTENT_BINARY, LS-PrePost will now combine the solid and shell internal energy densities when fringing 'Internal Energy Density' in the Misc menu.
- By putting SIGFLG/EPSFLG = 3 in *DATABASE_EXTENT_BINARY, the stresses and plastic trains are excluded not only for shell elements but also for solids. This applies to d3plot and d3eigv.
- Added new file option *DATABASE_BINARY_INTFOR_FILE to define interface file name.
- Fixed legend in bndout in the case of multiple *BOUNDARY_PRESCRIBED_MOTION_SET_ID.
- Fix corrupt d3part data caused by DECOMP = 4 in *DATABASE_EXTENT_BINARY.
- Fixed the legend of ssstat in binout.
- Added *DATABASE_EXTENT_SSSTAT_ID. The subsystem id will be included in the ASCII ssstat file.
- Fixed bug in stbout (seatbelt output) if NEWLEG = 0 in *CONTROL_OUTPUT.
- Fixed bug in which DECOMP = 2 corrupted d3part.
- Fixed d3plot bug if dynamic relaxation was activated in the input deck.

- Added another digit for coordinates in *NODE in dynain, e.g., what was written as 0.999266236E+00 is now written as 9.992662368E+00.
- Added *DATABASE_EXTENT_BINARY_COMP for alternative (simpler) control of output to d3plot and d3eigv.

    - Output control flags: 0-no 1-yes
    - IGLB : Global data
    - IXYZ : Current coordinate
    - IVEL : Velocity
    - IACC : Acceleration
    - ISTRS: 6 stress data + plastic strain
    - ISTRA: 6 strain data
    - ISED : Strain energy density

  This command can be used in combination with regular *DATABASE_EXTENT_BINARY but will disable most of the options in the latter, including output of extra history variables.
- Bugfix: *DATABASE_TRACER without the optional NID parameter was read incorrectly when used with *INCLUDE_TRANSFORM, but is now fixed
- Fixed incomplete output from Windows version of LS-DYNA. This affected demtrh (*DATABASE_TRACER_DE) and curvout (*DATABASE_CURVOUT).

- **Restarts**

  - Enable definition of sensors in full restarts.
  - For a small restart in MPP, the value of "memory=" (M1) needed for each processor is stored in the dump files. This is the minimum requirement to read back the model info. If the value of "memory2=" (M2) is specified on the command line, the code will take the maximum of M1 and M2.
  - Fix input error during structured input when using *INITIAL_VELOCITY_GENERATION and *CHANGE_VELOCITY_GENERATION together in a full deck restart.
  - Fix incorrect full deck restart analysis if initial run was implicit and the full deck restart run is explicit. This affects MPP only.
  - Fix insufficient tying of nodes when doing full deck restart and the contact is newly added to the restart involving newly added parts. This applies to SMP contact only.
  - Fix incorrect velocity initialization for SMP full deck restart when using *INITIAL_VELOCITY_GENERATION and *INITIAL_VELOCITY_GENERATION_START_TIME.
  - Fix incorrect initialization of velocities for SMP full deck restart when using *CHANGE_VELOCITY_OPTION & *INITIAL_VELOCITY_OPTION.

Velocities of existing parts defined by *STRESS_INITIALIZATION should not be zeroed.
  - Fix *CHANGE_CURVE_DEFINITION for curve specifying d3plot output.
  - Fixed bug in full deck restart if the new mesh has different part numbers.

- ***SENSOR**

  - Fix a bug regarding *SENSOR_JOINT_FORCE for *CONSTRAINED_JOINT_STIFFNESS, that was triggered when the force refers to a local coordinate system.
  - Add the option of "ELESET" to *SENSOR_CONTROL to erode elements.
  - Add the option of NFAILE to *SENSOR_DEFINE_MISC to track number of eroded elements.
  - Fix a bug that was triggered when using a sensor to control spotwelds. The bug was triggered when the spotweld-connected nodal pairs happen to belong to more than 1 core (MPP only).
  - Add FAIL option to *SENSOR_DEFINE_ELEMENT to track the failure of element(s).
  - Fix a bug related to *SENSOR_DEFINE_FUNCTION when there are more than 10 sensor definitions.
  - Effect of TIMEOFF in *SENSOR_CONTROL is implemented for TYPE = PRESC-ORI.
  - *SENSOR_CONTROL can be used to control *BOUNDARY_PRESCRIBED_ORIENTATION_RIGID.
  - Add optional filter id to SENSORD of *DEFINE_CURVE_FUNCTION.
  - Enable *CONSTRAINED_JOINT_..._LOCAL to be monitored by *SENSOR_DEFINE_FORCE.
  - Allow moments in SPCFORC and BNDOUT to be tracked by *SENSOR_DEFINE_FORCE.
  - Fix *SENSOR_CONTROL using TYPE="PRESC-MOT" which was not switching at all.

- **SPG (Smooth Particle Galerkin)**

  - MPP is ready in 3D SPG fluid particle stabilization (ITB = 1 & 2 in *SECTION_SOLID_SPG).
  - Added one SPG control parameter (itb = 2) for semi-brittle fracture analysis. In comparison to itb = 0 or itb = 1, itb = 2 is more efficient in modeling the fragmentation and debris in semi-brittle fracture analysis such as impact and penetration in concrete materials.
  - Fixed a bug related to E.O.S. in SPG.
  - Removed some temporary memory allocations to improve efficiency.
  - Changed the sequence of SPG initialization so that all state variables are properly initialized.

  - Subroutines were developed for SPG failure analysis with thermal effects. Both explicit and implicit (diagonal scaled conjugate gradient iterative only) SPG thermal solvers are available in SMP version only. However, thermal effect is applied only on material properties, which means thermal induced deformation (i.e., thermal strain or thermal expansion) is not currently included.
  - Modified *MAT_072R3 for SPG method in concrete applications.
  - Fixed a bug for SPG method in using continuum damage mechanics. (IDAM = 0).
  - Added the "fluid particle algorithm" (itb = 1) to SPG method. This algorithm is implemented in R10.0 as an alternative to the (itb = 0) option in previous version to enhance the numerical stability for SPG method. Users are recommended to use this new option for their ductile failure analysis.

- **SPH (Smooth Particle Hydrodynamics)**

  - Add ITHK flag in *CONTROL_SPH, card 3. If flag is set to 1, the volume of the SPH particles is used to estimate a node thickness to be employed by contacts.

    - Affects *AUTOMATIC_NODES_TO_SURFACE and *CONTACT_2D_NODE_TO_SOLID.
    - The thickness calculated by ITHK = 1 is used only if SST or OFFD are set to zero in the contact cards definitions.

  - Add SOFT = 1 option to *CONTACT_2D_NODE_TO_SOLID. This should help obtain reasonable contact forces in axisymmetric simulations. Default penalty PEN is 0.1 when SOFT = 1.
  - Implemented non-reflecting boundary conditions for SPH using a new keyword *BOUNDARY_SPH_NON_REFLECTING.
  - Bug fix for renormalized SPH formulations with symmetry planes. The renormalization was slightly incorrect in the vicinity of symmetry planes.
  - Density smoothing in SPH formulations 15 and 16 is now material sensitive. The smoothing only occurs over neighbors of the same material.
  - Resolved an MPP bug in SPH total Lagrangian formulations (FORM = 7/8) which was causing strain concentrations at the interfaces between CPU zones.
  - SPH total Lagrangian (FORM = 7/8) in SMP was pretty much serial, hence much slower than forms 0 or 1. SPH with FORM 7 and 8 now scales properly.
  - Added support for FORMs 0/1 in axisymmetric. Until now, renormalization was always active (equivalent to FORM = 1) which can be problematic for very large deformations or material fragmentation.
  - Improved tracer particles output for SPH: Use normalized kernel function for interpolation between particles.

- Implemented enhanced fluid flow formulations (FORMs 15/16) with pressure smoothing.
- Recode SPH neighborhood search algorithm to reduce the memory requirement and produce consistent results from MPP and HYBRID code.
- *DEFINE_ADAPTIVE_SOLID_TO_SPH now reports both active and inactive adaptive SPH particles in the fragment file sldsph_frag. This file gives a report of nodal mass, coordinates, and velocities.
- MPP now supports:
    - SPH type 3 inflow
    - Multiple *BOUNDARY_SPH_FLOW
    - Bulk viscosity option for SPH
- Sort SPH by part and then node ID to ensure consistent results while changing order of input files.
- *DEFINE_SPH_TO_SPH_COUPLING:
    - Corrected the SPH sphere radius (half of the particles distance) for node to node contact detecting algorithm.
    - Updated masses for SPH node to node coupling with damping contact force option.
    - Added a new option (Soft = 1) for SPH to SPH coupling: contact stiffness comes from particles masses and time step for softer contact.
- *DEFINE_ADAPTIVE_SOLID_TO_SPH:
    - Updated temperature transfer (from solid elements to SPH particles) when converting solid elements into SPH particles with ICPL = 1, IOPT = 0.
    - Bug fixed when part ID for newly generated SPH particles is smaller than the original SPH part ID.
    - Introduced a new pure thermal coupling between SPH part and solid parts with ICPL = 3 and IOPT = 0 option (no structural coupling provided).
    - Added a thermal coupling conductivity parameter CPCD. Applies to ICPL = 3 option.
    - Normalized the nodal temperatures for the corner SPH particles with ICPL = 3 and IOPT = 0 option (MPP only).
    - Extended ICPL = 3 and IOPT = 0 option to Lagrangian formulation (form = 7, 8).
- *BOUNDARY_SPH_SYMMETRY_PLANE:
    - Added in an error message if TAIL and HEAD points are at the same location.

- *CONTACT_2D_NODE_TO_SOLID:
    - Added a variable OFFD to specify contact offset.
- Added a new option IEROD = 2 in *CONTROL_SPH in which SPH particles that satisfy a failure criterion are totally eactivated and removed from domain interpolation. This is in contrast to IEROD = 1 option in which particles are partially deactivated and only stress states are set to zero.
- Added *MAT_SPH_VISCOUS (*MAT_SPH_01) for fluid-like material behavior with constant or variable viscosity. Includes a Cross viscosity model.
- Output strain rates for SPH particles to d3plot, d3thdt, and sphout file.
- Added support of *MAT_ADD_EROSION, including GISSMO and DIEM damage, for SPH particles.
- Echo failed SPH particles into d3hsp and messag file.
- *DEFINE_SPH_INJECTION:
    - Changed the method of generating SPH particles. SPH particles will be generated based on the injection volume (injection area*injection velocity*dt)*density from the material model, resulting in more consistent particle masses and particle distribution.
    - Offset injecting distance inside each cycle so that outlet distance will be consistent for different outlet SPH layers.
    - Corrected mass output in d3hsp.

- **Thermal Solver**
    - Modify the thermal solver routines so they return instead of terminating, so that *CASE works properly.
    - *MAT_THERMAL_USER_DEFINED: Fixed bug in element numbering for IHVE = 1.
    - Accept load curve input for dtmin, dtmax and dtemp in *CONTROL_THERMAL_TIMESTEP. As usual if a negative integer number is given its absolute value refers to the load curve id.
    - The temperature results for the virtual nodes of thermal thick shells are now accounted for in *LOAD_THERMAL_D3PLOT. For the mechanics-only simulation thermal thick shells have to be activated.
    - New contact type for thermal solver that models heat transfer from and to a shell edge onto a surface (*CONTACT_..._THERMAL with ALGO > 1):
        - Shells have to be thermal thick shells.
        - Shells are on the slave side.
        - So far only implemented for SMP.
        - Includes support for quads and triangles.

- New keyword *BOUNDARY_THERMAL_WELD_TRAJECTORY for welding of solid or shell structures.
  - Keyword defines the movement of a heat source on a nodal path (*SET_NODE).
  - Orientation given either by vector or with a second node set.
  - Works for coupled and thermal only analyses.
  - Allows for thermal dumping.
  - Different equivalent heat source descriptions available.
  - Can also be applied to tshells and composite shells.
  - Weld torch motion can be defined relative to the weld trajectory.
- Solid element formulation 18 now supports thermal analysis.
- Thermal solver now supports the H8TOH20 option of *ELEMENT_SOLID. This includes support of *INITIAL_TEMPERATURE condition for the extra 12 nodes generated by H8TOH20.
- Thermal solver now supports the H8TOH27 option of *ELEMENT_SOLID.
- Explicit Thermal Solver
  - *CONTROL_EXPLICIT_THERMAL_SOLVER: Implement an explicit thermal solver and adapt it to support multi-material ALE cases.
  - *CONTROL_EXPLICIT_THERMAL_PROPERTIES: Enter thermal properties for the explicit thermal solver.
  - *CONTROL_EXPLICIT_THERMAL_CONTACT: Implement a thermal contact for the explicit thermal solver.
  - *CONTROL_EXPLICIT_THERMAL_ALE_COUPLING: Implement a thermal coupling between ALE and Lagrangian structures for use by the explicit thermal solver.
  - *CONSTRAINED_LAGRANGE_IN_SOLID_EDGE: For the explicit thermal ALE coupling, allow the heat transfer through the shell edges if _EDGE is added to *CONSTRAINED_LAGRANGE_IN_SOLID.
  - *CONSTRAINED_LAGRANGE_IN_SOLID: For the explicit thermal solver, add work due to friction to the enthalpies of ALE and structure elements coupled with *CONSTRAINED_LAGRANGE_IN_SOLID (CTYPE = 4).
  - *CONTROL_EXPLICIT_THERMAL_INITIAL: Initialize the temperatures for the explicit thermal solver.
  - *CONTROL_EXPLICIT_THERMAL_BOUNDARY: Control boundary temperatures for the explicit thermal solver.
  - *CONTROL_EXPLICIT_THERMAL_OUTPUT: Output the temperatures at element centers for the explicit thermal solver.
  - *DATABASE_PROFILE: For the explicit thermal solver, output temperature profiles.

- **Miscellaneous**

- *INITIAL_LAG_MAPPING: Implement a 3D to 3D lagrangian mapping and map the nodal temperatures.
- *CONTROL_REFINE_SHELL and *CONTROL_REFINE_SOLID: Add a parameter MASTERSET to call a set of nodes to flag element edges along which new child nodes are constrained.
- *BOUNDARY_PRESCRIBED_MOTION_SET_SEGMENT: Add DOF = 12 to apply velocities in local coordinate systems attached to segments.
- Fixed bug occurring when a part has non-zero *DAMPING_PART_STIFFNESS, AND is defined using *PART_COMPOSITE, AND the MIDs referenced by the different integration points have different material types. Symptoms could include many types of unexpected behavior or error termination, but in other cases it could be harmless.
- *DAMPING_FREQUENCY_RANGE (including _DEFORM option): Improved internal calculation of damping constants such that the level of damping more accurately matches the user-input value across the whole of the frequency range FLOW to FHIGH. As an example, for CDAMP = 0.01, FLOW = 1 Hz and FHIGH = 30 Hz, the actual damping achieved by the previous algorithm varied between 0.008 and 0.012 (different values at different frequencies between FLOW and FHIGH), i.e. there were errors of up to 20% of the target CDAMP. With the new algorithm, the errors are reduced to 1% of the target CDAMP. This change will lead to some small differences in results compared to previous versions of LS-DYNA. Users wishing to retain the old method for compatibility with previous work can do this by setting IFLG (7th field on Card 1) to 1.
- Fixed bug that could cause unpredictable symptoms if Nodal Rigid Bodies were included in the Part Set referenced by *DAMPING_FREQUENCY_RANGE or *DAMPING_FREQUENCY_RANGE_DEFORM. Now, the _DEFORM option silently ignores NRBs in the Part Set while *DAMPING_FREQUENCY_RANGE (non _DEFORM option) damps them.
- Fixed bug in *PART_COMPOSITE: if a layer had a very small thickness defined, such as 1E-9 times the total thickness, that layer would be assigned a weighting factor of 1 (it should be close to zero).
- Fix errors in implementation of *DEFINE_FILTER type CHAIN.
- Fix for *INTERFACE_LINKING_LOCAL when LCID is used. During keyword processing, the LCID value was not properly converted to internal numbering.
- Switch coordinates in keyword reader to double precision.
- Change "Warning" to "Error" for multiply defined materials, boxes, coordinate systems, vectors, and orientation vectors. The check for duplicate section IDs now includes the element type and remains a warning for now, because SPH is still detected as a SOLID. Once that is straightened out, this should be made an error.

- Add "TIMESTEP" as a variable for *DEFINE_CURVE_FUNCTION. This variable holds the current simulation time step.
- Fix a bug for the case of CODE = 5 in *DEFORMABLE_TO_RIGID_AUTOMATIC. (Fields 3 to 8 are now ignored.)
- Issue error message and terminate the simulation when illegal ACTION is used for *DEFINE_TRANSFORM.
- Add option of POS6N for *DEFINE_TRANSFORM to define transformation with 3 reference nodes and 3 target nodes.
- Fix a bug that can occur when adaptive elements are defined in a file included by *INCLUDE_TRANSFORM.
- Merge *DEFORMABLE_TO_RIGID_AUTOMATIC cards if they use the same switch time. This dependency of results on the order of the cards and also gives better performance.
- If *SET_PART_OPTION is used, a "group_file" will be created which can be read into LS-Prepost (Model > Groups > Load) for easy visualization of part sets.
- Forces on *RIGIDWALL_GEOMETRIC_CYLINDER can now be subdivided into sections for output to rwforc. This gives a better idea of the force distribution along the length of the cylinder. See the variable NSEGS.
- Added the keywords *DEFINE_PRESSURE_TUBE and *DATABASE_PRTUBE for simulating pressure tubes in pedestrian crash.
- Fix non-effective OPTIONs DBOX, DVOL, DSOLID, DSHELL, DTSHELL, DSEG for *SET_SEGMENT_GENERAL to delete segments.
- Fix incorrect transformation of valdmp in *DAMPING_GLOBAL with *INCLUDE_TRANSFORM.
- Make *SET_NODE_COLLECT work together with *NODE_SET_MERGE.
- Fixed bug in adaptivity for *INCLUDE_TRANSFORM if jobid is used.
- Bugfix: *INTERFACE_SSI with blank optional card is now read in correctly.

MATERIAL MODELS

Some of the material models presently implemented are:

- elastic,
- orthotropic elastic,
- kinematic/isotropic plasticity [Krieg and Key 1976],
- thermoelastoplastic [Hallquist 1979],
- soil and crushable/non-crushable foam [Key 1974],
- linear viscoelastic [Key 1974],

- Blatz-Ko rubber [Key 1974],
- high explosive burn,
- hydrodynamic without deviatoric stresses,
- elastoplastic hydrodynamic,
- temperature dependent elastoplastic [Steinberg and Guinan 1978],
- isotropic elastoplastic,
- isotropic elastoplastic with failure,
- soil and crushable foam with failure,
- Johnson/Cook plasticity model [Johnson and Cook 1983],
- pseudo TENSOR geological model [Sackett 1987],
- elastoplastic with fracture,
- power law isotropic plasticity,
- strain rate dependent plasticity,
- rigid,
- thermal orthotropic,
- composite damage model [Chang and Chang 1987a 1987b],
- thermal orthotropic with 12 curves,
- piecewise linear isotropic plasticity,
- inviscid, two invariant geologic cap [Sandler and Rubin 1979, Simo et al, 1988a
- 1988b],
- orthotropic crushable model,
- Mooney-Rivlin rubber,
- resultant plasticity,
- force limited resultant formulation,
- closed form update shell plasticity,
- Frazer-Nash rubber model,
- laminated glass model,
- fabric,
- unified creep plasticity,
- temperature and rate dependent plasticity,
- elastic with viscosity,
- anisotropic plasticity,

- user defined,
- crushable cellular foams [Neilsen, Morgan, and Krieg 1987],
- urethane foam model with hysteresis,

and some more foam and rubber models, as well as many materials models for springs and dampers. The hydrodynamic material models determine only the deviatoric stresses. Pressure is determined by one of ten equations of state including:

- linear polynomial [Woodruff 1973],
- JWL high explosive [Dobratz 1981],
- Sack "Tuesday" high explosive [Woodruff 1973],
- Gruneisen [Woodruff 1973],
- ratio of polynomials [Woodruff 1973],
- linear polynomial with energy deposition,
- ignition and growth of reaction in HE [Lee and Tarver 1980, Cochran and Chan 1979],
- tabulated compaction,
- tabulated,
- TENSOR pore collapse [Burton et al. 1982].

The ignition and growth EOS was adapted from KOVEC [Woodruff 1973]; the other subroutines, programmed by the authors, are based in part on the cited references and are nearly 100 percent vectorized. The forms of the first five equations of state are also given in the KOVEC user's manual and are retained in this manual. The high explosive programmed burn model is described by Giroux [Simo et al. 1988].

The orthotropic elastic and the rubber material subroutines use Green-St. Venant strains to compute second Piola-Kirchhoff stresses, which transform to Cauchy stresses. The Jaumann stress rate formulation is used with all other materials with the exception of one plasticity model which uses the Green-Naghdi rate.

SPATIAL DISCRETIZATION

are presently available. Currently springs, dampers, beams, membranes, shells, bricks, thick shells and seatbelt elements are included.

The first shell element in DYNA3D was that of Hughes and Liu [Hughes and Liu 1981a, 1981b, 1981c], implemented as described in [Hallquist et al. 1985, Hallquist and Benson 1986]. This element [designated as HL] was selected from among a substantial body of shell element literature because the element formulation has several desirable qualities:

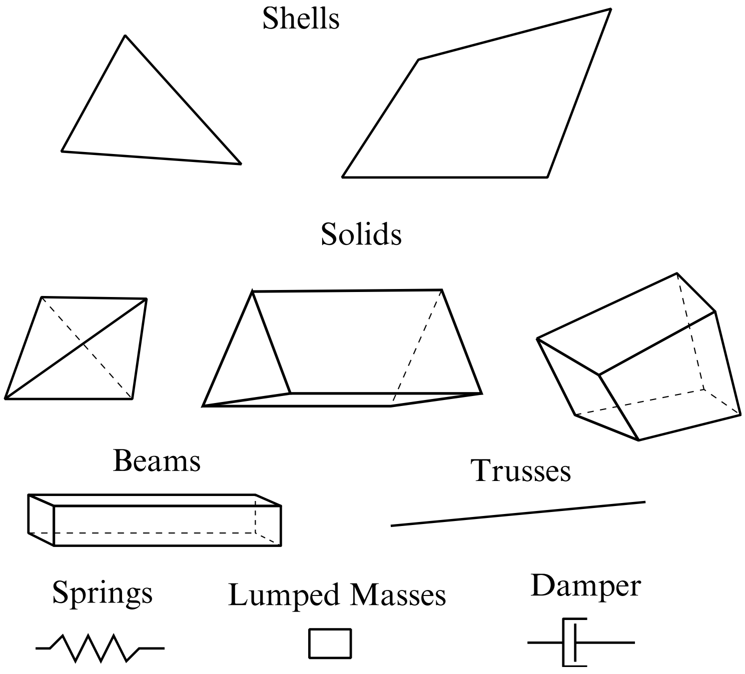


Figure 1-1. Elements in LS-DYNA. Three-dimensional plane stress constitutive subroutines are implemented for the shell elements which iteratively update the stress tensor such that the stress component normal to the shell midsurface is zero. An iterative update is necessary to accurately determine the normal strain component which is necessary to predict thinning. One constitutive evaluation is made for each integration point through the

- It is incrementally objective (rigid body rotations do not generate strains), allowing for the treatment of finite strains that occur in many practical applications.
- It is compatible with brick elements, because the element is based on a degenerated brick element formulation. This compatibility allows many of the efficient and effective techniques developed for the DYNA3D brick elements to be used with this shell element;
- It includes finite transverse shear strains;
- A through-the-thickness thinning option (see [Hughes and Carnoy 1981]) is also available.

All shells in our current LS-DYNA code must satisfy these desirable traits to at least some extent to be useful in metalforming and crash simulations.

The major disadvantage of the HL element turned out to be cost related and, for this reason, within a year of its implementation we looked at the Belytschko-Tsay [BT] shell [Belytschko and Tsay 1981, 1983, 1984] as a more cost effective, but possibly less accurate alternative. In the BT shell the geometry of the shell is assumed to be perfectly flat, the local coordinate system originates at the first node of the connectivity, and the co-rotational stress update does not use the costly Jaumann stress rotation. With these and other simplifications, a very cost effective shell was derived which today has become perhaps the most widely used shell elements in both metalforming and crash applications. Results generated by the BT shell usually compare favorably with those of the more costly HL shell. Triangular shell elements are implemented, based on work by Belytschko and co-workers [Belytschko and Marchertas 1974, Bazeley et al. 1965, Belytschko et al. 1984], and are frequently used since collapsed quadrilateral shell elements tend to lock and give very bad results. LS-DYNA automatically treats collapsed quadrilateral shell elements as C^0 triangular elements.

Since the Belytschko-Tsay element is based on a perfectly flat geometry, warpage is not considered. Although this generally poses no major difficulties and provides for an efficient element, incorrect results in the twisted beam problem and similar situations are obtained where the nodal points of the elements used in the discretization are not coplanar. The Hughes-Liu shell element considers non-planar geometries and gives good results on the twisted beam. The effect of neglecting warpage in a typical application cannot be predicted beforehand and may lead to less than accurate results, but the latter is only speculation and is difficult to verify in practice. Obviously, it would be better to use shells that consider warpage if the added costs are reasonable and if this unknown effect is eliminated. Another shell published by Belytschko, Wong, and Chiang [Belytschko, Wong, and Chiang 1989, 1992] proposes inexpensive modifications to include the warping stiffness in the Belytschko-Tsay shell. An improved transverse shear treatment also allows the element to pass the Kirchhoff patch test. This element is now available in LS-DYNA. Also, two fully integrated shell elements, based on the Hughes and Liu formulation, are available in LS-DYNA, but are rather expensive. A much faster fully integrated element which is essentially a fully integrated version of the Belytschko, Wong, and Chiang element, type 16, is a more recent addition and is recommended if fully integrated elements are needed due to its cost effectiveness.

Zero energy modes in the shell and solid elements are controlled by either an hourglass viscosity or stiffness. Eight node thick shell elements are implemented and have been found to perform well in many applications. All elements are nearly 100% vectorized. All element classes can be included as parts of a rigid body. The rigid body formulation is documented in [Benson and Hallquist 1986]. Rigid body point nodes, as well as concentrated masses, springs and dashpots can be added to this rigid body.

Membrane elements can be either defined directly as shell elements with a membrane formulation option or as shell elements with only one point for through thickness integration. The latter choice includes transverse shear stiffness and may be inappropriate. For airbag material a special fully integrated three and four node membrane element is available.

Two different beam types are available: a stress resultant beam and a beam with cross section integration at one point along the axis. The cross section integration allows for a more general definition of arbitrarily shaped cross sections taking into account material nonlinearities.

Spring and damper elements can be translational or rotational. Many behavior options can be defined, e.g., arbitrary nonlinear behavior including locking and separation.

Solid elements in LS-DYNA may be defined using from 4 to 8 nodes. The standard elements are based on linear shape functions and use one point integration and hourglass control. A selective-reduced integrated (called fully integrated) 8 node solid element is available for situations when the hourglass control fails. Also, two additional solid elements, a 4 noded tetrahedron and an 8 noded hexahedron, with nodal rotational degrees of freedom, are implemented based on the idea of Allman [1984] to replace the nodal midside translational degrees of freedom of the elements with quadratic shape functions by corresponding nodal rotations at the corner nodes. The latter elements, which do not need hourglass control, require many numerical operations compared to the hourglass controlled elements and should be used at places where the hourglass elements fail. However, it is well known that the elements using more than one point integration are more sensitive to large distortions than one point integrated elements.

The thick shell element is a shell element with only nodal translations for the eight nodes. The assumptions of shell theory are included in a non-standard fashion. It also uses hourglass control or selective-reduced integration. This element can be used in place of any four node shell element. It is favorably used for shell-brick transitions, as no additional constraint conditions are necessary. However, care has to be taken to know in which direction the shell assumptions are made; therefore, the numbering of the element is important.

Seatbelt elements can be separately defined to model seatbelt actions combined with dummy models. Separate definitions of seatbelts, which are one-dimensional elements, with accelerometers, sensors, pretensioners, retractors, and sliprings are possible. The actions of the various seatbelt definitions can also be arbitrarily combined.

CONTACT-IMPACT INTERFACES

The three-dimensional contact-impact algorithm was originally an extension of the NIKE2D [Hallquist 1979] two-dimensional algorithm. As currently implemented, one surface of the interface is identified as a master surface and the other as a slave. Each surface is defined by a set of three or four node quadrilateral segments, called master and slave segments, on which the nodes of the slave and master surfaces, respectively, must slide. In general, an input for the contact-impact algorithm requires that a list of master and slave segments be defined. For the single surface algorithm only the slave surface is defined and each node in the surface is checked each time step to ensure that it does not penetrate through the surface. Internal logic [Hallquist 1977, Hallquist et al. 1985] identifies a master segment for each slave node and a slave segment for each master node and updates this information every time step as the slave and master nodes slide along their respective surfaces. It must be noted that for general automatic definitions only parts/materials or three-dimensional boxes have to be given. Then the possible contacting outer surfaces are identified by the internal logic in LS-DYNA. More than 20 types of interfaces can presently be defined including:

- sliding only for fluid/structure or gas/structure interfaces
- tied
- sliding, impact, friction
- single surface contact
- discrete nodes impacting surface
- discrete nodes tied to surface
- shell edge tied to shell surface
- nodes spot welded to surface
- tiebreak interface
- one way treatment of sliding, impact, friction
- box/material limited automatic contact for shells
- automatic contact for shells (no additional input required)
- automatic single surface with beams and arbitrary orientations
- surface to surface eroding contact
- node to surface eroding contact
- single surface eroding contact
- surface to surface symmetric constraint method [Taylor and Flanagan 1989]
- node to surface constraint method [Taylor and Flanagan 1989]
- rigid body to rigid body contact with arbitrary force/deflection curve
- rigid nodes to rigid body contact with arbitrary force/deflection curve
- edge-to-edge
- draw beads

Interface friction can be used with most interface types. The tied and sliding only interface options are similar to the two-dimensional algorithm used in LS-DYNA2D [Hallquist 1976, 1978, 1980]. Unlike the general option, the tied treatments are not symmetric; therefore, the surface which is more coarsely zoned should be chosen as the master surface. When using the one-way slide surface with rigid materials, the rigid material should be chosen as the master surface.

For geometric contact entities, contact has to be separately defined. It must be noted that for the contact of a rigid body with a flexible body, either the sliding interface definitions as explained above or the geometric contact entity contact can be used.

Currently, the geometric contact entity definition is recommended for metalforming problems due to high accuracy and computational efficiency.

INTERFACE DEFINITIONS FOR COMPONENT ANALYSIS

Interface definitions for component analyses are used to define surfaces, nodal lines, or nodal points (*INTERFACE_COMPONENTS) for which the displacement and velocity time histories are saved at some user specified frequency (*CONTROL_OUTPUT). This data may then used to drive interfaces (*INTERFACE_LINKING) in subsequent analyses. This capability is especially useful for studying the detailed response of a small member in a large structure. For the first analysis, the member of interest need only be discretized sufficiently that the displacements and velocities on its boundaries are reasonably accurate. After the first analysis is completed, the member can be finely discretized and interfaces defined to correspond with the first analysis. Finally, the second analysis is performed to obtain highly detailed information in the local region of interest.

When starting the analysis, specify a name for the interface segment file using the Z = parameter on the LS-DYNA command line. When starting the second analysis, the name of the interface segment file (created in the first run) should be specified using the L = parameter on the LS-DYNA command line.

Following the above procedure, multiple levels of sub-modeling are easily accommodated. The interface file may contain a multitude of interface definitions so that a single run of a full model can provide enough interface data for many component analyses. The interface feature represents a powerful extension of LS-DYNA’s analysis capability.

PRECISION

The explicit time integration algorithms used in LS-DYNA are in general much less sensitive to machine precision than other finite element solution methods. Consequently, double precision is not generally required. The benefits of this are greatly improved utilization of memory and disk. When problems have been found we have usually been able to overcome them by reorganizing the algorithm or by converting to double precision locally in the subroutine where the problem occurs. Particularly sensitive problems (e.g. some buckling problems, which can be sensitive to small imperfections) may require the fully double precision version, which is available on all platforms. Very large problems requiring more than 2 billion words of memory will also need to be run in double precision, due to the array indexing limitation of single precision integers.

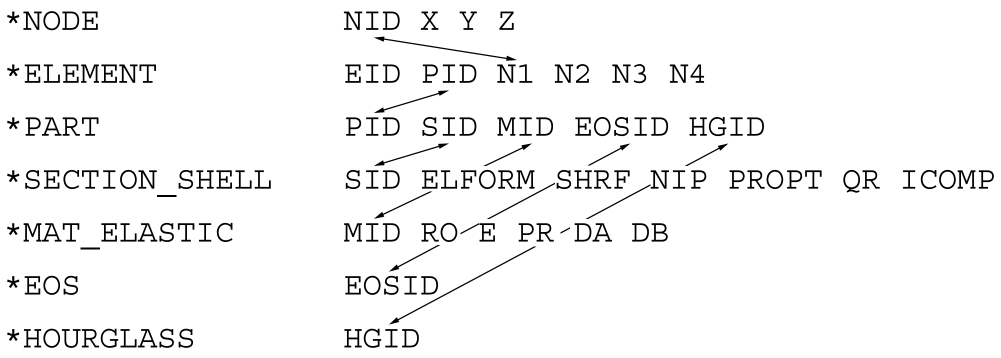
GETTING STARTED

DESCRIPTION OF KEYWORD INPUT

The keyword input provides a flexible and logically organized database that is simple to understand. Similar functions are grouped together under the same keyword. For example, under the keyword *ELEMENT are included solid, beam, shell elements, spring elements, discrete dampers, seat belts, and lumped masses. Many keywords have options that are identified as follows: *"OPTIONS"* and *"{OPTIONS}"*. The difference is that *"OPTIONS"* requires that one of the options must be selected to complete the keyword command. The option <BLANK> is included when {} are used to further indicate that these particular options are not necessary to complete the keyword.

LS-DYNA User's Manual is alphabetically organized in logical sections of input data. Each logical section relates to a particular input. There is a control section for resetting LS-DYNA defaults, a material section for defining constitutive constants, an equation-of-state section, an element section where element part identifiers and nodal connectivities are defined, a section for defining parts, and so on. Nearly all model data can be input in block form. For example, consider the following where two nodal points with their respective coordinates and shell elements with their part identity and nodal connectivity's are defined:

```
$define two nodes
$
*NODE
10101x y z
10201x y z
$    define two shell elements
$
*ELEMENT_SHELL
10201pidn1n2n3n4
10301pidn1n2n3n4
Alternatively, acceptable input could also be of the form:
$     define one node
$
*NODE
10101x y z
$    define one shell element
$
*ELEMENT_SHELL
10201pidn1n2n3n4
$
$     define one more node
$
*NODE
10201x y z
$ define one more shell element
$
*ELEMENT_SHELL
10301pidn1n2n3n4
```



```
*NODE              NID X Y Z
*ELEMENT           EID PID N1 N2 N3 N4
*PART              PID SID MID EOSID HGID
*SECTION_SHELL     SID ELFORM SHRF NIP PROPT QR ICOMP
*MAT_ELASTIC       MID RO E PR DA DB
*EOS               EOSID
*HOURGLASS         HGID
```

Figure 2-1. Organization of the keyword input.

A data block begins with a keyword followed by the data pertaining to the keyword. The next keyword encountered during the reading of the block data defines the end of the block and the beginning of a new block. A keyword must be left justified with the "*" contained in column one. A dollar sign "$" in column one precedes a comment and causes the input line to be ignored. Data blocks are not a requirement for LS-DYNA but they can be used to group nodes and elements for user convenience. Multiple blocks can be defined with each keyword if desired as shown above. It would be possible to put all nodal points definitions under one keyword *NODE, or to define one *NODE keyword prior to each node definition. The entire LS-DYNA input is order independent with the exception of the optional keyword, *END, which defines the end of input stream. Without the *END termination is assumed to occur when an end-of-file is encountered during the reading.

Figure 2-1 highlights how various entities relate to each other in LS-DYNA input. In this figure the data included for the keyword, *ELEMENT, is the element identifier, EID, the part identifier, PID, and the nodal points identifiers, the NID's, defining the *element connectivity*: N1, N2, N3, and N4. The nodal point identifiers are defined in the *NODE section where each NID should be defined just once. A part defined with the *PART keyword has a unique part identifier, PID, a section identifier, SID, a material or constitutive model identifier, MID, an equation of state identifier, EOSID, and the hourglass control identifier, HGID. The *SECTION keyword defines the section identifier, SID, where a section has an element formulation specified, a shear factor, SHRF, a numerical integration rule, NIP, among other parameters.

Constitutive constants are defined in the *MAT section where constitutive data is defined for all element types including solids, beams, shells, thick shells, seat belts, springs, and dampers. Equations of state, which are used only with certain *MAT materials for solid elements, are defined in the *EOS section. Since many elements in LS-DYNA use uniformly reduced numerical integration, zero energy deformation modes may develop. These modes are controlled numerically by either an artificial stiffness or viscosity which resists the formation of these undesirable modes. The hourglass control can optionally be user specified using the input in the *HOURGLASS section.

During the keyword input phase where data is read, only limited checking is performed on the data since the data must first be counted for the array allocations and then reordered. Considerably more checking is done during the second phase where the input data is printed out. Since LS-DYNA has retained the option of reading older non-keyword input files, we print out the data into the output file d3hsp (default name) as in previous versions of LS-DYNA. An attempt is made to complete the input phase before error terminating if errors are encountered in the input. Unfortunately, this is not always possible and the code may terminate with an error message. The user should always check either output file, d3hsp or messag, for the word "Error".

The input data following each keyword can be input in free format. In the case of free format input the data is separated by commas, i.e.,

```
*NODE
10101,x ,y ,z
10201,x ,y ,z

*ELEMENT_SHELL
10201,pid,n1,n2,n3,n4
10301,pid,n1,n2,n3,n4
```

When using commas, the formats *must not* be violated. An I8 integer is limited to a maximum positive value of 99999999, and larger numbers having more than eight characters are unacceptable. The format of the input can change from free to fixed anywhere in the input file. The input is case insensitive and keywords can be given in either upper or lower case. *The asterisks "*" preceding each keyword must be in column one.*

To provide a better understanding behind the keyword philosophy and how the options work, a brief review the keywords is given below.

*AIRBAG

The geometric definition of airbags and the thermodynamic properties for the airbag inflator models can be made in this section. This capability is not necessarily limited to the modeling of automotive airbags, but it can also be used for many other applications such as tires and pneumatic dampers.

*ALE

This keyword provides a way of defining input data pertaining to the Arbitrary-Lagrangian-Eulerian capability.

*BOUNDARY

This section applies to various methods of specifying either fixed or prescribed boundary conditions. For compatibility with older versions of LS-DYNA it is still possible to specify some nodal boundary conditions in the *NODE card section.

*CASE

This keyword option provides a way of running multiple load cases sequentially. Within each case, the input parameters, which include loads, boundary conditions, control cards, contact definitions, initial conditions, etc., can change. If desired, the results from a previous case can be used during initialization. Each case creates unique file names for all output results files by appending CID*n* to the default file name.

*COMPONENT

This section contains analytical rigid body dummies that can be placed within vehicle and integrated implicitly.

*CONSTRAINED

This section applies constraints within the structure between structural parts. For example, nodal rigid bodies, rivets, spot welds, linear constraints, tying a shell edge to a shell edge with failure, merging rigid bodies, adding extra nodes to rigid bodies and defining rigid body joints are all options in this section.

*CONTACT

This section is divided in to three main sections. The *CONTACT section allows the user to define many different contact types. These contact options are primarily for treating contact of deformable to deformable bodies, single surface contact in deformable bodies, deformable body to rigid body contact, and tying deformable structures with an option to release the tie based on plastic strain. The surface definition for contact is made up of segments on the shell or solid element surfaces. The keyword options and the corresponding numbers in previous code versions are:

STRUCTURED INPUT TYPE ID	KEYWORD NAME
1	SLIDING_ONLY
p 1	SLIDING_ONLY_PENALTY
2	TIED_SURFACE_TO_SURFACE

3	SURFACE_TO_SURFACE
a 3	AUTOMATIC_SURFACE_TO_SURFACE
4	SINGLE_SURFACE
5	NODES_TO_SURFACE
a 5	AUTOMATIC_NODES_TO_SURFACE
6	TIED_NODES_TO_SURFACE
7	TIED_SHELL_EDGE_TO_SURFACE
8	TIEBREAK_NODES_TO_SURFACE
9	TIEBREAK_SURFACE_TO_SURFACE
10	ONE_WAY_SURFACE_TO_SURFACE
a 10	AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE
13	AUTOMATIC_SINGLE_SURFACE
a 13	AIRBAG_SINGLE_SURFACE
14	ERODING_SURFACE_TO_SURFACE
15	ERODING_SINGLE_SURFACE
16	ERODING_NODES_TO_SURFACE
17	CONSTRAINT_SURFACE_TO_SURFACE
18	CONSTRAINT_NODES_TO_SURFACE
19	RIGID_BODY_TWO_WAY_TO_RIGID_BODY
20	RIGID_NODES_TO_RIGID_BODY
21	RIGID_BODY_ONE_WAY_TO_RIGID_BODY
22	SINGLE_EDGE
23	DRAWBEAD

The *CONTACT_ENTITY section treats contact between a rigid surface, usually defined as an analytical surface, and a deformable structure. Applications of this type of contact exist in the metal forming area where the punch and die surface geometries can be input as VDA surfaces which are treated as rigid. Another application is treating contact between rigid body occupant dummy hyper-ellipsoids and deformable structures such as airbags and instrument panels. This option is particularly valuable in coupling with the rigid body occupant modeling codes MADYMO and CAL3D. The *CONTACT_1D is for modeling rebars in concrete structure.

*CONTROL

Options available in the *CONTROL section allow the resetting of default global parameters such as the hourglass type, the contact penalty scale factor, shell element formulation, numerical damping, and termination time.

*DAMPING

Defines damping either globally or by part identifier.

*DATABASE

This keyword with a combination of options can be used for controlling the output of ASCII databases and binary files output by LS-DYNA. With this keyword the frequency of writing the various databases can be determined.

*DEFINE

This section allows the user to define curves for loading, constitutive behaviors, etc.; boxes to limit the geometric extent of certain inputs; local coordinate systems; vectors; and orientation vectors specific to spring and damper elements. Items defined in this section are referenced by their identifiers throughout the input. For example, a coordinate system identifier is sometimes used on the *BOUNDARY cards, and load curves are used on the *AIRBAG cards.

*DEFORMABLE_TO_RIGID

This section allows the user to switch parts that are defined as deformable to rigid at the start of the analysis. This capability provides a cost efficient method for simulating events such as rollover events. While the vehicle is rotating the computation cost can be reduced significantly by switching deformable parts that are not expected to deform to rigid parts. Just before the vehicle comes in contact with ground, the analysis can be stopped and restarted with the part switched back to deformable.

*ELEMENT

Define identifiers and connectivities for all elements which include shells, beams, solids, thick shells, springs, dampers, seat belts, and concentrated masses in LS-DYNA.

*EOS

This section reads the equations of state parameters. The equation of state identifier, EOSID, points to the equation of state identifier on the *PART card.

*HOURGLASS

Defines hourglass and bulk viscosity properties. The identifier, HGID, on the *HOURGLASS card refers to HGID on *PART card.

*INCLUDE

To make the input file easy to maintain, this keyword allows the input file to be split into sub-files. Each sub-file can again be split into sub-sub-files and so on. This option is beneficial when the input data deck is very large.

*INITIAL

Initial velocity and initial momentum for the structure can be specified in this section. The initial velocity specification can be made by *INITIAL_VELOCITY_NODE card or *INITIAL_VELOCITY cards. In the case of *INITIAL_VELOCITY_NODE nodal identifiers are used to specify the velocity components for the node. Since all the nodes in the system are initialized to zero, only the nodes with non-zero velocities need to be specified. The *INITIAL_VELOCITY card provides the capability of being able to specify velocities using the set concept or boxes.

*INTEGRATION

In this section the user defined integration rules for beam and shell elements are specified. IRID refers to integration rule number IRID on *SECTION_BEAM and *SECTION_SHELL cards respectively. Quadrature rules in the *SECTION_SHELL and *SECTION_BEAM cards need to be specified as a negative number. The absolute value of the negative number refers to user defined integration rule number. Positive rule numbers refer to the built in quadrature rules within LS-DYNA.

*INTERFACE

Interface definitions are used to define surfaces, nodal lines, and nodal points for which the displacement and velocity time histories are saved at some user specified frequency. This data may then be used in subsequent analyses as an interface ID in the *INTERFACE_LINKING_DISCRETE_NODE as master nodes, in *INTERFACE_LINKING_-SEGMENT as master segments and in *INTERFACE_LINKING_EDGE as the master edge for a series of nodes.

This capability is especially useful for studying the detailed response of a small member in a large structure. For the first analysis, the member of interest need only be discretized sufficiently that the displacements and velocities on its boundaries are reasonably accurate. After the first analysis is completed, the member can be finely discretized in the region bounded by the interfaces. Finally, the second analysis is performed to obtain highly detailed information in the local region of interest.

When beginning the first analysis, specify a name for the interface segment file using the Z=parameter on the LS-DYNA execution line. When starting the second analysis, the name of the interface segment file created in the first run should be specified using the L=parameter on the LS-DYNA command line. Following the above procedure, multiple levels of sub-modeling are easily accommodated. The interface file may contain a multitude of interface definitions so that a single run of a full model can provide enough interface data for many component analyses. The interface feature represents a powerful extension of LS-DYNA's analysis capabilities. A similar capability using *INTERFACE_SSI may be used for soil-structure interaction analysis under earthquake excitation.

*KEYWORD

Flags LS-DYNA that the input deck is a keyword deck. To have an effect this must be the very first card in the input deck. Alternatively, by typing "keyword" on the execute line, keyword input formats are assumed and the "*KEYWORD" is not required. If a number is specified on this card after the word KEYWORD it defines the memory size to used in words. The memory size can also be set on the command line.

NOTE: The memory specified on the execution line overrides memory specified on the *keyword card.

*LOAD

This section provides various methods of loading the structure with concentrated point loads, distributed pressures, body force loads, and a variety of thermal loadings.

*MAT

This section allows the definition of constitutive constants for all material models available in LS-DYNA including springs, dampers, and seat belts. The material identifier, MID, points to the MID on the *PART card.

*NODE

Define nodal point identifiers and their coordinates.

*PARAMETER

This option provides a way of specifying numerical values of parameter names that are referenced throughout the input file. The parameter definitions, if used, should be placed at the beginning of the input file following *KEYWORD. *PARAMETER_EXPRESSION permits general algebraic expressions to be used to set the values.

*PART

This keyword serves two purposes.

1. Relates part ID to *SECTION, *MATERIAL, *EOS and *HOURGLASS sections.

2. Optionally, in the case of a rigid material, rigid body inertia properties and initial conditions can be specified. Deformable material repositioning data can also be specified in this section if the reposition option is invoked on the *PART card, i.e., *PART_REPOSITION.

*PERTURBATION

This keyword provides a way of defining deviations from the designed structure such as, buckling imperfections.

*RAIL

This keyword provides a way of defining a wheel-rail contact algorithm intended for railway applications but can also be used for other purposes. The wheel nodes (defined on *RAIL_TRAIN) represent the contact patch between wheel and rail.

*RIGIDWALL

Rigid wall definitions have been divided into two separate sections, PLANAR and GEOMETRIC. Planar walls can be either stationary or moving in translational motion with mass and initial velocity. The planar wall can be either finite or infinite. Geometric walls can be planar as well as have the geometric shapes such as rectangular prism, cylindrical prism and sphere. By default, these walls are stationary unless the option MOTION is invoked for either prescribed translational velocity or displacement. Unlike the planar walls, the motion of the geometric wall is governed by a load curve. Multiple geometric walls can be defined to model combinations of geometric shapes available. For example, a wall defined with the CYLINDER option can be combined with two walls defined with the SPHERICAL option to model hemispherical surface caps on the two ends of a cylinder. Contact entities are also analytical surfaces but have the significant advantage that the motion can be influenced by the contact to other bodies, or prescribed with six full degrees-of-freedom.

*SECTION

In this section, the element formulation, integration rule, nodal thicknesses, and cross sectional properties are defined. All section identifiers (SECID's) defined in this section must be unique, i.e., if a number is used as a section ID for a beam element then this number cannot be used again as a section ID for a solid element.

*SENSOR

This keyword provides a convenient way of activating and deactivating boundary conditions, airbags, discrete elements, joints, contact, rigid walls, single point constraints, and constrained nodes. The sensor capability is new in the second release of version 971 and will evolve in later releases to encompass many more LS-DYNA capabilities and replace some of the existing capabilities such as the airbag sensor logic.

*SET

A concept of grouping nodes, elements, materials, etc., in sets is employed throughout the LS-DYNA input deck. Sets of data entities can be used for output. So-called slave nodes used in contact definitions, slaves segment sets, master segment sets, pressure segment sets and so on can also be defined. The keyword, *SET, can be defined in two ways:

1. Option LIST requires a list of entities, eight entities per card, and define as many cards as needed to define all the entities.

2. Option COLUMN, where applicable, requires an input of one entity per line along with up to four attribute values which are used by other keywords to specify, for example, the failure criterion input that is needed for *CONTACT_-CONSTRAINT_NODES_TO_SURFACE.

*TERMINATION

This keyword provides an alternative way of stopping the calculation before the termination time is reached. The termination time is specified on the *CONTROL_TERMINATION input and will terminate the calculation whether or not the options available in this section are active.

*TITLE

In this section a title for the analysis is defined.

*USER_INTERFACE

This section provides a method to provide user control of some aspects of the contact algorithms including friction coefficients via user defined subroutines.

RESTART

This section of the input is intended to allow the user to restart the simulation by providing a restart file and optionally a restart input defining changes to the model such as deleting contacts, materials, elements, switching materials from rigid to deformable, deformable to rigid, etc.

*RIGID_DEFORMABLE

This section switches rigid parts back to deformable in a restart to continue the event of a part impacting the ground which may have been modeled with a rigid wall.

*STRESS_INITIALIZATION

This is an option available for restart runs. In some cases there may be a need for the user to add contacts, elements, etc., which are not available options for standard restart runs. A full input containing the additions is needed if this option is invoked upon restart.

SUMMARY OF COMMONLY USED OPTIONS

The following table gives a list of the commonly used keywords related by topic.

Topic	Component	Keywords
Geometry	Nodes	*NODE
	Elements	*ELEMENT_BEAM *ELEMENT_SHELL *ELEMENT_SOLID *ELEMENT_TSHELL
	Discrete Elements	*ELEMENT_DISCRETE *ELEMENT_SEATBELT *ELEMENT_MASS
Materials	Part	PART cards glues the model together: $\text{*PART} \rightarrow \begin{cases} \text{*MAT} \\ \text{*SECTION} \\ \text{*EOS} \\ \text{*HOURGLASS} \end{cases}$
	Material	*MAT
	Sections	*SECTION_BEAM *SECTION_SHELL *SECTION_SOLID *SECTION_TSHELL
	Discrete sections	*SECTION_DISCRETE *SECTION_SEATBELT
	Equation of state	*EOS
	Hourglass	*CONTROL_HOURGLASS *HOURGLASS
Contacts & Rigid walls	Defaults for contacts	*CONTROL_CONTACT
	Definition of contacts	*CONTACT_OPTION
	Definition of rigid walls	*RIGIDWALL_OPTION

Topic	Component	Keywords
Boundary Conditions & Loadings	Restraints	*NODE *BOUNDARY_SPC_*OPTION*
	Gravity (body) load	*LOAD_BODY_*OPTION*
	Point load	*LOAD_NODE_*OPTION*
	Pressure load	*LOAD_SEGMENT_*OPTION* *LOAD_SHELL_*OPTION*
	Thermal load	*LOAD_THERMAL_*OPTION*
	Load curves	*DEFINE_CURVE
Constraints and spot welds	Constrained nodes	*CONSTRAINED_NODE_SET
	Welds	*CONSTRAINED_GENERALIZED_WELD *CONSTRAINED_SPOT_WELD
	Rivet	*CONSTRAINED_RIVET
Output Control	Items in time history blocks	*DATABASE_HISTORY_*OPTION*
	Default	*CONTROL_OUTPUT
	ASCII time history files	*DATABASE_OPTION
	Binary plot/time history/restart files	*DATABASE_BINARY_OPTION
	Nodal reaction output	*DATABASE_NODAL_FORCE_GROUP
Termination	Termination time	*CONTROL_TERMINATION
	Termination cycle	*CONTROL_TERMINATION
	CPU termination	*CONTROL_CPU
	Degree of freedom	*TERMINATION_NODE

Table 2.1. Keywords for the most commonly used options.

EXECUTION SYNTAX

The execution line for LS-DYNA, sometimes referred to as the command line, is as follows:

```
LS-DYNA I=inf O=otf G=ptf D3PART=d3part D=dpf F=thf T=tpf A=rrd
M=sif S=iff H=iff Z=isf1 L=isf2 B=rlf W=root E=efl X=scl C=cpu K=kill
V=vda Y=c3d BEM=bof {KEYWORD} {THERMAL} {COUPLE} {INIT}
{CASE} {PGPKEY} MEMORY=nwds MODULE=dll NCPU=ncpu
PARA=para ENDTIME=time NCYCLE=ncycle JOBID=jobid
D3PROP=d3prop GMINP=gminp GMOUT=gmout MCHECK=y
```

where,

inf = input file (user specified)

otf = high speed printer file (default = d3hsp)

ptf = binary plot file for postprocessing (default = d3plot)

d3part = binary plot file for subset of parts; see *DATABASE_BINARY_D3PART (default = d3part)

dpf = dump file to write for purposes of restarting (default = d3dump). This file is written at the end of every run and during the run as requested by *DATABASE_BINARY_D3DUMP. To stop the generation of this dump file, specify "d=nodump" (case insensitive).

thf = binary plot file for time histories of selected data (default = d3thdt)

tpf = optional temperature file

rrd = running restart dump file (default = runrsf)

sif = stress initialization file (user specified)

iff = interface force file (user specified). See *DATBASE_BINARY_INTFOR and *DATABASE_BINARY_FSIFOR.

isf1 = interface segment save file to be created (default = infmak)

isf2 = existing interface segment save file to be used (user specified)

rlf = binary plot file for dynamic relaxation (default = d3drfl)

efl = echo file containing optional input echo with or without node/element data

root = root file name for general print option

scl = scale factor for binary file sizes (default =70)

cpu = cumulative cpu time limit in seconds for the entire simulation, including all restarts, if cpu is positive. If cpu is negative, the absolute value of cpu is the cpu time limit in seconds for the first run and for each subsequent restart run.

kill = if LS-DYNA encounters this file name it will terminate with a restart file (default = d3kil)

vda = VDA/IGES database for geometrical surfaces

c3d = CAL3D input file

bof = *FREQUENCY_DOMAIN_ACOUSTIC_BEM output file

nwds = Number of words to be allocated. On engineering workstations a word isusually 32bits. This number overwrites the memory size specified on the *KEYWORD card at the beginning of the input deck.

dll = The dynamic library for user subroutines. Only one dynamic library can be loaded via "module=dll". See *MODULE_LOAD command for loading multiple dynamic libraries.

ncpu = Overrides NCPU and CONST defined in *CONTROL_PARALLEL. A positive value sets CONST = 2 and a negative values sets CONST = 1. See the *CONTROL_PARALLEL command for an explanation of these parameters. The *KEYWORD command provides an alternative way to set the number of CPUs.

para = Overrides PARA defined in *CONTROL_PARALLEL.

time = Overrides ENDTIM defined in *CONTROL_TERMINATION.

ncycle = Overrides ENDCYC defined in *CONTROL_TERMINATION.

jobid = Character string which acts as a prefix for all output files. Maximum length is 72 characters. Do not include the following characters:) (* / ? \.

d3prop = See *DATABASE_BINARY_D3PROP input parameter IFILE for options.

gminp = Input file for reading recorded motions in *INTERFACE_SSI (default = gmbin).

gmout = Output file for writing recorded motions in *INTERFACE_SSI_AUX (default = gmbin).

In order to avoid undesirable or confusing results, each LS-DYNA run should be performed in a separate directory, unless using the command line parameter "jobid" described above. If rerunning a job in the same directory, old files should first be removed or renamed to avoid confusion since the possibility exists that the binary database may contain results from both the old and new run.

By including "keyword" anywhere on the execute line or instead if *KEYWORD is the first card in the input file, the keyword formats are expected; otherwise, the older structured input file will be expected.

To run a coupled thermal analysis the command "couple" must be in the execute line. A thermal only analysis may be run by including the word "thermal" in the execution line.

The execution line option “pgpkey” will output the current public PGP key used by LS-DYNA for encryption of input. The public key and some instructions on how to use the key are written to the screen as well as a file named “lstc_pgpkey.asc”.

The “init” (or sw1. can be used instead) command on the execution line causes the calculation to run just one cycle followed by termination with a full restart file. No editing of the input deck is required. The calculation can then be restarted with or without any additional input. Sometimes this option can be used to reduce the memory on restart if the required memory is given on the execution line and is specified too large in the beginning when the amount of required memory is unknown. Generally, this option would be used at the beginning of a new calculation.

If the word “case” appears on the command line, then *CASE statements will be handled by the built in driver routines. Otherwise they should be processed by the external “lscasedriver” program, and if any *CASE statements are encountered it will cause an error.

If “mcheck=y” is given on the command line, the program switches to “model check” mode. In this mode the program will run only 10 cycles – just enough to verify that the model will start. For implicit problems, all initialization is performed, but execution halts before the first cycle. If the network license is being used, the program will attempt to check out a license under the program name “LS-DYNAMC” so as not to use up one of the normal DYNA licenses. If this fails, a normal execution license will be used.

If the word “memory” is found anywhere on the execution line and if it is not set via “memory=nwds” LS-DYNA will give the default size of memory, request, and then read in the desired memory size. This option is necessary if the default value is insufficient memory and termination occurs as a result. Occasionally, the default value is too large for execution and this option can be used to lower the default size. Memory can also be specified on the *KEYWORD card.

SENSE SWITCH CONTROLS

The status of an in-progress LS-DYNA simulation can be determined by using the sense switch. On UNIX versions, this is accomplished by first typing a “^C” (Control-C). This sends an interrupt to LS-DYNA which is trapped and the user is prompted to input the sense switch code. LS-DYNA has nine terminal sense switch controls that are tabulated below:

Type	Response
SW1.	A restart file is written and LS-DYNA terminates.

Type	Response
SW2.	LS-DYNA responds with time and cycle numbers.
SW3.	A restart file is written and LS-DYNA continues.
SW4.	A plot state is written and LS-DYNA continues.
SW5.	Enter interactive graphics phase and real time visualization.
SW7.	Turn off real time visualization.
SW8.	Interactive 2D rezoner for solid elements and real time visualization.
SW9.	Turn off real time visualization (for option SW8).
SWA.	Flush ASCII file buffers.
lprint	Enable/Disable printing of equation solver memory, cpu requirements.
nlprint	Enable/Disable printing of nonlinear equilibrium iteration information.
iter	Enable/Disable output of binary plot database "d3iter" showing mesh after each equilibrium iteration. Useful for debugging convergence problems.
conv	Temporarily override nonlinear convergence tolerances.
stop	Halt execution immediately, closing open files.

On UNIX/LINUX and Windows systems the sense switches can still be used if the job is running in the background or in batch mode. To interrupt LS-DYNA simply create a file called d3kil containing the desired sense switch, e.g., "sw1." LS-DYNA periodically looks for this file and if found, the sense switch contained therein is invoked and the d3kil file is deleted. A null d3kil file is equivalent to a "sw1."

When LS-DYNA terminates, all scratch files are destroyed: the restart file, plot files, and high-speed printer files remain on disk. Of these, only the restart file is needed to continue the interrupted analysis.

PROCEDURE FOR LS-DYNA/MPP

As described above the serial/SMP code supports the use of the SIGINT signal (usually Ctrl-C) to interrupt the execution and prompt the user for a "sense switch." The MPP code also supports this capability. However, on many systems a shell script or front end program (generally "mpirun") is required to start MPI applications. Pressing Ctrl-C on some systems will kill this process, and thus kill the running MPP-DYNA executable. On UNIX/LINUX systems, as workaround, when the MPP code begins execution it creates a file named, "bg_switch", in the current working directory. This file contains the following single line:

```
rsh <machine name> kill -INT <PID>
```

where <machine name> is the hostname of the machine on which the root MPP-DYNA process is running, and <PID> is its process id. (on HP systems, "rsh" is replaced by "remsh"). Thus, simply executing this file will send the appropriate signal. For Windows, usually the D3KIL file is used to interrupt the execution.

For more information about running the LS-DYNA/MPP Version see Appendix O.

Files: Input and Output

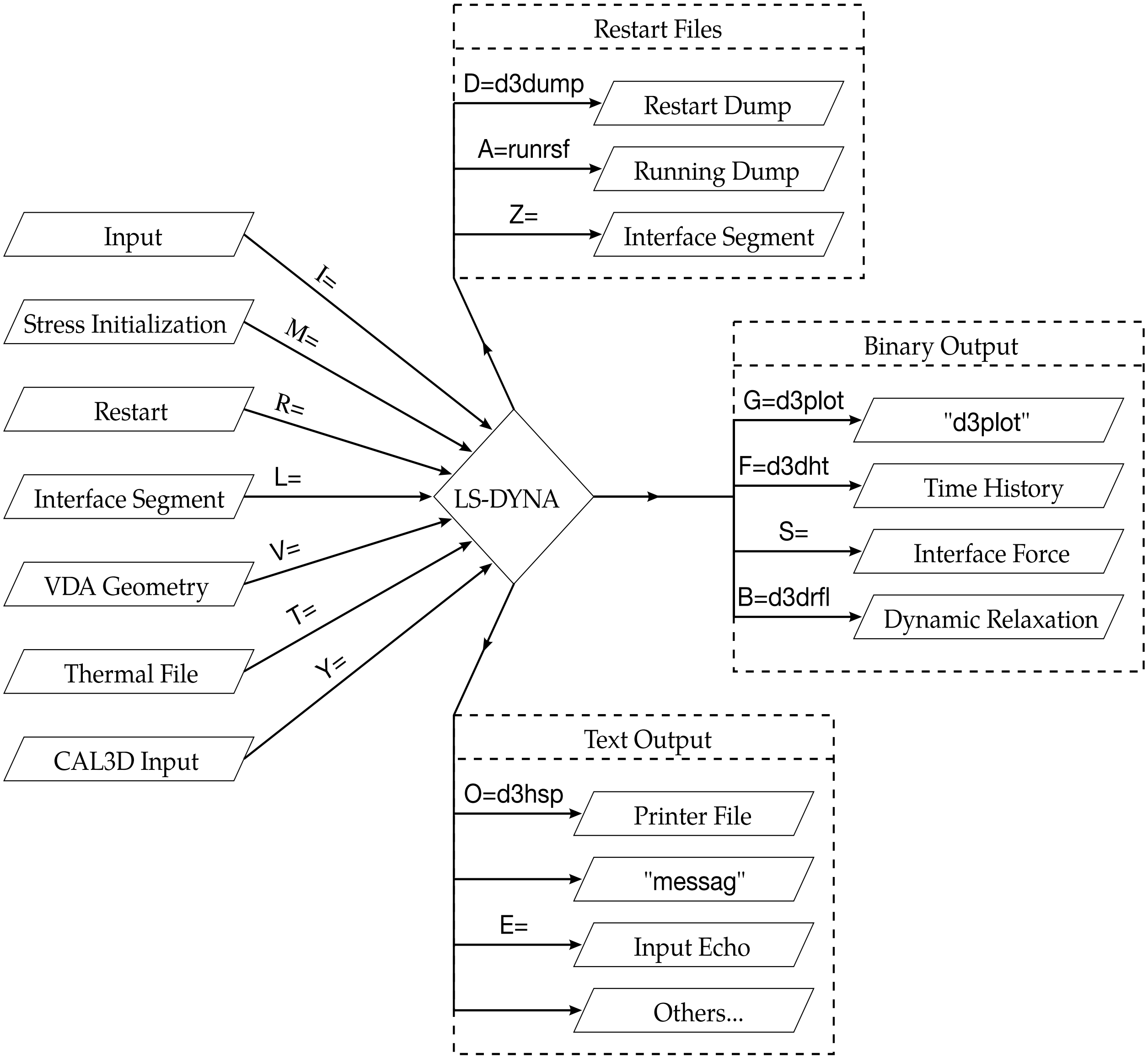


Figure 2-2. Files Input and Output.

FILES

1. **Uniqueness.** File names must be unique.

2. **Interface forces.** The interface force file is created only if it is specified on the execution line "S=**iff**".

3. **File size limits.** For very large models, the default size limits for binary output files may not be large enough for a single file to hold even a single plot state, in

which case the file size limit may be increased by specifying "X=**scl**" on the execution line. The default file size limit (X=70) is 70 times one-million octal words or 18.35 Mwords. That translates into 73.4 Mbytes (for 32-bit output) or 146.8 Mbytes (for 64-bit output).

4. **CPU limits.** Using "C=**cpu**" defines the maximum CPU usage allowed. When the CPU usage limit is exceeded LS-DYNA will terminate with a restart file. During a restart, **cpu** should be set to the total CPU used up to the current restart plus whatever amount of additional time is wanted.

5. **File usage in restart.** When restarting from a dump file, the execution line becomes

   LS-DYNA I=**inf** O=**otf** G=**ptf** D=**dpf** R=**rtf** F=**thf** T=**tpf** A=**rrd** S=**iff** Z=**isf1** L=**isf2** B=**rlf** W=**root** E=**efl** X=**scl** C=**cpu** K=**kill** Q=**option** KEYWORD MEMORY=**nwds**

   where,

   rtf=[name of dump file written by LS-DYNA]

   The root names of the dump files written by LS-DYNA = are controlled by dpf (default = d3dump) and rrd (default = runrsf). A two-digit number follows the root name, e.g., d3dump01, d3dump02, etc., to distinguish one dump file from another. Typically, each dump file corresponds to a different simulation time.

   The adaptive dump files contain all information required to successfully restart so that no other files are needed except when CAD surface data is used. When restarting a problem that uses VDA/IGES surface data, the vda input file must be specified, e.g.:

   LS-DYNA R=**d3dump01** V=**vda**

   If the data from the last run is to be remapped onto a new mesh, then specify: "Q=**remap**". The remap file is the dump file from which the remapping data is taken. The remap option is available in SMP for brick elements only, MPP does not support this option currently.

   No stress initialization is possible at restart. Also the VDA files and the CAL3D files must not be changed.

6. **Default file names.** File name dropouts are permitted; for example, the following execution lines are acceptable:

   LS-DYNA I=**inf**

   and

LS-DYNA R=**rtf**

7. **Interface segments.** For an analysis using interface segments, the execution line in the first analysis is given by:

   LS-DYNA I=**inf** Z=**isf1**

   and in the second by:

   LS-DYNA I=**inf** L=**isf1**

8. **Batch execution.** In some installations (e.g., GM) calculations are controlled by a file called "names" on unit 88. The names files consists of two lines in which the second line is blank. The first line of names contains the execution line, for instance:

   I=**inf**

   For a restart the execution line becomes:

   I=**inf** R=**rtf**

RESTART ANALYSIS

The LS-DYNA restart capability allows analyses to be broken down into stages. After the completion of each stage in the calculation a "restart dump" is written that contains all information necessary to continue the analysis. The size of this "dump" file is roughly the same size as the memory required for the calculation. Results can be checked at each stage by post-processing the output databases in the normal way, so the chance of wasting computer time on incorrect analyses is reduced.

The restart capability is frequently used to modify models by deleting excessively distorted elements, materials that are no longer important, and contact surfaces that are no longer needed. Output frequencies of the various databases can also be altered. Often, these simple modifications permit a calculation that might otherwise not to continue on to a successful completion. Restarting can also help to diagnose why a model is giving problems. By restarting from a dump that is written before the occurrence of a numerical problem and obtaining output at more frequent intervals, it is often possible to identify where the first symptoms appear and what aspect of the model is causing them.

The format of the restart input file is described in this manual. If, for example, the user wishes to restart the analysis from dump state *nn*, contained in file D3DUMP*nn*, then the following procedure is followed:

1. Create the restart input deck, if required, as described in the Restart Section of this manual. Call this file restartinput.

2. Start dyna from the command line by invoking:

   LS-DYNA I=**restartinput** R=**D3DUMP*nn***

3. If no alterations to the model are made, then the execution line:

   LS-DYNA R=**D3DUMP*nn***

   will suffice. Of course, the other output files should be assigned names from the command line if the defaults have been changed in the original run.

The full deck restart option allows the user to begin a new analysis, with deformed shapes and stresses carried forward from a previous analysis for selected materials. The new analysis can be different from the original, e.g., more contact surfaces, different geometry (of parts which are not carried forward), etc. Examples of applications include:

- Crash analysis continued with extra contact surfaces;
- Sheet metalforming continued with different tools for modeling a multi-stage forming process.

A typical restart file scenario:

Dyna is run using an input file named "job1.inf", and a restart dump named "d3dump01" is created. A new input file, "job2.inf", is generated and submitted as a restart with, "R=d3dump01", as the dump file. The input file job2.inf contains the entire model in its original *undeformed* state but with more contact surfaces, new output databases, and so on.

Since this is a restart job, information must be given to tell LS-DYNA which parts of the model should be initialized in the full deck restart. When the calculation begins the restart database contained in the file d3dump01 is read, and a new database is created to initialize the model in the input file, job2.inf. The data in file job2.inf is read and the LS-DYNA proceeds through the entire input deck and initialization. At the end of the initialization process, all the parts selected are initialized from the data saved from d3dump01. This means that the deformed position and velocities of the nodes on the elements of each part, and the stresses and strains in the elements (and, if the material of the part is rigid, the rigid body properties) will be assigned.

It is assumed during this process that any initialized part has the same elements, in the same order, with the same topology, in job1 and job2. If this is not the case, the parts cannot be initialized. However, the parts may have different identifying numbers.

For discrete elements and seat belts, the choice is all or nothing. All discrete and belt elements, retractors, sliprings, pretensioners and sensors must exist in both files and will be initialized.

Materials which are not initialized will have no initial deformations or stresses. However, if initialized and non-initialized materials have nodes in common, the nodes will be moved by the initialized material causing a sudden strain in the non-initialized material. *This effect can give rise to sudden spikes in loading.*

Points to note are:

- Time and output intervals are continuous with job1, i.e., the time is not reset to zero.
- Don't try to use the restart part of the input to change anything since this will be overwritten by the new input file.
- Usually, the complete input file part of job2.inf will be copied from job1.inf, with the required alterations. We again mention that there is no need to update the nodal coordinates since the deformed shapes of the initialized materials will be carried forward from job1.
- Completely new databases will be generated with the time offset.

VDA/IGES DATABASES

VDA surfaces are surfaces of geometric entities which are given in the form of polynomials. The format of these surfaces is as defined by the German automobile and supplier industry in the VDA guidelines, [VDA 1987].

The advantage of using VDA surfaces is twofold. First, the problem of meshing the surface of the geometric entities is avoided and, second, smooth surfaces can be achieved which are very important in metalforming. With smooth surfaces, artificial friction introduced by standard faceted meshes with corners and edges can be avoided. This is a big advantage in springback calculations.

A very simple and general handling of VDA surfaces is possible allowing arbitrary motion and generation of surfaces. For a detailed description, see Appendix L.

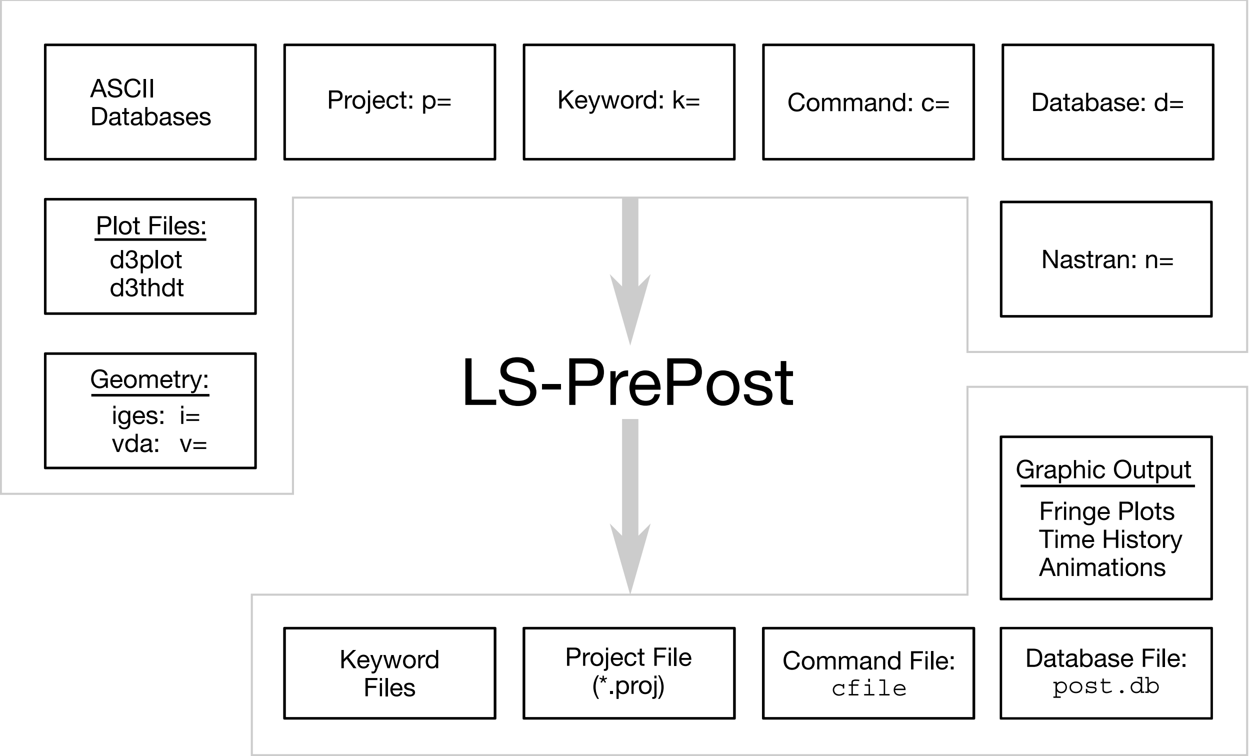


Figure 2-3. File Organization

LS-PrePost®

LS-DYNA is designed to operate with a variety of commercial pre- and post-processing packages. Currently, direct support is available from TRUEGRID, PATRAN, eta/VPG, HYPERMESH, EASi-CRASH DYNA and FEMAP. Several third-party translation programs are available for PATRAN and IDEAS.

Alternately, the pre- and post-processor LS-PrePost is available from LSTC and is specialized for LS-DYNA. LS-PrePost is an advanced pre- and post-processor that is delivered free with LS-DYNA. The user interface is designed to be both efficient and intuitive. LS-PrePost runs on Windows, Linux, and Unix, utilizing OpenGL graphics to achieve fast model rendering and XY plotting.

Some of the capabilities available in LS-PrePost are:

- Complete support for all LS-DYNA keyword data.
- Importing and combining multiple models from many sources (LS-DYNA keyword, IDEAS neutral file, NASTRAN bulk data, STL ASCII, and STL binary formats).
- Improved renumbering of model entities.
- Model Manipulation: Translate, Rotate, Scale, Project, Offset, Reflect

- LS-DYNA Entity Creation: Coordinate Systems, Sets, Parts, Masses, CNRBs, Boxes, Spot welds, SPCs, Rigidwalls, Rivets, Initial Velocity, Accelerometers, Cross Sections, etc.
- Mesh Generation: 2Dmesh Sketchboard, nLine Meshing, Line sweep into shell, Shell sweep into solid, Tet-Meshing, Automatic surface meshing of IGES and VDA data, Meshing of simple geometric objects (Plate, Sphere, Cylinder)
- Special Applications: Airbag folding, Dummy positioning, Seatbelt fitting, Initial penetration check, Spot weld generation using MAT_100
- Complete support of LS-DYNA results data file: d3plot file, d3thdt file, All ASCII time history data file, Interface force file

LS-PrePost processes output from LS-DYNA. LS-PrePost reads the binary plot-files generated by LS-DYNA and plots contours, fringes, time histories, and deformed shapes. Color contours and fringes of a large number of quantities may be interactively plotted on meshes consisting of plate, shell, and solid type elements. LS-PrePost can compute a variety of strain measures, reaction forces along constrained boundaries.

LS-DYNA generates three binary databases. One contains information for complete states at infrequent intervals; 50 to 100 states of this sort is typical in a LS-DYNA calculation. The second contains information for a subset of nodes and elements at frequent intervals; 1000 to 10,000 states is typical. The third contains interface data for contact surfaces.

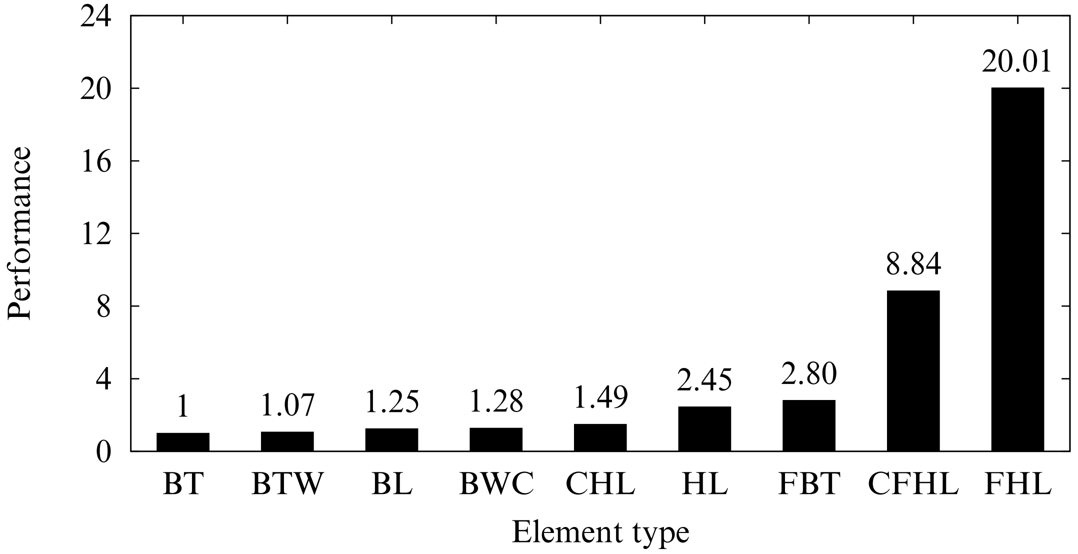


Figure 2-4. Relative cost of the four noded shells available in LS-DYNA where BT is the Belytschko-Tsay shell, BTW is the Belytschko-Tsay shell with the warping stiffness taken from the Belytschko-Wong-Chiang, BWC, shell. The BL shell is the Belytschko-Leviathan shell. CHL denotes the Hughes-Liu shell, HL, with one point quadrature and a co-rotational formulation. FBT is a Belytschko-Tsay like shell with full integration, FHL is the fully integrated Hughes-Liu shell, and the CFHL shell is its co-rotational version.

EXECUTION SPEEDS

The relative execution speeds for various elements in LS-DYNA are tabulated below:

Element Type	Relative Cost
8 node solid with 1 point integration and default hourglass control	4
as above but with Flanagan-Belytschko hourglass control	5
constant stress and Flanagan-Belytschko hourglass control, i.e., the Flanagan-Belytschko element	7
4 node Belytschko-Tsay shell with four thickness integration points	4
4 node Belytschko-Tsay shell with resultant plasticity	3

Element Type	Relative Cost
BCIZ triangular shell with four thickness integration points	7
C^0 triangular shell with four thickness integration points	4
2 node Hughes-Liu beam with four integration points	9
2 node Belytschko-Schwer beam	2
2 node simple truss elements	1
8 node solid-shell with four thickness integration points	11

These relative timings are very approximate. Each interface node of the sliding interfaces is roughly equivalent to one-half zone cycle in cost. Figure 2-4. illustrates the relative cost of the various shell formulations in LS-DYNA.

UNITS

The units in LS-DYNA must be consistent. One way of testing whether a set of units is consistent is to check that:

$$[\text{force unit}] = [\text{mass unit}] \times [\text{acceleration unit}]$$

and that

$$[\text{acceleration unit}] = \frac{[\text{length unit}]}{[\text{time unit}]^2}.$$

Examples of sets of consistent units are tabulated below. For a more comprehensive table, see http://ftp.lstc.com/anonymous/outgoing/support/FAQ/consistent_units .

	(a)	(b)	(c)
Length unit	meter	millimeter	millimeter
Time unit	second	second	millisecond
Mass unit	kilogram	tonne	kilogram
Force unit	Newton	Newton	kiloNewton
Young’s Modulus of Steel	210.0E+09	210.0E+03	210.0
Density of Steel	7.85E+03	7.85E–09	7.85E–06
Yield stress of Mild Steel	200.0E+06	200.0	0.200
Acceleration due to gravity	9.81	9.81E+03	9.81E-03
Velocity equivalent to 30 mph	13.4	13.4E+03	13.4

GENERAL CARD FORMAT

The following sections specify, for each keyword command, the cards that must be defined and those cards that are optional. Each card is described in its fixed format form and is shown as a number of fields in an 80 character string. With the exception of "long format input" as described later in this section, most cards are 8 fields with a field length of 10 characters. A sample card is shown below. The card format is clearly stated when it is different than 8 fields of 10 characters.

As an alternative to fixed format, a card may be in free format with the values of the variables separated by commas. When using comma-delimited values on a card, the number of characters used to specify a value must not exceed the field length for fixed format. For example, an I8 number is limited to a value of 99999999 and a larger number with more than 8 characters is unacceptable. A further restriction is that characters beyond column 80 of each line are ignored by the code. Fixed format and free, comma-delimited format can be mixed throughout the deck and even within different cards of a single command but not within a card.

The limits on number of characters per variable and number of characters per line as stated above are raised in the case of long format input. See the description of long format input below.

Example Card.

Card [N]	1	2	3	4	5	6	7	8
Variable	NSID	PSID	A1	A2	A3	KAT		
Type	I	I	F	F	F	I		
Default	none	none	1.0	1.0	0	1		
Remarks	1			2		3		

In the example shown above, the row labeled "Type" gives the variable type and is either F, for floating point or I, for an integer. The row labeled "Default" reveals the default value set for a variable if zero is specified, the field is left blank, or the card is not defined. The "Remarks" row refers to enumerated remarks at the end of the section.

Optional Cards:

Each keyword card (line beginning with "*") is followed by a set of data cards. Data cards are either,

1. *Required Cards.* Unless otherwise indicated, cards are required.

2. *Conditional Cards.* Conditional cards are required provided some condition is satisfied. The following is a typical conditional card:

ID Card. Additional card for the ID keyword option.

ID	1	2	3	4	5	6	7	8
Variable	ABID	HEADING						
Type	I	A70						

3. *Optional Cards.* An optional card is one that may be replaced by the next keyword card. The fields in the omitted optional data cards are assigned their default values.

   Example. Suppose the data set for *KEYWORD consists of 2 required cards and 3 optional cards. Then, the fourth card may be replaced by the next keyword card. All the fields in the omitted fourth and fifth cards are assigned their default values.

WARNING: In this example, even though the fourth card is optional, the input deck may *not* jump from the third to fifth card. The *only* card that card 4 may be replaced with is the next keyword card.

Long Format Input:

To accommodate larger or more precise values for input variables than are allowed by the standard format input as described above, a "long format" input option is available. One way of invoking long format keyword input is by adding "long=**y**" to the execution line. A second way is to add "long=**y**" to the *KEYWORD command in the input deck.

long=**y**: read long keyword input deck; write long structured input deck.
long=**s**: read standard keyword input deck; write long structured input deck.
long=**k**: read long keyword input deck; write standard structured input deck.

The “long=**s**” option may be helpful in the rare event that the keyword input is of standard format but LS-DYNA reports an input error and the dyna.str file (see *CONTROL_STRUCTURED) reveals that one of more variables is incorrectly written to dyna.str as a series of asterisks due to inadequate field length(s) in dyna.str.

The “long=**k**” option really serves no practical purpose.

When long format is invoked for keyword input, field lengths for each variable become 20 characters long (160 character limit per line for 8 variables; 200 character limit per line for 10 variables). In this way, the number of input lines in long format is unchanged from regular format.

To convert a standard format input deck to a long format input deck, run LS-DYNA with “newformat = long” on the execution line. For example, if standard.k is a standard format input deck,

```
ls-dyna i = standard.k newformat = long
```

will create a long format input deck called standard.k.long.

You can mix long and standard format within one input deck by use of “+” or “-“ signs within the deck. If the execution line indicates standard format, you can add “ +” at the end of any keywords to invoke long format just for those keywords. For example, “*NODE +” in place of “*NODE” invokes a read format of two lines per node (I20, 3E20.0 on the first line and 2F20.0 on the second line).

Similarly, if the execution line indicates long format, you can add “-” at the end of any keywords to invoke standard format for those keywords. For example, “*NODE –” in place of “*NODE” invokes the standard read format of one line per node (I8, 3E16.0, 2F8.0).

Taking this idea a step further, adding a “-” or “+” to the end of the *INCLUDE keyword command signals to LS-DYNA that all the commands in the included file are standard format or long format, respectively.

*AIRBAG

Purpose: Define an airbag or control volume.

The keyword *AIRBAG provides a way of defining thermodynamic behavior of the gas flow into the airbag as well as a reference configuration for the fully inflated bag. The keyword cards in this section are defined in alphabetical order:

AIRBAG_*OPTION1_{OPTION2}_{OPTION3}_{OPTION4}

*AIRBAG_ADVANCED_ALE

*AIRBAG_ALE

*AIRBAG_INTERACTION

*AIRBAG_PARTICLE

AIRBAG_REFERENCE_GEOMETRY_*OPTION_OPTION

*AIRBAG_SHELL_REFERENCE_GEOMETRY

*AIRBAG_*OPTION1*_*{OPTION2}*_*{OPTION3}*_*{OPTION4}*

OPTION1 specifies one of the following thermodynamic models:

SIMPLE_PRESSURE_VOLUME

SIMPLE_AIRBAG_MODEL

ADIABATIC_GAS_MODEL

WANG_NEFSKE

WANG_NEFSKE_JETTING

WANG_NEFSKE_MULTIPLE_JETTING

LOAD_CURVE

LINEAR_FLUID

HYBRID

HYBRID_JETTING

HYBRID_CHEMKIN

FLUID_AND_GAS

OPTION2 specifies that an additional line of data is read for the WANG_NEFSKE type thermodynamic relationships. The additional data controls the initiation of exit flow from the airbag. *OPTION2* takes the single option:

POP

OPTION3 specifies that a constant momentum formulation is used to calculate the jetting load on the airbag an additional line of data is read in: *OPTION3* takes the single option:

CM

OPTION4 given by:

ID

Specifies that an airbag ID and heading information will be the first card of the airbag definition. This ID is a unique number that is necessary for the identification of the airbags in the definition of airbag interaction via *AIRBAG_INTERACTION keyword. The numeric ID's and heading are written into the abstat and d3hsp files.

Core Cards: Common to all airbags

ID Card. Additional card for the ID keyword option. To use the *AIRBAG_INTERACTION keyword ID Cards are required.

ID	1	2	3	4	5	6	7	8
Variable	ABID	HEADING						
Type	I	A70						

Card 1a	1	2	3	4	5	6	7	8
Variable	SID	SIDTYP	RBID	VSCA	PSCA	VINI	MWD	SPSF
Type	I	I	I	F	F	F	F	F
Default	none	0	0	1.	1.	0.	0.	0.
Remark			optional					

VARIABLE	DESCRIPTION
ABID	Airbag ID. This must be a unique number.
HEADING	Airbag descriptor. It is suggested that unique descriptions be used.
SID	Set ID
SIDTYP	Set type: EQ.0: segment, NE.0: part set ID.

VARIABLE	DESCRIPTION
RBID	Rigid body part ID for user defined activation subroutine: LT.0: -RBID is taken as the rigid body part ID. Built in sensor subroutine initiates the inflator. Load curves are offset by initiation time. EQ.0: The control volume is active from time zero. GT.0: RBID is taken as the rigid body part ID. User sensor subroutine initiates the inflator. Load curves are offset by initiation time. See Appendix D.
VSCA	Volume scale factor (default = 1.0)
PSCA	Pressure scale factor (default = 1.0)
VINI	Initial filled volume
MWD	Mass weighted damping factor, D
SPSF	Stagnation pressure scale factor, $0 <= \gamma <= 1$

Remarks:

The first card is necessary for all airbag options. The option dependent cards follow.

Lumped parameter control volumes are a mechanism for determining volumes of closed surfaces and applying a pressure based on some thermodynamic relation. The volume is specified by a list of polygons similar to the pressure boundary condition cards or by specifying a material subset which represents shell elements which form the closed boundary. All polygon normal vectors must be oriented to face *outwards* from the control volume, however for *AIRBAG_PARTICLE, which does not rely on control volumes, all polygon normal vectors must be oriented to face *inwards* to get proper volume (see *AIRBAG_PARTICLE for more information). If holes are detected, they are assumed to be covered by planar surfaces.

There are two sets of volume and pressure variables used for each control volume model. First, the finite element model computes a volume V_{femodel} and applies a pressure P_{femodel}. The thermodynamics of a control volume may be computed in a different unit system with its own set of varriables: V_{cvolume} and pressure *P*cvolume which are used for integrating the differential equations for the control volume. The conversion is as follows:

$$V_{\mathrm{cvolume}} = (\mathrm{VSCA} \times V_{\mathrm{femodel}}) - \mathrm{VINI}$$
$$P_{\mathrm{femodel}} = \mathrm{PSCA} \times P_{\mathrm{cvolume}}$$

Where VSCA, PSCA, and VINI are input parameters. Damping can be applied to the structure enclosing a control volume by using a mass weighted damping formula:

$$F_i^d = m_i D(v_i - v_{\mathrm{cg}})$$

where F_i^d is the damping force, m_i is the nodal mass, v_i is the velocity for a node, v_{cg} is the mass weighted average velocity of the structure enclosing the control volume, and D is the damping factor.

An alternative, separate damping is based on the stagnation pressure. The stagnation pressure is roughly the maximum pressure on a flat plate oriented normal to a steady state flow field. The stagnation pressure is defined as $p = \gamma \rho V^2$ where V is the normal velocity of the control volume relative to the ambient velocity, ρ is the ambient air density, and γ is a factor which varies from 0 to 1 and has to be chosen by the user. Small values are recommended to avoid excessive damping.

Sensor input:

The sensor is mounted on a rigid body which is attached to the structure. *The motion of the sensor is evaluated in the local coordinate system of the rigid body. See *MAT_RIGID.* This local system rotates and translates with the rigid material. The default local system for a rigid body is taken as the principal axes of the inertia tensor.

When the user defined criterion for airbag deployment is satisfied, a flag is set and deployment begins. All load curves relating to the mass flow rate versus time are then shifted by the initiation time.

RBID = 0: No rigid body

For this case there is no rigid body, and the control volume is active from time zero. There are no additional sensor cards.

RBID > 0: User supplied sensor subroutine

The value of RBID is taken as a rigid body part ID, and a user supplied sensor subroutine will be called to determine the flag that initiates deployment. See Appendix D for details regarding the user supplied subroutine. For RBID > 0 the additional cards are specified below:

User Subroutine Control Card. This card is read in when RBID > 0.

Card 1b	1	2	3	4	5	6	7	8
Variable	N							
Type	I							
Default	none							

User Subroutine Constant Cards. Define N constants for the user subroutine. Include only the number of cards necessary, i.e. for nine constants use 2 cards.

Card 1c	1	2	3	4	5	6	7	8
Variable	C1	C2	C3	C4	C5			
Type	F	F	F	F	F			
Default	0.	0.	0.	0.	0.			

VARIABLE	DESCRIPTION
N	Number of input parameters (not to exceed 25).
C1, …, CN	Up to 25 constants for the user subroutine.

RBID < 0: User supplied sensor subroutine

The value of –RBID is taken as rigid body part ID and a built in sensor subroutine is called. For RBID < 0 there are three additional cards.

Acceleration Sensor Card.

Card 1d	1	2	3	4	5	6	7	8
Variable	AX	AY	AZ	AMAG	TDUR			
Type	F	F	F	F	F			
Default	0.	0.	0.	0.	0.			

Velocity Sensor Card.

Card 1e	1	2	3	4	5	6	7	8
Variable	DVX	DVY	DVZ	DVMAG				
Type	F	F	F	F				
Default	0.	0.	0.	0.				

Displacement Sensor Card.

Card 1f	1	2	3	4	5	6	7	8
Variable	UX	UY	UZ	UMAG				
Type	F	F	F	F				
Default	0.	0.	0.	0.				

VARIABLE **DESCRIPTION**

AX

Acceleration level in local x-direction to activate inflator. The absolute value of the x-acceleration is used.

EQ.0: inactive.

AY

Acceleration level in local y-direction to activate inflator. The absolute value of the y-acceleration is used.

EQ.0: inactive.

AZ

Acceleration level in local z-direction to activate inflator. The absolute value of the z-acceleration is used.

EQ.0: inactive.

AMAG

Acceleration magnitude required to activate inflator.

EQ.0: inactive.

TDUR

Time duration acceleration must be exceeded before the inflator activates. This is the cumulative time from the beginning of the calculation, i.e., it is not continuous.

DVX

Velocity change in local x-direction to activate the inflator. (The absolute value of the velocity change is used.)

EQ.0: inactive.

DVY

Velocity change in local y-direction to activate the inflator. (The absolute value of the velocity change is used.)

EQ.0: inactive.

DVZ

Velocity change in local z-direction to activate the inflator. (The absolute value of the velocity change is used.)

EQ.0: inactive.

DVMAG

Velocity change magnitude required to activate the inflator.

EQ.0: inactive.

UX

Displacement increment in local x-direction to activate the inflator. (The absolute value of the x-displacement is used.)

EQ.0: inactive.

VARIABLE	DESCRIPTION
UY	Displacement increment in local y-direction to activate the inflator. (The absolute value of the y-displacement is used.) EQ.0: inactive.
UZ	Displacement increment in local z-direction to activate the inflator. (The absolute value of the z-displacement is used.) EQ.0: inactive.
UMAG	Displacement magnitude required to activate the inflator. EQ.0: inactive.

AIRBAG_SIMPLE_PRESSURE_VOLUME_*OPTION

Additional card for SIMPLE_PRESSURE_VOLUME option. (For card 1 see the "core cards" section of *AIRBAG.)

Card 2	1	2	3	4	5	6	7	8
Variable	CN	BETA	LCID	LCIDDR				
Type	F	F	I	I				
Default	none	none	none	0				

VARIABLE	DESCRIPTION
CN	Coefficient. Define if the load curve ID, LCID, is unspecified. LT.0.0: |CN| is the load curve ID, which defines the coefficient as a function of time.
BETA	Scale factor, β. Define if a load curve ID is not specified.
LCID	Optional load curve ID defining pressure versus relative volume.
LCIDDR	Optional load curve ID defining the coefficient, CN, as a function of time during the dynamic relaxation phase.

Remarks:

The relationship is the following:

$$\text{Pressure} = \frac{\beta \times \text{CN}}{\text{Relative Volume}}$$

$$\text{Relative Volume} = \frac{\text{Current Volume}}{\text{Initial Volume}}$$

The pressure is then a function of the ratio of current volume to the initial volume. The constant, CN, is used to establish a relationship known from the literature. The scale factor β is simply used to scale the given values. This simple model can be used when an initial pressure is given and no leakage, no temperature, and no input mass flow is assumed. A typical application is the modeling of air in automobile tires.

The load curve, LCIDDR, can be used to ramp up the pressure during the dynamic relaxation phase in order to avoid oscillations after the desired gas pressure is reached.

In the DEFINE_CURVE section this load curve must be flagged for dynamic relaxation. After initialization either the constant or load curve ID, |CN| is used to determine the pressure.

AIRBAG_SIMPLE_AIRBAG_MODEL_*OPTION

Additional cards for SIMPLE_AIRBAG_MODEL option. (For card 1 see the "core cards" section of *AIRBAG.)

Card 2	1	2	3	4	5	6	7	8
Variable	CV	CP	T	LCID	MU	AREA	PE	RO
Type	F	F	F	I	F	F	F	F
Default	none	none	none	none	none	none	none	none

Card 3	1	2	3	4	5	6	7	8
Variable	LOU	T_EXT	A	B	MW	GASC		
Type	I	F	F	F	F	F		
Default	0	0.	0.	0.	0.	0.		
Remarks	0	optional	optional	optional	optional	optional		

VARIABLE	DESCRIPTION
CV	Heat capacity at constant volume, e.g., Joules/kg/°K.
CP	Heat capacity at constant pressure, e.g., Joules/kg/°K.
T	Temperature of input gas
LCID	Load curve ID specifying input mass flow rate. See *DEFINE_CURVE.
MU	Shape factor for exit hole, μ: LT.0.0: $\|\mu\|$ is the load curve number defining the shape factor as a function of absolute pressure.

VARIABLE	DESCRIPTION
AREA	Exit area, A: GE.0.0: A is the exit area and is constant in time, LT.0.0: |A| is the load curve number defining the exit area as a function of absolute pressure.
PE	Ambient pressure, p_e
RO	Ambient density, ρ
LOU	Optional load curve ID giving mass flow out versus gauge pressure in bag. See *DEFINE_CURVE.

Leave the following 5 fields blank blank if CV ≠ 0

T_EXT	Ambient temperature.
A	First heat capacity coefficient of inflator gas (e.g., Joules/mole/°K).
B	Second heat capacity coefficient of inflator gas, (e.g., Joules/mole/°K^2).
MW	Molecular weight of inflator gas (e.g., Kg/mole).
GASC	Universal gas constant of inflator gas (e.g., 8.314 Joules/mole/°K).

Remarks:

The gamma law equation of state used to determine the pressure in the airbag:

$$p = (\gamma - 1)\rho e$$

where p is the pressure, ρ is the density, e is the specific internal energy of the gas, and γ is the ratio of the specific heats:

$$\gamma = \frac{c_p}{c_v}$$

From conservation of mass, the time rate of change of mass flowing into the bag is given as:

$$\frac{dM}{dt} = \frac{dM_{\text{in}}}{dt} - \frac{dM_{\text{out}}}{dt}$$

The inflow mass flow rate is given by the load curve ID, LCID. Leakage, the mass flow rate out of the bag, can be modeled in two alternative ways. One is to give an exit area with the corresponding shape factor, then the load curve ID, LOU, must be set to zero. The other is to define a mass flow out by a load curve, then μ and A have to both be set to zero.

If CV = 0. then the constant-pressure specific heat is given by:

$$c_p = \frac{(a + bT)}{\mathrm{MW}}$$

and the constant-volume specific heat is then found from:

$$c_v = c_p - \frac{R}{\mathrm{MW}}$$

AIRBAG_ADIABATIC_GAS_MODEL_*OPTION

Additional card for ADIABATIC_GAS_MODEL option. (For card 1 see the "core cards" section of *AIRBAG.)

Card 2	1	2	3	4	5	6	7	8
Variable	PSF	LCID	GAMMA	P0	PE	RO		
Type	F	I	F	F	F	F		
Default	1.0	none	none	none	none	none		

VARIABLE	DESCRIPTION
PSF	Pressure scale factor
LCID	Optional load curve for preload flag. See *DEFINE_CURVE.
GAMMA	Ratio of specific heats
P0	Initial pressure (gauge)
PE	Ambient pressure
RO	Initial density of gas

Remarks:

The optional load curve ID, LCID, defines a preload flag. During the preload phase the function value of the load curve versus time is zero, and the pressure in the control volume is given as:

$$p = \mathrm{PSF} \times p_0$$

When the **first nonzero** function value is encountered, the preload phase stops and the ideal gas law applies for the rest of the analysis. If LCID is zero, no preload is performed.

The gamma law equation of state for the adiabatic expansion of an ideal gas is used to determine the pressure after preload:

$$p = (\gamma - 1)\rho e$$

where p is the pressure, ρ is the density, e is the specific internal energy of the gas, and γ is the ratio of the specific heats:

$$\gamma = \frac{c_p}{c_v}$$

The pressure above is the absolute pressure, the resultant pressure acting on the control volume is:

$$p_s = \text{PSF} \times (p - p_e)$$

where PSF is the pressure scale factor. Starting from the initial pressure p_0 an initial internal energy is calculated:

$$e_0 = \frac{p_0 + p_e}{\rho(\gamma - 1)}$$

AIRBAG_WANG_NEFSKE_*OPTIONS

The following sequence of cards is read in for the all variations of the WANG_NEFSKE option to *AIRBAG. For card 1 see the "core cards" section of *AIRBAG.

Card 2	1	2	3	4	5	6	7	8
Variable	CV	CP	T	LCT	LCMT	TVOL	LCDT	IABT
Type	F	F	F	I	I	F	I	F
Default	none	none	0.	0	none	0.	0.	not used

Card 3	1	2	3	4	5	6	7	8
Variable	C23	LCC23	A23	LCA23	CP23	LCCP23	AP23	LCAP23
Type	F	I	F	I	F	I	F	I
Default	none	0	none	0	none	0	0.0	0

Card 4	1	2	3	4	5	6	7	8
Variable	PE	RO	GC	LCEFR	POVER	PPOP	OPT	KNKDN
Type	F	F	F	I	F	F	F	I
Default	none	none	none	0	0.0	0.0	0.0	0

Inflator Card. If the inflator is modeled, LCMT = 0 fill in the following card. *If not, include but leave blank.*

Card 5	1	2	3	4	5	6	7	8
Variable	IOC	IOA	IVOL	IRO	IT	LCBF		
Type	F	F	F	F	F	I		
Default	none	none	none	none	none	none		

Temperature Dependent Heat Capacities Card. Include this card when CV = 0.

Card 6	1	2	3	4	5	6	7	8
Variable	TEXT	A	B	MW	GASC	HCONV		
Type	F	F	F	F	F	F		
Default	none	none	none	none	none	none		

Criteria for Initiating Exit Flow Card. Additional card for the POP option to the *AIRBAG_WANG_NEFSKE card.

Card 7	1	2	3	4	5	6	7	8
Variable	TDP	AXP	AYP	AZP	AMAGP	TDURP	TDA	RBIDP
Type	F	F	F	F	F	F	F	I
Default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	none

VARIABLE	DESCRIPTION
CV	Specific heat at constant volume, e.g., Joules/kg/°K.
CP	Specific heat at constant pressure, e.g., Joules/kg/°K.
T	Temperature of input gas. For temperature variations a load curve, LCT, may be defined.

VARIABLE	DESCRIPTION
LCT	Optional load curve number defining temperature of input gas versus time. This overrides columns T.
LCMT	Load curve specifying input mass flow rate or tank pressure versus time. If the tank volume, TVOL, is nonzero the curve ID is assumed to be tank pressure versus time. If LCMT = 0, then the inflator has to be modeled, see Card 5. During the dynamic relaxation phase the airbag is ignored unless the curve is flagged to act during dynamic relaxation.
TVOL	Tank volume which is required only for the tank pressure versus time curve, LCMT.
LCDT	Load curve for time rate of change of temperature (dT/dt) versus time.
IABT	Initial airbag temperature. (Optional, generally not defined.)
C23	Vent orifice coefficient which applies to exit hole. Set to zero if LCC23 is defined below.
LCC23	The absolute value, |LCC23|, is a load curve ID. If the ID is positive, the load curve defines the vent orifice coefficient which applies to exit hole as a function of time. If the ID is negative, the vent orifice coefficient is defined as a function of relative pressure, $P_{\text{air}}/P_{\text{bag}}$, see [Anagonye and Wang 1999]. In addition, LCC23 can be defined through *DEFINE_CURVE_FUNCTION. A nonzero value for C23 overrides LCC23.
A23	If defined as a positive number, A23 is the vent orifice area which applies to exit hole. If defined as a negative number, the absolute value |A23| is a part ID, see [Anagonye and Wang, 1999]. The area of this part becomes the vent orifice area. Airbag pressure will not be applied to part |A23| representing venting holes if part |A23| is not included in SID, the part set representing the airbag. Set A23 to zero if LCA23 is defined below.
LCA23	Load curve number defining the vent orifice area which applies to exit hole as a function of absolute pressure, or LCA23 can be defined through *DEFINE_CURVE_FUNCTION. A nonzero value for A23 overrides LCA23.
CP23	Orifice coefficient for leakage (fabric porosity). Set to zero if LCCP23 is defined below.

VARIABLE DESCRIPTION

LCCP23

Load curve number defining the orifice coefficient for leakage (fabric porosity) as a function of time, or LCCP23 can be defined through *DEFINE_CURVE_FUNCTION. A nonzero value for CP23 overrides LCCP23.

AP23

Area for leakage (fabric porosity)

LCAP23

Load curve number defining the area for leakage (fabric porosity) as a function of (absolute) pressure, or LCAP23 can be defined through *DEFINE_CURVE_FUNCTION. A nonzero value for AP23 overrides LCAP23.

PE

Ambient pressure

RO

Ambient density

GC

Gravitational conversion constant (mandatory - no default). If consistent units are being used for all parameters in the airbag definition then unity should be input.

LCEFR

Optional curve for exit flow rate (mass/time) versus (gauge) pressure

POVER

Initial relative overpressure (gauge), P_{over} in control volume

PPOP

Pop Pressure: relative pressure (gauge) for initiating exit flow, P_{pop}

OPT

Fabric venting option, if nonzero CP23, LCCP23, AP23, and LCAP23 are set to zero.

EQ.1: Wang-Nefske formulas for venting through an orifice are used. Blockage is not considered.

EQ.2: Wang-Nefske formulas for venting through an orifice are used. Blockage of venting area due to contact is considered.

EQ.3: Leakage formulas of Graefe, Krummheuer, and Siejak [1990] are used. Blockage is not considered.

EQ.4: Leakage formulas of Graefe, Krummheuer, and Siejak [1990] are used. Blockage of venting area due to contact is considered.

EQ.5: Leakage formulas based on flow through a porous media are used. Blockage is not considered.

VARIABLE **DESCRIPTION**

EQ.6: Leakage formulas based on flow through a porous media are used. Blockage of venting area due to contact is considered.

EQ.7: Leakage is based on gas volume outflow versus pressure load curve. Blockage of flow area due to contact is not considered. Absolute pressure is used in the porous-velocity-versus-pressure load curve, given as FAC(P) in the *MAT_FABRIC card.

EQ.8: Leakage is based on gas volume outflow versus pressure load curve. Blockage of flow area due to contact is considered. Absolute pressure is used in the porous-velocity-versus-pressure load curve, given as FAC(P) in the *MAT_FABRIC card.

KNKDN Optional load curve ID defining the knock down pressure scale factor versus time. This option only applies to jetting. The scale factor defined by this load curve scales the pressure applied to airbag segments which

do not have a clear line-of-sight to the jet. Typically, at very early times this scale factor will be less than unity and equal to unity at later times. The full pressure is always applied to segments which can see the jets.

IOC Inflator orifice coefficient

IOA Inflator orifice area

IVOL Inflator volume

IRO Inflator density

IT Inflator temperature

LCBF Load curve defining burn fraction versus time

TEXT Ambient temperature.

A First molar heat capacity coefficient of inflator gas (e.g., Joules/mole/^{o}K)

B Second molar heat capacity coefficient of inflator gas, (e.g., Joules/mole/$^{o}K^{2}$)

VARIABLE	DESCRIPTION
MW	Molecular weight of inflator gas (e.g., Kg/mole).
GASC	Universal gas constant of inflator gas (e.g., 8.314 Joules/mole/^{O}K)
HCONV	Effective heat transfer coefficient between the gas in the air bag and the environment at temperature TEXT. If HCONV < 0, then HCONV defines a load curve of data pairs (time, hconv).
TDP	Time delay before initiating exit flow after pop pressure is reached.
AXP	Pop acceleration magnitude in local x-direction. EQ.0.0: Inactive.
AYP	Pop acceleration magnitude in local y-direction. EQ.0.0: Inactive.
AZP	Pop acceleration magnitude in local z-direction. EQ.0.0: Inactive.
AMAGP	Pop acceleration magnitude. EQ.0.0: Inactive.
TDURP	Time duration pop acceleration must be exceeded to initiate exit flow. This is a cumulative time from the beginning of the calculation, i.e., it is not continuous.
TDA	Time delay before initiating exit flow after pop acceleration is exceeded for the prescribed time duration.
RBIDP	Part ID of the rigid body for checking accelerations against pop accelerations.

Remarks:

The gamma law equation of state for the adiabatic expansion of an ideal gas is used to determine the pressure after preload:

$$p = (\gamma - 1)\rho e$$

where p is the pressure, ρ is the density, e is the specific internal energy of the gas, and γ is the ratio of the specific heats:

$$\gamma = \frac{c_p}{c_v}$$

where c_v is the specific heat at constant volume, and c_p is the specific heat at constant pressure. A pressure relation is defined:

$$Q = \frac{p_e}{p}$$

where p_e is the external pressure and p is the internal pressure in the bag. A critical pressure relationship is defined as:

$$Q_{\mathrm{crit}} = \left(\frac{2}{\gamma + 1}\right)^{\frac{\gamma}{\gamma-1}}$$

where γ is the ratio of specific heats:

$$\gamma = \frac{c_p}{c_v}$$

and

$$Q \le Q_{\mathrm{crit}} \Rightarrow Q = Q_{\mathrm{crit}}.$$

Wang and Nefske define the mass flow through the vents and leakage by

$$\dot{m}_{23} = C_{23}A_{23}\frac{p}{R\sqrt{T_2}}Q^{\frac{1}{\gamma}}\sqrt{2g_c\left(\frac{\gamma R}{\gamma - 1}\right)\left(1 - Q^{\frac{\gamma-1}{\gamma}}\right)}$$

and

$$\dot{m}'_{23} = C'_{23}A'_{23}\frac{p}{R\sqrt{T_2}}Q^{\frac{1}{\gamma}}\sqrt{2g_c\left(\frac{\gamma R}{\gamma - 1}\right)\left(1 - Q^{\frac{\gamma-1}{\gamma}}\right)}$$

It must be noted that the gravitational conversion constant has to be given in consistent units. As an alternative to computing the mass flow out of the bag by the Wang-Nefske model, a curve for the exit flow rate depending on the internal pressure can be taken. Then, no definitions for C23, LCC23, A23, LCA23, CP23, LCCP23, AP23, and LCAP23 are necessary.

The airbag inflator assumes that the control volume of the inflator is constant and that the amount of propellant reacted can be defined by the user as a tabulated curve of fraction reacted versus time. A pressure relation is defined:

$$Q_{\mathrm{crit}} = \frac{p_c}{p_i} = \left(\frac{2}{\gamma + 1}\right)^{\frac{\gamma}{\gamma-1}}$$

where p_c is a critical pressure at which sonic flow occurs, p_I, is the inflator pressure. The exhaust pressure is given by

$$p_e = \begin{cases} p_a & \text{if } p_a \ge p_c \\ p_c & \text{if } p_a < p_c \end{cases}$$

where p_a is the pressure in the control volume. The mass flow into the control volume is governed by the equation:

$$\dot{m}_{\text{in}} = C_0 A_0 \sqrt{2 p_I \rho_I} \sqrt{\frac{g_c \gamma \left(Q^{\frac{2}{\gamma}} - Q^{\frac{\gamma+1}{\gamma}} \right)}{\gamma - 1}}$$

where C_0, A_0, and ρ_I are the inflator orifice coefficient, area, and gas density, respectively.

If OPT is defined, then for OPT set to 1 or 2 the mass flow rate out of the bag, $\dot{m}_{out}$ is given by:

$$\dot{m}_{out} = \sqrt{g_c} \left\{ \sum_{n=1}^{\text{nairmats}} [\text{FLC}(t)_n \times \text{FAC}(p)_n \times \text{Area}_n] \right\} \sqrt{2 p \rho} \sqrt{\frac{\gamma \left(Q^{\frac{2}{k}} - Q^{\frac{\gamma+1}{\gamma}} \right)}{\gamma - 1}}$$

where, ρ is the density of airbag gas, "nairmats" is the number of fabrics used in the airbag, and "Area$_\text{n}$" is the current unblocked area of fabric number n.

If OPT set to 3 or 4 then:

$$\dot{m}_{\text{out}} = \left\{ \sum_{n=1}^{\text{nairmats}} [\text{FLC}(t)_n \times \text{FAC}(p)_n \times \text{Area}_n] \right\} \sqrt{2 (p - p_{\text{ext}}) \rho}$$

and for OPT set to 5 or 6:

$$\dot{m}_{\text{out}} = \left\{ \sum_{n=1}^{\text{nairmats}} [\text{FLC}(t)_n \times \text{FAC}(p)_n \times \text{Area}_n] \right\} (p - p_{\text{ext}})$$

and for OPT set to 7 or 8 (may be comparable to an equivalent model ALE model):

$$\dot{m}_{\text{out}} = \sum_{n=1}^{\text{nairmats}} \text{FLC}(t)_n \times \text{FAC}(p)_n \times \text{Area}_n \times \rho_n$$

Note that for different OPT settings, $\text{FAC}(p)_n$ has different meanings (all units shown just as demonstrations):

1. For OPT of 1, 2, 3 and 4, FAC(P) is unit-less.

2. For OPT of 5 and 6, FAC(P) has a unit of (s/m).

3. For OPT of 7 or 8, FAC(P) is the gas volume outflow through a unit area per unit time thus has the unit of speed,

4. $[\text{FAC}(P)] = \frac{[\text{volume}]}{[\text{area}][t]} = \frac{[\text{L}]^3}{[\text{L}]^2[t]} = \frac{[L]}{[t]} = [\text{velocity}].$

Multiple airbags may share the same part ID since the area summation is over the airbag segments whose corresponding part ID's are known. Currently, we assume that no more than ten materials are used per bag for purposes of the output. This constraint can be eliminated if necessary.

The total mass flow out will include the portion due to venting, i.e., constants C23 and A23 or their load curves above.

If CV = 0. then the constant-pressure specific heat is given by:

$$c_p = \frac{(a + bT)}{MW}$$

and the constant-volume specific heat is then found from:

$$c_v = c_p - \frac{R}{MW}$$

Two additional cards are required for JETTING models:

The following additional cards are defined for the WANG_NEFSKE_JETTING and WANG_NEFSKE_MULTIPLE_JETTING options, two further cards are defined for each option. The jet may be defined by specifying either the coordinates of the jet focal point, jet vector head and secondary jet focal point, or by specifying three nodes located at these positions. The nodal point option is recommended when the location of the airbag changes as a function of time.

NOTE: For Jetting models define either of the two cards below but not both.

Card format 8 for WANG_NEFSKE keyword option.

Card 8	1	2	3	4	5	6	7	8
Variable	XJFP	YJFP	ZJFP	XJVH	YJVH	ZJVH	CA	BETA
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	1.0
Remark	1	1	1	1	1	1		

Card format 8 for WANG_NEFSKE_MUTTIPLE_JETTING keyword options.

Card 8	1	2	3	4	5	6	7	8
Variable	XJFP	YJFP	ZJFP	XJVH	YJVH	ZJVH	LCJRV	BETA
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	1.0
Remark	1	1	1	1	1	1		

Card 9 for both WANG_NEFSKE_JETTING and WANG_NEFSKE_MULTIPLE_JETTING.

Card 9	1	2	3	4	5	6	7	8
Variable	XSJFP	YSJFP	ZSJFP	PSID	ANGLE	NODE1	NODE2	NODE3
Type	F	F	F	I	F	I	I	I
Default	none	none	none	none	none	0	0	0
Remark						1	1	1

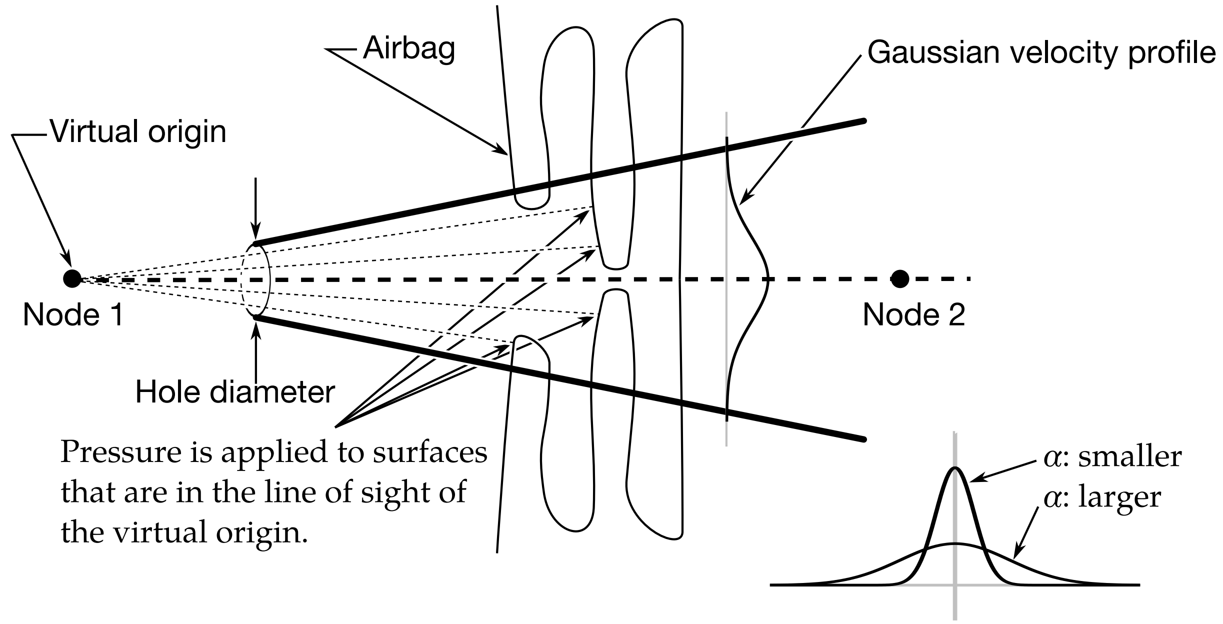

Airbag
Virtual origin
Gaussian velocity profile
Node 1
Node 2
Hole diameter
Pressure is applied to surfaces that are in the line of sight of the virtual origin.
α: smaller
α: larger

Figure 3-1. Jetting configuration for driver's side airbag (pressure applied only if centroid of surface is in line-of-sight)

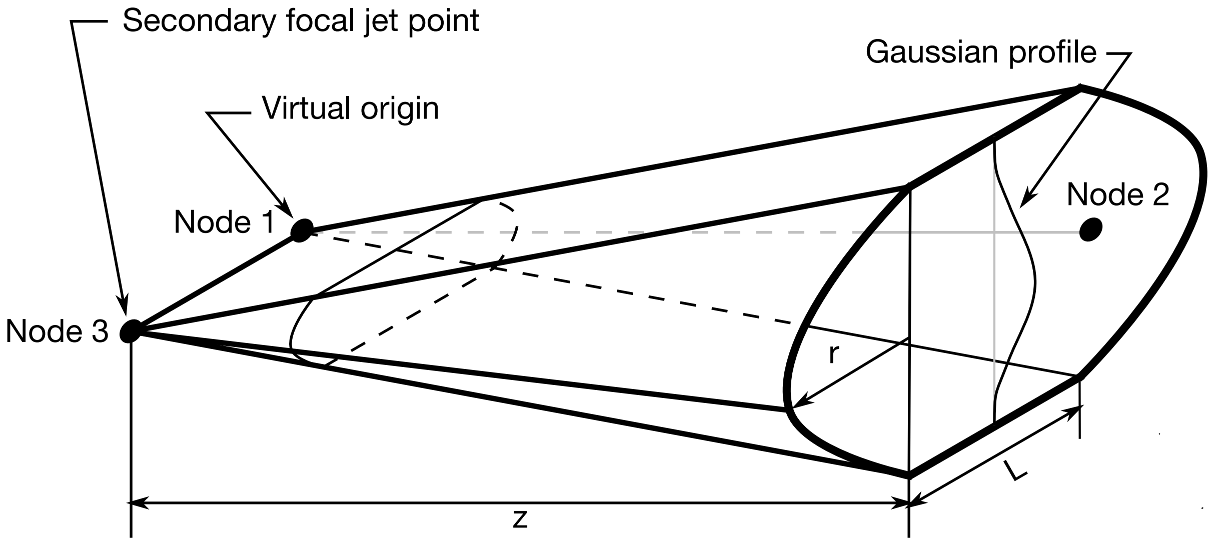


Figure 3-2. Jetting configuration for the passenger's side bag.

VARIABLE	DESCRIPTION
XJFP	x-coordinate of jet focal point, i.e., the virtual origin in Figures 3-1 and 3-2. See Remark 1 below.
YJFP	y-coordinate of jet focal point, i.e., the virtual origin in Figures 3-1 and 3-2.

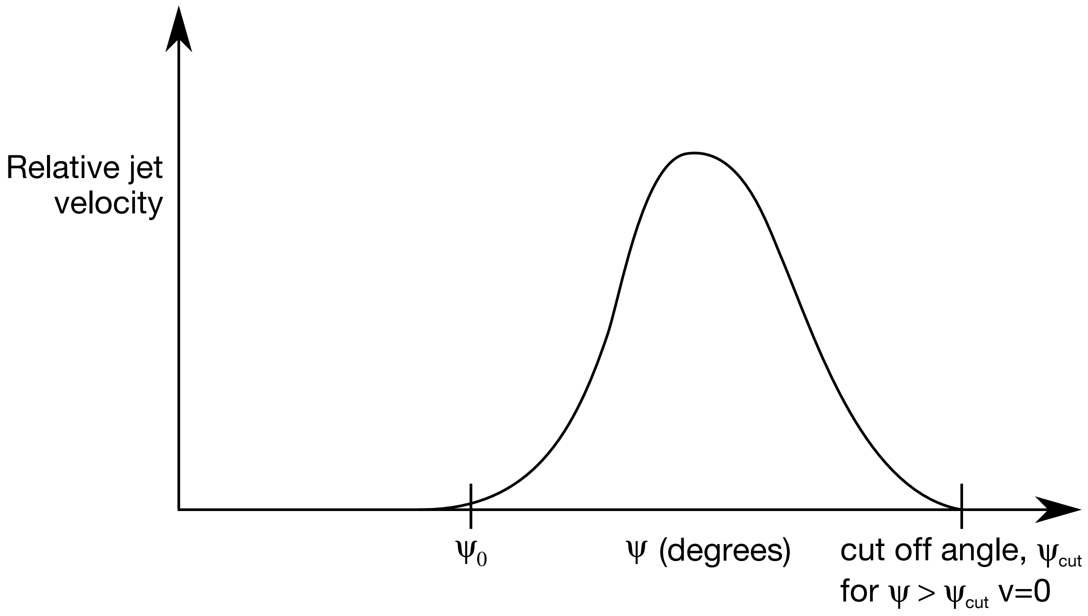


Figure 3-3. Normalized jet velocity versus angle for multiple jet driver's side airbag

VARIABLE **DESCRIPTION**

ZJFP
z-coordinate of jet focal point, i.e., the virtual origin in Figures 3-1 and 3-2.

XJVH
x-coordinate of jet vector head to defined code centerline

YJVH
y-coordinate of jet vector head to defined code centerline

ZJVH
z-coordinate of jet vector head to defined code centerline

CA
Cone angle, α, defined in radians.

LT.0.0: $|\alpha|$ is the load curve ID defining cone angle as a function of time

LCJRV
Load curve ID giving the spatial jet relative velocity distribution, see Figures 3-1, 3-2, and 3-3. The jet velocity is determined from the inflow mass rate and scaled by the load curve function value corresponding to the value of the angle ψ. Typically, the values on the load curve vary between 0 and unity. See *DEFINE_CURVE.

BETA
Efficiency factor, β, which scales the final value of pressure obtained from Bernoulli's equation.

LT.0.0: $|\beta|$ is the load curve ID defining the efficiency factor as a function of time

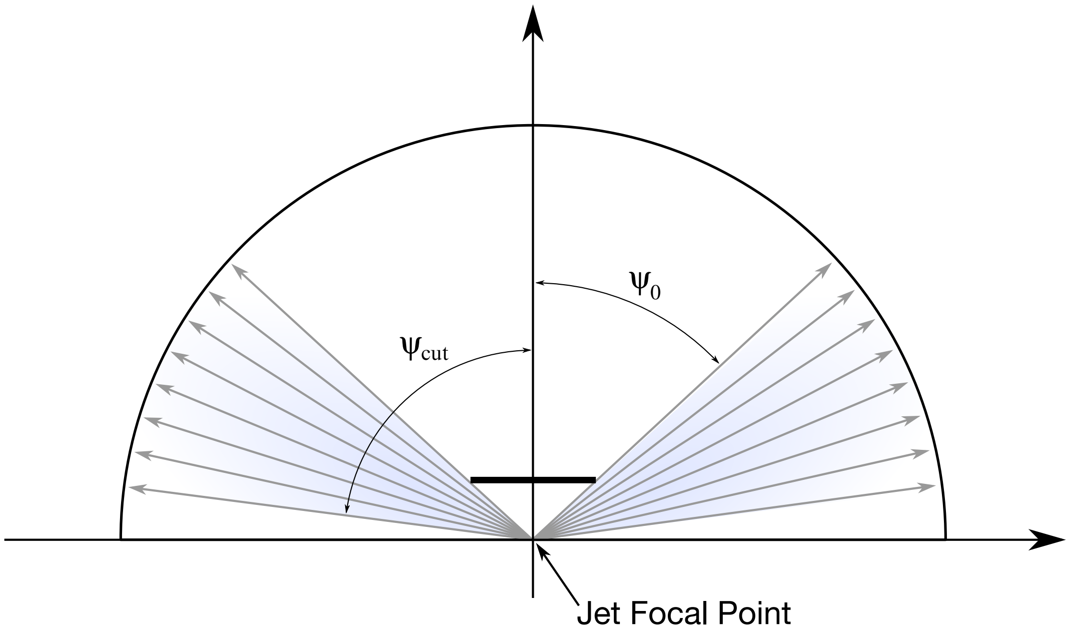


Figure 3-4. Multiple jet model for driver's side airbag. Typically, ψ_{cut} (see input ANGLE) is close to 90°. The angle ψ_0 is included to indicate that there is some angle below which the jet is negligible; see Figure 3-3.

VARIABLE	DESCRIPTION
XSJFP	x-coordinate of secondary jet focal point, passenger side bag. If the coordinates of the secondary point are (0,0,0) then a conical jet (driver’s side airbag) is assumed.
YSJFP	y-coordinate of secondary jet focal point
ZSJFP	z-coordinate of secondary jet focal point
PSID	Optional part set ID, see *SET_PART. If zero all elements are included in the airbag.
ANGLE	Cutoff angle in degrees. The relative jet velocity is set to zero for angles greater than the cutoff. See Figure 3-3. This option applies to the MULTIPLE jet only.
NODE1	Node ID located at the jet focal point, i.e., the virtual origin in Figures 3-1 and 3-2. See Remark 1 below.
NODE2	Node ID for node along the axis of the jet.
NODE3	Optional node ID located at secondary jet focal point.

Remarks:

1. It is assumed that the jet direction is defined by the coordinate method (XJFP, YJFP, ZJFP) and (XJVH, YJVH, ZJVH) unless both NODE1 and NODE2 are defined. In which case the coordinates of the nodes give by NODE1, NODE2 and NODE3 will override (XJFP, YJFP, ZJFP) and (XJVH, YJVH, ZJVH). The use of nodes is recommended if the airbag system is undergoing rigid body motion. The nodes should be attached to the vehicle to allow for the coordinates of the jet to be continuously updated with the motion of the vehicle.

2. The jetting option provides a simple model to simulate the real pressure distribution in the airbag during the breakout and early unfolding phase. Only the surfaces that are in the line of sight to the virtual origin have an increased pressure applied. With the optional load curve LCRJV, the pressure distribution with the code can be scaled according to the so-called relative jet velocity distribution.

3. For passenger side airbags the cone is replaced by a wedge type shape. The first and secondary jet focal points define the corners of the wedge and the angle α then defines the wedge angle.

4. Instead of applying pressure to all surfaces in the line of sight of the virtual origin(s), a part set can be defined to which the pressure is applied.

5. Care must be used to place the jet focal point within the bag. If the focal point is outside the bag, inside surfaces will not be visible so jetting pressure will not be applied correctly.

Additional card required for CM option:

The following additional card is defined for the WANG_NEFSKE_JETTING_CM and WANG_NEFSKE_MULTIPLE_JETTING_CM options.

Additional card required for CM keyword option.

Card 10	1	2	3	4	5	6	7	8
Variable	NREACT							
Type	I							
Default	none							
Remark								

VARIABLE	DESCRIPTION
NREACT	Node for reacting jet force. If zero the jet force will not be applied.

Remarks:

Compared with the standard LS-DYNA jetting formulation, the Constant Momentum option has several differences. Overall, the jetting usually has a more significant effect on airbag deployment than the standard LS-DYNA jetting: the total force is often greater, and does not reduce with distance from the jet.

The velocity at the jet outlet is assumed to be a choked (sonic) adiabatic flow of a perfect gas. Therefore the velocity at the outlet is given by:

$$v_{\text{outlet}} = \sqrt{\gamma RT} = \sqrt{\frac{(c_p - c_v)Tc_p}{c_v}}$$

The density in the nozzle is then calculated from conservation of mass flow.

$$\rho_0 v_{\text{outlet}} A_{\text{outlet}} = \dot{m}$$

This is different from the standard LS-DYNA jetting formulation, which assumes that the density of the gas in the jet is the same as atmospheric air, and then calculates the jet velocity from conservation of mass flow.

The velocity distribution at any radius, r, from the jet centerline and distance, z, from the focus, $v_{r,z}$relates to the velocity of the jet centerline, $v_r = 0, z$, in the same way as the standard LS-DYNA jetting options:

$$v_{r,z} = v_{r=0,z} e^{-\left(\frac{r}{\alpha z}\right)^2}$$

The velocity at the jet centerline, $v_r = 0$, at the distance, z, from the focus of the jet is calculated such that the momentum in the jet is conserved.

$$\text{momentum at nozzle} = \text{momentum at z}$$

$$\rho_0 v_{\text{outlet}}^2 A_{\text{outlet}} = \rho_0 \int v_{\text{jet}}^2 dA_{\text{jet}}$$

$$= \rho_0 v_{r=0,Z}^2 \left\{b + F\sqrt{b}\right\}$$

where, $b = \frac{\pi(\alpha z)^2}{2}$, and F is the distance between the jet foci (for a passenger jet).

Finally, the pressure exerted on an airbag element in view of the jet is given as:

$$p_{r,z} = \beta \rho_0 v_{r,z}^2$$

By combining the equations above

$$p_{r,z} = \frac{\beta \dot{m} v_{\text{outlet}} \left[e^{-(r/\alpha z)^2}\right]^2}{\left\{\frac{\pi(\alpha z)^2}{2} + F\sqrt{\frac{\pi(\alpha z)^2}{2}}\right\}}$$

The total force exerted by the jet is given by

$$F_{\text{jet}} = \dot{m} v_{\text{outlet}},$$

which is independent of the distance from the nozzle. Mass flow in the jet is not necessarily conserved, because gas is entrained into the jet from the surrounding volume. By contrast, the standard LS-DYNA jetting formulation conserves mass flow but not momentum. This has the effect of making the jet force reduce with distance from the nozzle.

The jetting forces can be reacted onto a node (NREACT), to allow the reaction force through the steering column or the support brackets to be modeled. The jetting force is written to the ASCII abstat file and the binary xtf file.

AIRBAG_LOAD_CURVE_*OPTION

Additional card required for LOAD_CURVE option. (For card 1 see the "core cards" section of *AIRBAG.)

Card 2	1	2	3	4	5	6	7	8
Variable	STIME	LCID	RO	PE	P0	T	T0	
Type	F	I	F	F	F	F	F	
Default	0.0	none	none	none	none	none	none	

VARIABLE	DESCRIPTION
STIME	Time at which pressure is applied. The load curve is offset by this amount.
LCID	Load curve ID defining pressure versus time, see *DEFINE_CURVE.
RO	Initial density of gas (ignored if LCID > 0)
PE	Ambient pressure (ignored if LCID > 0)
P0	Initial gauge pressure (ignored if LCID > 0)
T	Gas Temperature (ignored if LCID > 0)
T0	Absolute zero on temperature scale (ignored if LCID > 0)

Remarks:

Within this simple model the control volume is inflated with a pressure defined as a function of time or calculated using the following equation if LCID = 0.

$$P_{\mathrm{total}} = C\rho(T - T_0)$$

$$P_{\mathrm{gauge}} = P_{\mathrm{total}} - P_{\mathrm{ambient}}$$

The pressure is uniform throughout the control volume.

AIRBAG_LINEAR_FLUID_*OPTION

Additional card required for LINEAR_FLUID option. (For card 1 see the “core cards” section of *AIRBAG.)

Card 2	1	2	3	4	5	6	7	8
Variable	BULK	RO	LCINT	LCOUTT	LCOUTP	LCFIT	LCBULK	LCID
Type	F	F	I	I	I	I	I	I
Default	none	none	none	optional	optional	optional	optional	none

Card 3 is optional.

Card 3	1	2	3	4	5	6	7	8
Variable	P_LIMIT	P_LIMLC						
Type	F	I						
Default	optional	optional						

VARIABLE	DESCRIPTION
BULK	K, bulk modulus of the fluid in the control volume. Constant as a function of time. Define if LCBULK = 0.
RO	ρ, density of the fluid
LCINT	$F(t)$ input flow curve defining mass per unit time as a function of time, see *DEFINE_CURVE.
LCOUTT	$G(t)$, output flow curve defining mass per unit time as a function of time. This load curve is optional.
LCOUTP	$H(p)$, output flow curve defining mass per unit time as a function of pressure. This load curve is optional.
LFIT	$L(t)$, added pressure as a function of time. This load curve is optional.

VARIABLE	DESCRIPTION
LCBULK	Curve defining the bulk modulus as a function of time. This load curve is optional, but if defined, the constant, BULK, is not used.
LCID	Load curve ID defining pressure versus time, see *DEFINE_CURVE.
P_LIMIT	Limiting value on total pressure (optional).
P_LIMLC	Curve defining the limiting pressure value as a function of time. If nonzero, P_LIMIT is ignored.

Remarks:

If LCID = 0 then the pressure is determined from:

$$P(t) = K(t)\ln\left[\frac{V_0(t)}{V(t)}\right] + L(t).$$

where,

$$\begin{aligned}
P(t) &= \text{Pressure,}\\
V(t) &= \text{Volume of fluid in compressed state,}\\
V_{0(t)} &= V_0(t)\\
&= \frac{M(t)}{\rho}\\
&= \text{Volume of fluid in uncompressed state,}\\
M(t) &= M(0) + \int F(t)dt - \int G(t)dt - \int H(p)dt\\
&= \text{Current fluid mass,}\\
M(0) &= V(0)\rho\\
&= \text{Mass of fluid at time zero } P(0) = 0.
\end{aligned}$$

By setting LCID ≠ 0 a pressure time history may be specified for the control volume and the mass of fluid within the volume is then calculated from the volume and density.

This model is for the simulation of hydroforming processes or similar problems. The pressure is controlled by the mass flowing into the volume and by the current volume. The pressure is uniformly applied to the control volume.

Note the signs used in the equation for $M(t)$. The mass flow should always be defined as positive since the output flow is subtracted.

AIRBAG_HYBRID_*OPTIONS

AIRBAG_HYBRID_JETTING_*OPTIONS

Additional cards required for HYBRID and HYBRID_JETTING options. (For card 1 see the "core cards" section of *AIRBAG.)

Card 2	1	2	3	4	5	6	7	8
Variable	ATMOST	ATMOSP	ATMOSD	GC	CC	HCONV		
Type	F	F	F	F	F	F		
Default	none	none	none	none	1.0	none		

Card 3	1	2	3	4	5	6	7	8
Variable	C23	LCC23	A23	LCA23	CP23	LCP23	AP23	LCAP23
Type	F	I	F	I	F	I	F	I
Default	none	0	none	0	none	0	none	0

Card 4	1	2	3	4	5	6	7	8
Variable	OPT	PVENT	NGAS	LCEFR	LCIDM0	VNTOPT		
Type	I	F	I	I	I	I		
Default	none	none	none	0	0	0		

Include NGAS pairs of cards 5 and 6:

Card 5	1	2	3	4	5	6	7	8
Variable	LCIDM	LCIDT		MW	INITM	A	B	C
Type	I	I		F	F	F	F	F
Default	none	none		none	none	none	none	none

Card 6	1	2	3	4	5	6	7	8
Variable	FMASS							
Type	F							
Default	none							

VARIABLE	DESCRIPTION
ATMOST	Atmospheric temperature
ATMOSP	Atmospheric pressure
ATMOSD	Atmospheric density
GC	Universal molar gas constant
CC	Conversion constant EQ.0: Set to 1.0.
HCONV	Effective heat transfer coefficient between the gas in the air bag and the environment at temperature at ATMOST. If HCONV < 0, then HCONV defines a load curve of data pairs (time, hconv).
C23	Vent orifice coefficient which applies to exit hole. Set to zero if LCC23 is defined below.

VARIABLE	DESCRIPTION
LCC23	The absolute value, |LCC23|, is a load curve ID. If the ID is positive, the load curve defines the vent orifice coefficient which applies to exit hole as a function of time. If the ID is negative, the vent orifice coefficient is defined as a function of relative pressure, $P_{\mathrm{air}}/P_{\mathrm{bag}}$, see [Anagonye and Wang 1999]. In addition, LCC23 can be defined through *DEFINE_CURVE_FUNCTION. A nonzero value for C23 overrides LCC23
A23	If defined as a positive number, A23 is the vent orifice area which applies to exit hole. If defined as a negative number, the absolute value |A23| is a part ID, see [Anagonye and Wang 1999]. The area of this part becomes the vent orifice area. Airbag pressure will not be applied to part |A23| representing venting holes if part |A23| is not included in SID, the part set representing the airbag. Set A23 to zero if LCA23 is defined below.
LCA23	Load curve number defining the vent orifice area which applies to exit hole as a function of *absolute* pressure, or LCA23 can be defined through *DEFINE_CURVE_FUNCTION. A nonzero value for A23 overrides LCA23.
CP23	Orifice coefficient for leakage (fabric porosity). Set to zero if LCCP23 is defined below.
LCCP23	Load curve number defining the orifice coefficient for leakage (fabric porosity) as a function of time, or LCCP23 can be defined through *DEFINE_CURVE_FUNCTION. A nonzero value for CP23 overrides LCCP23.
AP23	Area for leakage (fabric porosity)
LCAP23	Load curve number defining the area for leakage (fabric porosity) as a function of (absolute) pressure, or LCAP23 can be defined through *DEFINE_CURVE_FUNCTION. A nonzero value for AP23 overrides LCAP23.

VARIABLE **DESCRIPTION**

OPT

Fabric venting option, if nonzero CP23, LCCP23, AP23, and LCAP23 are set to zero.

EQ.1: Wang-Nefske formulas for venting through an orifice are used. Blockage is not considered.

EQ.2: Wang-Nefske formulas for venting through an orifice are used. Blockage of venting area due to contact is considered.

EQ.3: Leakage formulas of Graefe, Krummheuer, and Siejak [1990] are used. Blockage is not considered.

EQ.4: Leakage formulas of Graefe, Krummheuer, and Siejak [1990] are used. Blockage of venting area due to contact is considered.

EQ.5: Leakage formulas based on flow through a porous media are used. Blockage due to contact is not considered.

EQ.6: Leakage formulas based on flow through a porous media are used. Blockage due to contact is considered.

EQ.7: Leakage is based on gas volume outflow versus pressure load curve. Blockage of flow area due to contact is not considered. Absolute pressure is used in the porous-velocity-versus-pressure load curve, given as FAC(P) in the *MAT_FABRIC card.

EQ.8: Leakage is based on gas volume outflow versus pressure load curve. Blockage of flow area due to contact is considered.

PVENT

Gauge pressure when venting begins

NGAS

Number of gas inputs to be defined below (Including initial air). The maximum number of gases is 17.

LCEFR

Optional curve for exit flow rate (mass/time) versus (gauge) pressure

LCIDM0

Optional curve representing inflator's total mass inflow rate. When defined, LCIDM in the following $2 \times$NGAS cards defines the molar fraction of each gas component as a function of time and INITM defines the initial molar ratio of each gas component.

VARIABLE DESCRIPTION

VNTOPT

Additional options for venting area definition.

For A23 ≥ 0

EQ.1: Vent area is equal to A23.

EQ.2: Vent area is A23 plus the eroded surface area of the airbag parts.

EQ.10: Same as VNTOPT = 2

For A23 < 0

EQ.1: Vent area is the increase in surface area of part |A23|. If there is no change in surface area of part |A23| over the initial surface area or if the surface area reduces from the initial area, there is no venting.

EQ.2: Vent area is the total (not change in) surface area of part |A23| plus the eroded surface area of all other parts comprising the airbag.

EQ.10: Vent area is the increase in surface area of part |A23| plus the eroded surface area of all other parts comprising the airbag.

LCIDM

Load curve ID for inflator mass flow rate (eq. 0 for gas in the bag at time = 0)

GT.0: piecewise linear interpolation

LT.0: cubic spline interpolation

LCIDT

Load curve ID for inflator gas temperature (eq.0 for gas in the bag at time 0)

GT.0: piecewise linear interpolation

LT.0: cubic spline interpolation

MW

Molecular weight

INITM

Initial mass fraction of gas component present in the airbag, prior to injection of gas by the inflator

A

Coefficient for molar heat capacity of inflator gas at constant pressure, (e.g., Joules/mole/^{o}K)

B

Coefficient for molar heat capacity of inflator gas at constant pressure, (e.g., Joules/mole/$^oK^2$)

VARIABLE	DESCRIPTION
C	Coefficient for molar heat capacity of inflator gas at constant pressure, (e.g., Joules/mole/°K^3)
FMASS	Fraction of additional aspirated mass.

Aditional cards are required for HYBRID_JETTING and HYBRID_JETTING_CM

The following two additional cards are defined for the HYBRID_JETTING options. The jet may be defined by specifying either the coordinates of the jet focal point, jet vector head and secondary jet focal point, or by specifying three nodes located at these positions. The nodal point option is recommended when the location of the airbag changes as a function of time.

Card 7	1	2	3	4	5	6	7	8
Variable	XJFP	YJFP	ZJFP	XJVH	YJVH	ZJVH	CA	BETA
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none
Remark	1	1	1	1	1	1		

Card 8	1	2	3	4	5	6	7	8
Variable	XSJFP	YSJFP	ZSJFP	PSID	IDUM	NODE1	NODE2	NODE3
Type	F	F	F	I	F	I	I	I
Default	none	none	none	none	none	0	0	0
Remark					2	1	1	1

Additional card required for HYBRID_JETTING_CM option.

Card 9	1	2	3	4	5	6	7	8
Variable	NREACT							
Type	I							
Default	none							
Remark	4							

VARIABLE	DESCRIPTION
XJFP	*x*-coordinate of jet focal point, i.e., the virtual origin in Figures 3-1 and 3-2. See Remark 1 below.
YJFP	*y*-coordinate of jet focal point, i.e., the virtual origin in Figures 3-1 and 3-2.
ZJFP	*z*-coordinate of jet focal point, i.e., the virtual origin in Figures 3-1 and 3-2.
XJVH	*x*-coordinate of jet vector head to defined code centerline
YJVH	*y*-coordinate of jet vector head to defined code centerline
ZJVH	*z*-coordinate of jet vector head to defined code centerline
CA	Cone angle, α, defined in radians. LT.0.0: $\|\alpha\|$ is the load curve ID defining cone angle as a function of time
BETA	Efficiency factor, β, which scales the final value of pressure obtained from Bernoulli's equation. LT.0.0: $\|\beta\|$ is the load curve ID defining the efficiency factor as a function of time
XSJFP	*x*-coordinate of secondary jet focal point, passenger side bag. If the coordinate of the secondary point is (0,0,0) then a conical jet (driver's side airbag) is assumed.
YSJFP	*y*-coordinate of secondary jet focal point

VARIABLE	DESCRIPTION
ZSJFP	*z*-coordinate of secondary jet focal point
PSID	Optional part set ID, see *SET_PART. If zero all elements are included in the airbag.
IDUM	Dummy field (Variable not used)
NODE1	Node ID located at the jet focal point, i.e., the virtual origin in Figure 3-7. See Remark 1 below.
NODE2	Node ID for node along the axis of the jet.
NODE3	Optional node ID located at secondary jet focal point.
NREACT	Node for reacting jet force. If zero the jet force will not be applied.

Remarks:

1. **Jetting.** It is assumed that the jet direction is defined by the coordinate method (XJFP, YJFP, ZJFP) and (XJVH, YJVH, ZJVH) unless both NODE1 and NODE2 are defined. In which case the coordinates of the nodes given by NODE1, NODE2 and NODE3 will override (XJFP, YJFP, ZJFP) and (XJVH, YJVH, ZJVH). The use of nodes is recommended if the airbag system is undergoing rigid body motion. The nodes should be attached to the vehicle to allow for the coordinates of the jet to be continuously updated with the motion of the vehicle.

   The jetting option provides a simple model to simulate the real pressure distribution in the airbag during the breakout and early unfolding phase. Only the surfaces that are in the line of sight to the virtual origin have an increased pressure applied. With the optional load curve LCRJV, the pressure distribution with the code can be scaled according to the so-called relative jet velocity distribution.

   For passenger side airbags the cone is replaced by a wedge type shape. The first and secondary jet focal points define the corners of the wedge and the angle α then defines the wedge angle.

   Instead of applying pressure to all surfaces in the line of sight of the virtual origin(s), a part set can be defined to which the pressure is applied.

2. **IDUM.** This variable is not used and has been included to maintain the same format as the WANG_NEFSKE_JETTING options.

3. **Focal Point Placement.** Care must be used to place the jet focal point within the bag. If the focal point is outside the bag, inside surfaces will not be visible so jetting pressure will not be applied correctly.

4. **NREACT.** See the description related to the WANG_NEFSKE_JETTING_CM option. For the hybrid inflator model the heat capacities are compute from the combination of gases which inflate the bag.

AIRBAG_HYBRID_CHEMKIN_*OPTION

The HYBRID_CHEMKIN model includes 3 control cards. For each gas species an additional set of cards must follow consisting of a control card and several thermodynamic property data cards. (For card 1 see the "core cards" section of *AIRBAG.)

Card 2	1	2	3	4	5	6	7	8
Variable	LCIDM	LCIDT	NGAS	DATA	ATMT	ATMP	RG	
Type	I	I	I	I	F	F	F	
Default	none	none	none	none	none	none	none	

Card 3	1	2	3	4	5	6	7	8
Variable	HCONV							
Type	F							
Default	0.							

Card 4	1	2	3	4	5	6	7	8
Variable	C23	A23						
Type	F	F						
Default	0.	0.						

VARIABLE	DESCRIPTION
LCIDM	Load curve specifying input mass flow rate versus time. GT.0: piece wise linear interpolation LT.0: cubic spline interpolation

VARIABLE DESCRIPTION

LCIDT Load curve specifying input gas temperature versus time.

GT.0: piece wise linear interpolation

LT.0: cubic spline interpolation

NGAS Number of gas inputs to be defined below. (Including initial air)

DATA Thermodynamic database

EQ.1: NIST database (3 additional property cards are required below)

EQ.2: CHEMKIN database (no additional property cards are required)

EQ.3: Polynomial data (1 additional property card is required below)

ATMT Atmospheric temperature.

ATMP Atmospheric pressure

RG Universal gas constant

HCONV Effective heat transfer coefficient between the gas in the air bag and the environment at temperature ATMT. If HCONV < 0, then HCONV defines a load curve of data pairs (time, hconv).

C23 Vent orifice coefficient

A23 Vent orifice area

NGAS Sets of Gas Species Data Cards:

For each gas species include a set of cards consisting of a Gas Species Control Card followed by several thermo-dynamic property data cards whose format depends on the DATA parameter on card in format “card 5”. The next "*" card terminates the reading of this data.

Gas Species Control Card.

Card 5	1	2	3	4	5	6	7	8
Variable	CHNAME	MW	LCIDN	FMOLE	FMOLET			
Type	A	F	I	F	F			
Default	none	none	0	none	0.			

VARIABLE	DESCRIPTION
CHNAME	Chemical symbol for this gas species (e.g., N2 for nitrogen, AR for argon). Required for DATA = 2 (CHEMKIN), optional for DATA = 1 or DATA = 3.
MW	Molecular weight of this gas species.
LCIDN	Load curve specifying the input mole fraction versus time for this gas species. If > 0, FMOLE is not used.
FMOLE	Mole fraction of this gas species in the inlet stream.
FMOLET	Initial mole fraction of this gas species in the tank.

Additional thermodynamic data cards for each gas species.

If DATA = 1, include the following 3 cards for the NIST database:

The required data can be found on the NIST web site at http://webbook.nist.gov/chemistry/.

Card 5a	1	2	3	4	5	6	7	8
Variable	TLOW	TMID	THIGH					
Type	F	F	F					
Default	none	none	none					

Card 5b	1	2	3	4	5	6	7	8
Variable	alow	blow	clow	dlow	elow	flow	hlow	
Type	F	F	F	F	F	F	F	
Default	none	none	none	none	none	none	none	

Card 5c	1	2	3	4	5	6	7	8
Variable	ahigh	bhigh	chigh	dhigh	ehigh	fhigh	hhigh	
Type	F	F	F	F	F	F	F	
Default	none	none	none	none	none	none	none	

VARIABLE	DESCRIPTION
TLOW	Curve fit low temperature limit.
TMID	Curve fit low-to-high transition temperature.
THIGH	Curve fit high temperature limit.

VARIABLE	DESCRIPTION
alow, …, hlow	Low temperature range NIST polynomial curve fit coefficients (see below).
ahigh, …, hhigh	High temperature range NIST polynomial curve fit coefficients (see below).

No additional cards are needed if using the CHEMKIN database (DATA = 2):

Polynomial Fit Card (DATA = 3).

Card 5d	1	2	3	4	5	6	7	8
Variable	a	b	c	d	e			
Type	F	F	F	F	F			
Default	none	0.	0.	0.	0.			

VARIABLE	DESCRIPTION
a	Coefficient, see below.
b	Coefficient, see below.
c	Coefficient, see below.
d	Coefficient, see below.
e	Coefficient, see below.

Specific heat curve fits:

$$\begin{aligned} \text{NIST:} \quad & c_p = \frac{1}{M}\left(a + bT + cT^2 + dT^3 + \frac{e}{T^2}\right) \\ \text{CHEMKIN:} \quad & c_p = \frac{\overline{R}}{M}(a + bT + cT^2 + dT^3 + eT^4) \\ \text{POLYNOMIAL:} \quad & c_p = \frac{1}{M}(a + bT + cT^2 + dT^3 + eT^4) \end{aligned}$$

where,

$$\overline{R} = \text{universal gas constant } 8.314\frac{\text{Nm}}{\text{mole} \times K}$$

M = gas molecular weight

AIRBAG_FLUID_AND_GAS_*OPTIONS

Additional cards required for FLUID_AND_GAS option. (For card 1 see the "core cards" section of *AIRBAG.) Currently this option only works in SMP and explicit analysis.

Card 2	1	2	3	4	5	6	7	8
Variable	XWINI	XWADD	XW	P	TEND	RHO	LCXW	LCP
Type	F	F	F	F	F	F	I	I
Default	none	none	none	none	none	none	none	none

Card 3	1	2	3	4	5	6	7	8
Variable	GDIR	NPROJ	IDIR	IIDIR	KAPPA	KBM		
Type	F	I	I	I	F	F		
Default	none	3	none	none	1.0	none		

VARIABLE	DESCRIPTION
XWINI	Fluid level at time $t = 0$ in $\vert\text{GDIR}\vert$ direction.
XWADD	Fluid level filling increment per time step.
XW	Final fluid level in filling process.
P	Gas pressure at time t = TEND.
TEND	Time when gas pressure P is reached.
RHO	Density of the fluid (e.g. for water, RHO ≈ 1.0 kg/m^3)
LCXW	Load curve ID for fluid level vs. time. XW, XWADD, and XWINI are with this option.
LCP	Load curve ID for gas pressure vs. time. P and TEND are ignored with this option.

VARIABLE	DESCRIPTION
GDIR	Global direction of gravity (e.g. -3.0 for negative global z-axis). EQ.1.0: global x-direction, EQ.2.0: global y-direction, EQ.3.0: global z-direction.
NPROJ	Number of projection directions (only global axis) for volume calculation.
IDIR	First direction of projection (if $\lvert\text{NPROJ}\rvert \neq 3$), only global axis.
IIDIR	Second direction of projection (if $\lvert\text{NPROJ}\rvert = 2$), only global axis.
KAPPA	Adiabatic exponent
KBM	Bulk modulus of the fluid (e.g. for water, BKM ≈ 2080 N/mm^2)

Remarks:

The *AIRBAG_FLUID_AND_GAS option models a quasi-static multi chamber fluid/gas structure interaction in a simplified way including three possible load cases: (i) only gas, (ii) only incompressible fluid, or (iii) the combination of incompressible fluid with additional gas "above". see Figure 3-5.

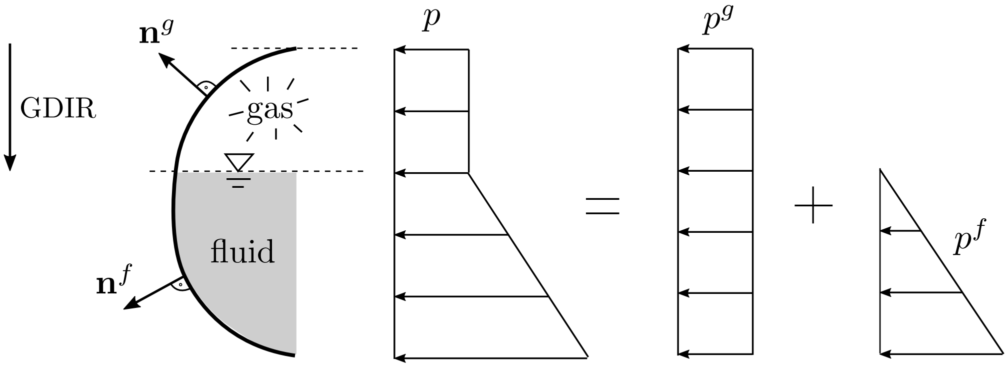


Figure 3-5. Hydrostatic pressure distribution in a chamber filled with gas and incompressible fluid

The theory is based on the description of gases and fluids as energetically equivalent pressure loads. The calculation of the fluid volume is carried out using the directions of projection and a non-normalized normal vector. This model, therefore, requires that the normal of the shell elements belonging to a filled structure must point outwards. Holes are not detected, but can be taken into account as described below.

In case of a pure gas (no fluid), the *AIRBAG_SIMPLE_PRESSURE_VOLUME and *AIRBAG_FLUID_AND_GAS cards give identical results as they are based on the same theory. The update of the gas pressure due to volume change is calculated with the following simple gas law

$$p^g = \left(1 - \mathrm{KAPPA} \times \frac{v^g - v^g_{\mathrm{old}}}{v^g_{\mathrm{old}}}\right) p^g_{\mathrm{old}}$$

with adiabatic exponent KAPPA and gas volume v^g.

The theory of incompressible fluids is based on the variation of the potential energy and an update of the water level due to changes in the volume and the water surface, see Haßler and Schweizerhof [2007], Haßler and Schweizerhof [2008], Rumpel and Schweizerhof [2003], and Rumpel and Schweizerhof [2004].

In case of multiple fluid/gas filled chambers each chamber requires an additional *AIRBAG_FLUID_AND_GAS card. Some of the parameters which are called local parameters only belong to a single chamber (e.g. gas pressure). In contrast global, parameters belong to all chambers (e.g. direction of gravitation).

Because the theory only applies to quasi-static fluid-structure interaction the load has to be applied slowly so that the kinetic energy is almost zero throughout the process.

All parameters of card 1 are local parameters describing the filling of the chamber. The water level and the gas pressure can be defined by curves using LCXW and LCP. A second possibility are the parameters XWINI, XW, XWADD, P and TEND. When describing the fluid and gas filling using the parameters the gas pressure at time $t = 0$ is set to 0 and the initial water level is set to XWINI in |GDIR|-direction. At each timestep, XWADD is added to the water level, until XW is reached. The gas pressure will be raised until P is reached at time $t = \mathrm{TEND}$.

In general, global parameters belong to all chambers. To describe the global axis in GDIR, NPROJ, IDIR and IIDIR the following relations apply: x-axis is axis “1”, the y-axis is axis “2”, and the z-axis is axis 3.

The gas and fluid volume is calculated by contour integrals in the global x-, y- and z-coordinates. If one of the boundaries is discontinuous in one or two global directions, these directions have to be ignored in NPROJ, IDIR and IIDIR. At least one direction of projection must be set (NPROJ = 1, IDIR = value), but it is recommended to use as many directions of projection as possible.

In case of a structure filled exclusively with fluid, IDIR and IIDIR should not be set to |GDIR|. In case of holes in a structure (e.g. to take advantage of symmetry planes), IDIR and IIDIR should not be set to the normal direction of the plane describing the hole or symmetry plane.

An example of a water filled tube structure illustrating how to use NPROJ, IDIR, IIDIR, and GDIR is shown in Figure 3-6. In this example gravity is acting opposite to the global *z*-axis. In this case, then, GDIR = -3. The structure is filled exclusively with water, so the projection direction cannot be set to |GDIR| = 3. To use the symmetry of the tube only half of the structure has been modeled. The normal of the symmetry plane shows in *y* direction, so the projection direction cannot be set to 2. Because the symmetry axes (2 and 3) are not allowed, the only direction of projection is 1; therefore, NPROJ = 1 and IDIR = 1.

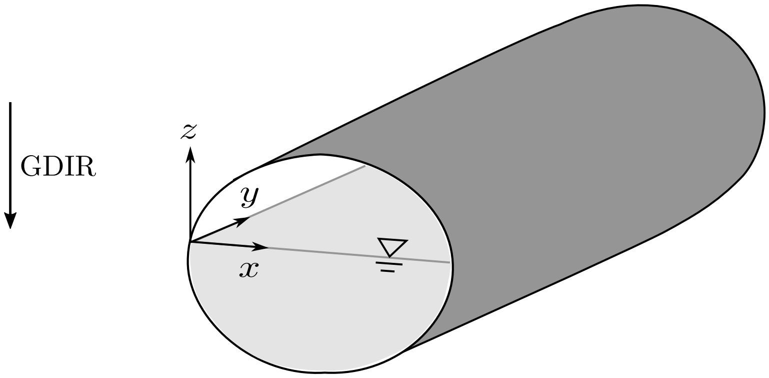


Figure 3-6. Example for water filled tube structure

For further explanations and examples see Haßler and Schweizerhof [2007], Haßler and Schweizerhof [2008], and Maurer, Gebhardt, and Schweizerhof [2010].

The possible entries for NPROJ, IDIR and IIDIR are:

NPROJ	IDIR	IIDIR
3		
2	1	2
2	1	3
2	2	3
1	1	
1	2	
1	3	

*AIRBAG_ALE

Purpose: The input in this section provides a simplified approach to defining the deployment of the airbag using the ALE capabilities with an option to switch from the initial ALE method to control volume (CV) method (*AIRBAG_HYBRID) at a chosen time. An enclosed airbag (and possibly the airbag canister/compartment and/or a simple representation of the inflator) shell structure interacts with the inflator gas(es). This definition provides a single fluid to structure coupling for the airbag-gas interaction during deployment in which the CV input data may be used directly.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	SIDTYP					MWD	SPSF
Type	I	I					F	F
Default	none	none					0	0
Remark	1							

VARIABLE	DESCRIPTION
SID	Set ID as defined on *AIRBAG card. This set ID contains the Lagrangian elements (segments) which make up the airbag and possibly the airbag canister/compartment and/or a simple representation of the inflator. See Remark 1.
SIDTYP	Set type: EQ.0: Segment set. EQ.1: Part set.
MWD	Mass weighted damping factor, D. This is used during the CV phase for *AIRBAG_HYBRID.
SPSF	Stagnation pressure scale factor, $0 \leq \gamma \leq 1$. This is used during the CV phase for *AIRBAG_HYBRID.

Ambient Environment Card.

Card 2	1	2	3	4	5	6	7	8
Variable	ATMOST	ATMOSP		GC	CC	TNKVOL	TNKFINP	
Type	F	F		F	F	F	F	
Default	0.	0.		none	1.0	0.0	0.0	
Remark	2	2				10	10	

VARIABLE **DESCRIPTION**

ATMOST Atmospheric ambient temperature. See Remark 2.

ATMOSP Atmospheric ambient pressure. See Remark 2.

GC Universal molar gas constant.

CC Conversion constant. If EQ: .0 Set to 1.0.

TNKVOL Tank volume from the inflator tank test or Inflator canister volume. See remark 10.

LCVEL = 0 and TNKFINP is defined:

TNKVOL is the defined Tank. Inlet gas velocity is estimated by LS-DYNA method (testing).

LCVEL = 0 and TNKFINP is not defined

TNKVOL is the estimated inflator canister volume Inlet gas velocity is estimated automatically by the Lian-Bhalsod-Olovsson method.

LCVEL ≠ 0

This must be left blank.

TNKFINP Tank final pressure from the inflator tank test data. Only define this parameter for option 1 of TNKVOL definition above. See Remark 10.

Coupling Card. See keyword *CONSTRAINED_LAGRANGE_IN_SOLID.

Card 3	1	2	3	4	5	6	7	8
Variable	NQUAD	CTYPE	PFAC	FRIC	FRCMIN	NORMTYP	ILEAK	PLEAK
Type	I	I	F	F	F	I	I	F
Default	4	4	0.1	0.0	0.3	0	2	0.1
Remark	13	13	14					

VARIABLE **DESCRIPTION**

NQUAD

Number of (quadrature) coupling points for coupling Lagrangian slave parts to ALE master solid parts. If NQUAD = n, then nXn coupling points will be parametrically distributed over the surface of each Lagrangian slave segment (default = 4). See Remark 13.

CTYPE

Coupling type (default = 4, see Remark 13):

EQ.4: (default) penalty coupling with DIREC = 2 implied.

EQ.6: penalty coupling in which DIREC is automatically set to DIREC = 1 for the unfolded region and DIREC = 2 for folded region.

PFAC

Penalty factor. PFAC is a scale factor for scaling the estimated stiffness of the interacting (coupling) system. It is used to compute the coupling forces to be distributed on the slave and master parts.

GT.0: Fraction of estimated critical stiffness (default = 0.1).

LT.0: -PFAC is a load curve ID. The curve defines the relative coupling pressure (y-axis) as a function of the tolerable fluid penetration distance (x-axis).

FRIC

Coupling coefficient of friction.

FRCMIN

Minimum fluid volume fraction in an ALE element to activate coupling (default is 0.3).

VARIABLE	DESCRIPTION
NORMTYP	Penalty coupling spring direction (DIREC 1 and 2): EQ.0: normal vectors are interpolated from nodal normals (default) EQ.1: normal vectors are interpolated from segment normals.
ILEAK	Leakage control flag. Default = 2 (with energy compensation).
PLEAK	Leakage control penalty factor (default = 0.1)

Venting Hole Card.

Card 4	1	2	3	4	5	6	7	8
Variable	IVSETID	IVTYPE	IBLOCK	VNTCOF				
Type	I	I	I	F				
Default	0	0	0	0.0				
Remark	4		5	6				

VARIABLE	DESCRIPTION
IVSETID	Set ID defining the venting hole surface(s). See Remark 4.
IVTYPE	Set type of IVSETID: EQ.0: Part Set (default). EQ.1: Part ID. EQ.2: Segment Set.
IBLOCK	Flag for considering blockage effects for porosity and vents (see Remark 5): EQ.0: no (blockage is NOT considered, default). EQ.1: yes (blockage is considered).
VNTCOF	Vent Coefficient for scaling the flow. See Remark 6.

ALE Mesh Card. Parameters for ALE mesh automatic definition and its transformation.

Card 5	1	2	3	4	5	6	7	8
Variable	NX/IDA	NY/IDG	NZ	MOVERN	ZOOM			
Type	I	I	I	I	I			
Default	none	none	none	0	0			
Remark	7	7	7	8	9			

VARIABLE **DESCRIPTION**

Option 1: Automatic ALE mesh, activated by NZ.NE.0 (blank):

NX NX is the number of ALE elements to be generated in the x direction. See remark 7.

NY NY is the number of ALE elements to be generated in the y direction. See remark 7.

NZ NZ is the number of ALE elements to be generated in the z direction. See remark 7.

Option 2: ALE mesh from part ID:

IDAIR IDAIR is the Part ID of the initial air mesh. See remark 7.

IDGAS IDGAS is defined as Part ID of the initial gas mesh. See remark 7.

NZ Leave blank to activate options 2. See remark 7.

Variables common to both options:

MOVERN ALE mesh automatic motion option (see Remark 8).

EQ.0: ALE mesh is fixed in space.

GT.0: Node group id. See *ALE_REFERENCE_SYSTEM_-NODE ALE mesh can be moved with PRTYP = 5, mesh motion follows a coordinate system defined by 3 reference nodes.

VARIABLE **DESCRIPTION**

ZOOM ALE mesh automatic expansion option (see Remark 9):

EQ.0: do not expand ALE mesh

EQ.1: Expand/contract ALE mesh by keeping all airbag parts contained within the ALE mesh (equivalent to PRTYP = 9).

Origin for ALE Mesh Card. Include Cards 5a and 5b when NZ > 0.

Card 5a	1	2	3	4	5	6	7	8
Variable	X0	Y0	Z0	X1	Y1	Z1		
Type	F	F	F	F	F	F		
Default	none	none	none	none	none	none		

Card 5b	1	2	3	4	5	6	7	8
Variable	X2	Y2	Z2	Z3	Y3	Z3		
Type	F	F	F	F	F	F		
Default	none	none	none	none	none	none		

VARIABLE **DESCRIPTION**

X0, Y0, Z0 Coordinates of origin for ALE mesh generation (node0).

X1, Y1, Z1 Coordinates of point 1 for ALE mesh generation (node1).

$$x\text{-extent} = \text{node1} - \text{node0}$$

X2, Y2, Z2 Coordinates of point 2 for ALE mesh generation (node2).

$$y\text{-extent} = \text{node2} - \text{node0}$$

X3, Y3, Z3 Coordinates of point 3 for ALE mesh generation(node3).

$$z\text{-extent} = \text{node3} - \text{node0}$$

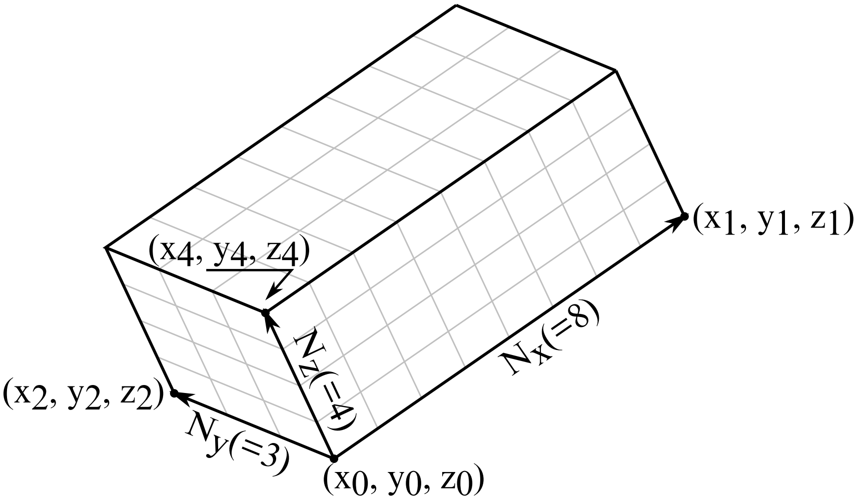


Figure 3-7. Illustration of automatic mesh generation for the ALE mesh in a hexahederal region

Miscellaneous Parameters Card.

Card 6	1	2	3	4	5	6	7	8
Variable	SWTIME		HG	NAIR	NGAS	NORIF	LCVEL	LCT
Type	F		F	I	I	I	I	I
Default	1e16		0.	0	0	0	0	0
Remarks	3						10	11

VARIABLE **DESCRIPTION**

SWTIME Time to switch from ALE method to control volume (CV) method. Once switched, a method similar to that used by the *AIRBAG_HYBRID card is used.

HG Hourglass control for ALE fluid mesh(es).

NAIR Number of Air components. For example, this equals 2 in case air contains 80% of N2 and 20% of O2. If air is defined as 1 single gas then NAIR = 1.

NGAS Number of inflator Gas components.

VARIABLE	DESCRIPTION
NORIF	Number of point sources or orifices. This determines the number of point source cards to be read.
LCVEL	Load curve ID for inlet velocity (see also TNKVOL & TNKFINP of card 2 above). This is the same estimated velocity curve used in *SECTION_POINT_SOURCE_MIXTURE card.
LCT	Load curve ID for inlet gas temperature (see *AIRBAG_HYBRID).

Air Component Card. Include NAIR cards, one for each air component.

Card 7	1	2	3	4	5	6	7	8
Variable				MWAIR	INITM	AIRA	AIRB	AIRC
Type				F	F	F	F	F
Default				0	0	0	0.	0.
Remarks						12	12	12

VARIABLE	DESCRIPTION
MWAIR	Molecular weight of air component
INITA	Initial Mass Fraction of Air component(s)
AIRA	First Coefficient of molar heat capacity at constant pressure (e.g., J/mole/K, remark 12).
AIRB	Second Coefficient of molar heat capacity at constant pressure (e.g., J/mole/K^2, remark 12).
AIRC	Third Coefficient of molar heat capacity at constant pressure (e.g., J/mole/K^3, remark 12).

Gas Component Card. Include NGAS cards, one for each gas component.

Card 8	1	2	3	4	5	6	7	8
Variable	LCMF			MWGAS		GASA	GASB	GASC
Type	I			F		F	F	F
Default	none			0		0	0.	0.
Remarks	11					12	12	12

VARIABLE	DESCRIPTION
LCMF	Load curve ID for mass flow rate (see *AIRBAG_HYBRID, e.g., kg/s).
MWGAS	Molecular weight of inflator gas components.
GASA	First Coefficient of molar heat capacity at constant pressure (e.g., J/mole/K, remark 12).
GASB	Second Coefficient of molar heat capacity at constant pressure (e.g., J/mole/K^2, remark 12).
GASC	Third Coefficient of molar heat capacity at constant pressure (e.g., J/mole/K^3, remark 12).

Point Source Cards. Include NORIF cards, one for each point source.

Card 9	1	2	3	4	5	6	7	8
Variable	NODEID	VECID	ORIFARE					
Type	I	I	I					
Default	0	0	0					

VARIABLE	DESCRIPTION
NODEID	The node ID defining the point source.

VARIABLE	DESCRIPTION
VECID	The vector ID defining the direction of flow at the point source.
ORIFARE	The orifice area at the point source.

Remarks:

1. This set ID typically contains the Lagrangian segments of the 3 parts that are coupled to the inflator gas: airbag, airbag canister (compartment), inflator. As in all control-volume, orientation of elements representing bag and canister should point outward. During the ALE phase the segment normal will be reversed automatically for fluid-structure coupling. *However, the orientation of inflator element normal vectors should point to its center.* See Figure 3-8.

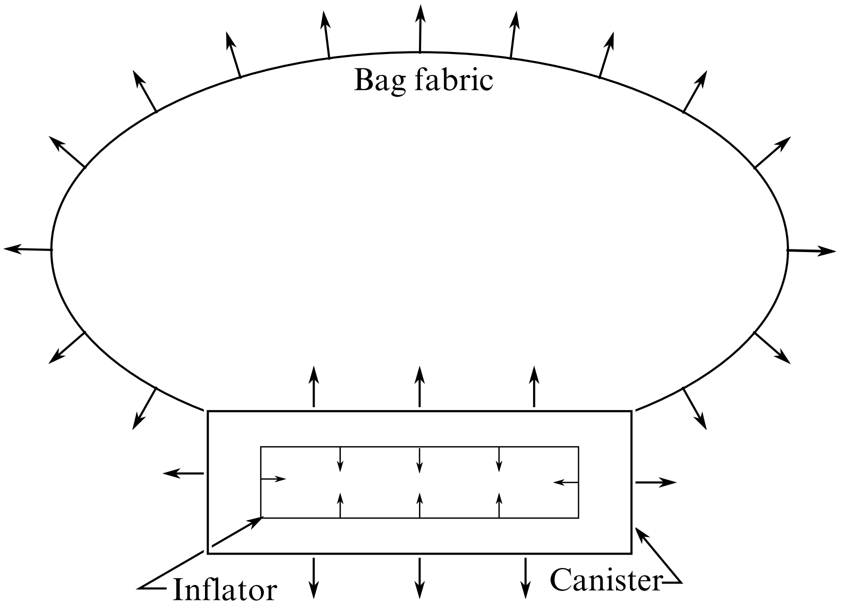


Figure 3-8. Arrows indicate "outward" normal

2. Atmospheric density for the ambient gas (air) can be computed from

$$\rho_{\text{amb}} = \frac{P_{\text{amb}}}{RT_{\text{amb}}}$$

3. Since ALL ALE related activities will be turned off after the switch from ALE method to control-volume method, no other ALE coupling will exist beyond t = SWTIME.

4. Vent definition will be used for ALE venting. Upon switching area of the segments will be used for venting as a23 in *AIRBAG_HYBRID.

5. Fabric porosity for ALE and *AIRBAG_HYBRID can be defined on MAT_FABRIC. Define FLC and FAC on *MAT_FABRIC. FVOPT 7 and 8 will be used for both ALE and *AIRBAG_HYBRID. IBLOCK = 0 will use FVOPT = 7 and IBLOCK = 1 will use FVOPT = 8.

6. VCOF will be used to scale the vent area for ALE venting and this coefficient will be used as vent coefficient c23 for *AIRBAG_HYBRID upon switching.

7. If NX, NY and NZ are defined (option 1), card 5a and card 5b should be defined to let LS-DYNA generate the mesh for ALE. Alternatively if NZ is 0 (option 2), then NX = IDAIR and NY = IDGAS. In the latter case the user need to supply the ALE mesh whose PID = IDAIR.

8. If the airbag moves with the vehicle, set MOVERN = GROUPID, this GROUPID is defined using *ALE_REFERENCE_SYSTEM_NODE. The 3 nodes defined in ALE_REFERENCE_SYSTEM_NODE will be used to transform the ALE mesh. The point sources will also follow this motion. This simulates PRTYP = 5 in the *ALE_REFERENCE_SYSTEM_GROUP card.

9. Automatic expansion/contraction of the ALE mesh to follow the airbag expansion can be turned on by setting zoom = 1. This feature is particularly useful for fully folded airbags requiring very fine ale mesh initially. As the airbag inflates the ale mesh will be automatically scaled such that the airbag will be contained within the ALE mesh. This simulates PRTYP = 9 in the *ALE_REFERENCE_SYSTEM_GROUP card.

10. There are 3 methods for defining the inlet gas velocity:

    a) Inlet gas velocity is estimated by LSDYNA method (testing), if

    $$\text{LCVEL} = 0 \Rightarrow \text{TNKVOL} = \text{Tank volume}$$

    and

    $$\text{TNKFINP} = \text{Tank final pressure from tank test data.}$$

    b) Inlet gas velocity is estimated automatically by Lian-Bhalsod-Olovsson method if,

    $$\text{LCVEL} = 0 \Rightarrow \text{TNKVOL} = \text{Tank volume.}$$

    and

    $$\text{TNKFINP} = \text{blank.}$$

    c) Inlet gas velocity is defined by user via a load curve LCVEL if,

    $$\text{LCVEL} = n,$$

    $$\text{TNKVOL} = 0,$$

    and

$$\text{TNKFINP} = 0$$

11. LCT and LCIDM should have the same number of sampling points.

12. The per-mass-unit, temperature-dependent, constant-pressure heat capacity is

$$C_p(T) = \frac{[A + BT + CT^2]}{MW}.$$

where,

$$A = \tilde{C}_{p_0}$$

these quantities often have units of,

$C_p(T)$	A	B	C
$\frac{\text{J}}{\text{kg} \times \text{K}}$	$\frac{\text{J}}{\text{mole} \times \text{K}}$	$\frac{\text{J}}{\text{mole} \times \text{K}^2}$	$\frac{\text{J}}{\text{mole} \times \text{K}^3}$

13. Sometimes CTYPE = 6 may be used for complex folded airbag. NQUAD = 2 may be used as a starting value and increase as necessary depending on the relative mesh resolutions of the Lagrangian and ALE meshes.

14. Use a load curve for PFAC whenever possible. It tends to be more robust.

Related Cards:

$$\text{AIR} \longrightarrow \begin{cases} \text{*PART (AMMG2)} \\ \text{*SECTION\_SOLID} \\ \text{*MAT\_GAS\_MIXTURE} \end{cases}$$

$$\text{GAS} \longrightarrow \begin{cases} \text{*PART (AMMG1)} \\ \text{*SECTION\_POINT\_SOURCE\_MIXTURE} \\ \text{*MAT\_GAS\_MIXTURE} \end{cases}$$

Couplings ⟶ *CONSTRAINED_LAGRANGE_IN_SOLID

ALE Mesh Motion ⟶ *ALE_REFERENCE_SYSTEM_GROUP

Control Volume ⟶ *AIRBAG_HYBRID

Vent ⟶ *AIRBAG_ALE/Card4

Example 1:

```
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
*AIRBAG_ALE
$#1     SID   SIDTYPE      NONE      NONE      NONE      NONE       MWD      SPSF
        123         1         0         0         0         0       0.0       0.0
```

```
$#2 ATMOST    ATMOSP      NONE        GC        CC    TNKVOL     TNKFP
    298.15 1.0132E-4         0     8.314       0.0       0.0       0.0
$#3  NQUAD     CTYPE      PFAC      FRIC    FRCMIN   NRMTYPE     ILEAK     PLEAK
         4         4     -1000       0.0       0.3         0         2       0.1
$#4 VSETID  IVSETTYP    IBLOCK  VENTCOEF
         1         2         0      1.00
$#5NXIDAIR   NYIDGAS        NZ    MOVERN      ZOOM
     50000     50003         0         0         0
$#6 SWTIME      NONE        HG      NAIR      NGAS     NORIF     LCVEL       LCT
   1000.00     0.000     1.e-4         1         1         8      2002      2001
$#7 AIR         NONE      NONE     MWAIR     INITM      AIRA      AIRB      AIRC
         0         0         0   0.02897      1.00    29.100   0.00000   0.00000
$#8 GASLCM      NONE      NONE     MWGAS      NONE      GASA      GASB      GASC
      2003         0         0    0.0235         0    28.000   0.00000   0.00000
$#9 NODEID    VECTID  ORIFAREA
    100019         1 13.500000
    100020         2 13.500000
    100021         3 13.500000
    100022         4 13.500000
    100023         5 13.500000
    100024         6 13.500000
    100017         7 13.500000
    100018         8 13.500000
$ PFAC CURVE = penalty factor curve.
*DEFINE_CURVE
$     lcid      sidr       sfa       sfo      offa      offo    dattyp
      1000         0       0.0       2.0       0.0       0.0
$                 a1                  o1
                 0.0          0.00000000
           1.0000000        4.013000e-04
*SET_SEGMENT_TITLE
vent segments (defined in IVSETID)
         1       0.0       0.0       0.0       0.0
      1735      1736       661      1697       0.0       0.0       0.0       0.0
      1735      2337      1993      1736       0.0       0.0       0.0       0.0
      1735      1969      1988      2337       0.0       0.0       0.0       0.0
      1735      1697       656      1969       0.0       0.0       0.0       0.0
*DEFINE_VECTOR
$#     vid        xt        yt        zt        xh        yh        zh
         1       0.0       0.0-16.250000 21.213200 21.213200-16.250000
         2       0.0       0.0-16.250000 30.000000-1.000e-06-16.250000
         3       0.0       0.0-16.250000 21.213200-21.213200-16.250000
         4       0.0       0.0-16.250000-1.000e-06-30.000000-16.250000
         5       0.0       0.0-16.250000-21.213200-21.213200-16.250000
         6       0.0       0.0-16.250000-30.0000001.0000e-06-16.250000
         7       0.0       0.0-16.250000-21.213200 21.213200-16.250000
         8       0.0       0.0-16.2500001.0000e-06 30.000000-16.250000
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
```

In this example, pre-existing background air mesh with part ID 50000 and gas mesh with part ID 50003 are used. Thus NZ = 0. There is no mesh motion nor expansion allowed. An inlet gas velocity curve is provided.

Example 2:

```
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
$ SIDTYP: 0=SGSID; 1=PSID
*AIRBAG_ALE
$#1    SID   SIDTYPE      NONE      NONE      NONE      NONE       MWD      SPSF
         1         1         0        0.        0.        0.        0.        0.
$#2 ATMOST    ATMOSP      NONE        GC        CC    TNKVOL     TNKFP
      298.   101325.       0.0     8.314        1.    6.0E-5         0
```

```
$#3  NQUAD     CTYPE      PFAC      FRIC    FRCMIN  NRMTYPE     ILEAK     PLEAK
         2         6      -321       0.0       0.3         1         2       0.1
$#4 VSETID  IVSETTYP    IBLOCK  VENTCOEF
         0         0         0         0
$#5NXIDAIR   NYIDGAS        NZ    MOVERN      ZOOM
        11        11         9
$5b     x0        y0        z0        x1        y1        z1  NOT-USED  NOT-USED
      -0.3      -0.3    -0.135       0.3      -0.3    -0.135
$5c     x2        y2        z2        x3        y3        z3  NOT-USED  NOT-USED
      -0.3       0.3    -0.135      -0.3      -0.3      0.39
$#6 SWTIME      NONE        HG      NAIR      NGAS     NORIF     LCVEL       LCT
   0.04000     0.005     1.e-4         2         1         1         0         2
$#7 AIR         NONE      NONE     MWAIR     INITM      AIRA      AIRB      AIRC
                                   0.028      0.80    27.296   0.00523
                                   0.032      0.20    25.723   0.01298
$#8 GASLCM      NONE      NONE     MWGAS      NONE      GASA      GASB      GASC
         1                        0.0249              29.680   0.00880
$#9 NODEID    VECTID  ORIFAREA
      9272         1   1.00e-4
$ Lagrangian shell structure to be coupled to the inflator gas
*SET_PART_LIST
         1       0.0       0.0       0.0       0.0
         1         2         3
*DEFINE_VECTOR
$0.100000E+01,  10.000000000
$      vid        xt        yt        zt        xh        yh        zh
         1       0.0       0.0       0.0       0.0       0.0  0.100000
$ bag penetration ~ 1 mm  <====>  P_coup ~ 500000 pascal ==> ~ 5 atm
*DEFINE_CURVE
$     lcid      sidr       sfa       sfo      offa      offo    dattyp
       321         0       0.0       0.0       0.0       0.0
$                 a1                  o1
                 0.0                 0.0
          0.00100000       5.0000000e+05
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
```

In this example, LS-DYNA automatically creates the background ALE mesh with:

NX = 11 ⇒ 11 elements in the x direction
NY = 11 ⇒ 11 Elements in the y direction
NZ = 9 ⇒ 9 Elements in the z direction

*AIRBAG_INTERACTION

Purpose: To define two connected airbags which vent into each other.

Card 1	1	2	3	4	5	6	7	8
Variable	AB1	AB2	AREA	SF	PID	LCID	IFLOW	
Type	I	I	F	F	I	I	I	
Default	none	none	none	none	0	0	0	

VARIABLE **DESCRIPTION**

AB1 First airbag ID, as defined on *AIRBAG card.

AB2 Second airbag ID, as defined on *AIRBAG card.

AREA Orifice area between connected bags.

LT.0.0: |AREA| is the load curve ID defining the orifice area as a function of absolute pressure.

EQ.0.0: AREA is taken as the surface area of the part ID defined below.

SF Shape factor.

LT.0.0: |SF| is the load curve ID defining vent orifice coefficient as a function of relative time.

PID Optional part ID of the partition between the interacting control volumes. AREA is based on this part ID. If PID is negative, the blockage of the orifice part due to contact is considered,

LCID Load curve ID defining mass flow rate versus pressure difference, see *DEFINE_CURVE. If LCID is defined AREA, SF and PID are ignored.

IFLOW Flow direction

LT.0: One way flow from AB1 to AB2 only.

EQ.0: Two way flow between AB1 and AB2.

GT.0: One way flow from AB2 to AB1 only.

Remarks:

Mass flow rate and temperature load curves for the secondary chambers must be defined as null curves, for example, in the DEFINE_CURVE definitions give two points (0.0, 0.0) and (10000., 0.0).

All input options are valid for the following airbag types:

*AIRBAG_SIMPLE_AIRBAG_MODEL

*AIRBAG_WANG_NEFSKE

*AIRBAG_WANG_NEFSKE_JETTING

*AIRBAG_WANG_NEFSKE_MULTIPLE_JETTING

*AIRBAG_HYBRID

*AIRBAG_HYBRID_JETTING

The LCID defining mass flow rate vs. pressure difference may additionally be used with:

*AIRBAG_LOAD_CURVE

*AIRBAG_LINEAR_FLUID

If the AREA, SF, and PID defined method is used to define the interaction then the airbags must contain the same gas, i.e. C_p, C_v and g must be the same. The flow between bags is governed by formulae which are similar to those of Wang-Nefske.

*AIRBAG_PARTICLE_{OPTION1}_{OPTION2}_{OPTION3}_{OPTION4}

Available options include:

OPTION1 applies to the MPP implementation only.

MPP

OPTION2 also applies to the MPP implementation only. When the DECOMPOSITON option is present, LS-DYNA will automatically insert *CONTROL_MPP_DECOMPOSITION_BAGREF and CONTROL_MPP_DECOMPOSITION_ARRANGE_PARTS keywords if they are not already present in the input.

DECOMPOSITION

OPTION3 provides a means to specify an airbag ID number and a heading for the airbag.

ID

TITLE

OPTION4:

MOLEFRACTION (see Remark 10)

Purpose: To define an airbag using the particle method.

> **NOTE:** This model requires that surface normal vectors be oriented *inward*, unlike that the traditional CV method for which they must point *outward*. To check bag and chamber integrity in the CPM model see the CPMERR option on the *CONTROL_CPM card.

MPP Card. Additional card for MPP keyword option.

MPP	1	2	3	4	5	6	7	8
Variable	SX	SY	SZ					
Type	F	F	F					
Default	none	none	none					

VARIABLE	DESCRIPTION
SX, SY, SZ	Scale factor for each direction used during the MPP decomposition. For instance, increasing SX from 1 to 10 will give increase the probability that the model is divided along the x-direction.

Title Card. Additional card for ID or TITLE keyword options.

TITLE	1	2	3	4	5	6	7	8
Variable	BAGID	HEADING						
Type	I	A60						

The BAGID is referenced in, e.g., *INITIAL_AIRBAG_PARTICLE_POSITION.

VARIABLE	DESCRIPTION
BAGID	Airbag ID. This must be a unique number.
HEADING	Airbag descriptor. It is suggested that unique descriptions be used.

Card 1	1	2	3	4	5	6	7	8
Variable	SID1	STYPE1	SID2	STYPE2	BLOCK	NPDATA	FRIC	IRPD
Type	I	I	I	I	I	I	F	I
Default	none	0	0	0	0	0.0	0.0	0

Card 2	1	2	3	4	5	6	7	8
Variable	NP	UNIT	VISFLG	TATM	PATM	NVENT	TEND	TSW
Type	I	I	I	F	F	I	F	F
Default	200,000	0	0	293K	1 atm	0	10^{10}	10^{10}

Optional control card. When the card after Card 2 begins with a "+" character the input reader processes it as this card, otherwise this card is skipped.

Card 3	1	2	3	4	5	6	7	8
Variable	TSTOP	TSMTH	OCCUP	REBL	SIDSV	PSID1		
Type	F	F	F	I	I	I		
Default	10^{10}	1msec	0.1	0	none	none		

Optional unit card. Additional Cards when Unit = 3.

Card 4	1	2	3	4	5	6	7	8
Variable		Mass		Time		Length		
Type		F		F		F		
Default		none		none		none		

Card 5	1	2	3	4	5	6	7	8
Variable	IAIR	NGAS	NORIF	NID1	NID2	NID3	CHM	CD_EXT
Type	I	I	I	I	I	I	I	F
Default	0	none	none	0	0	0	none	0.

Internal Part Set Cards. Additional Cards for STYPE2 = 2. Define SID2 cards, one for each internal part or part set.

Card 6	1	2	3	4	5	6	7	8
Variable	SIDUP	STYUP	PFRAC	LINKING				
Type	I	I	F	I				
Default	none	none	0.	none				

Heat Convection Part Set Cards. Additional Cards for NPDATA > 0. Define NPDATA cards, one for each heat convection part or part set.

Card 7	1	2	3	4	5	6	7	8
Variable	SIDH	STYPEH	HCONV	PFRIC	SDFBLK	KP	INIP	
Type	I	I	F	F	F	F	I	
Default	none	none	none	none	1.0	0.	0	

Vent Hole Card. Additional Cards for NVENT > 0. Define NVENT cards, one for vent hole.

Card 8	1	2	3	4	5	6	7	8
Variable	SID3	STYPE3	C23	LCTC23	LCPC23	ENH_V	PPOP	
Type	I	I	F	I	I	I	F	
Default	0	none	1.0	0	0	0	0.0	

Air Card. Additional Card for IAIR > 0.

Card 9	1	2	3	4	5	6	7	8
Variable	PAIR	TAIR	XMAIR	AAIR	BAIR	CAIR	NPAIR	NPRLX
Type	F	F	F	F	F	F	I	I/F
Default	PATM	TATM	none	none	0.0	0.0	0	0

MOLEFRACTION Card. Additional card for the MOLEFRACTION option.

Card 10	1	2	3	4	5	6	7	8
Variable	LCMASS							
Type	I							
Default	none							

Gas Cards. NGAS additional Cards, one for each gas (card format for i^{th} gas).

Card 11	1	2	3	4	5	6	7	8
Variable	LCMi	LCTi	XMi	Ai	Bi	Ci	INFGi	
Type	I	I	F	F	F	F	I	
Default	none	none	none	None	0.0	0.0	1	

Orifice Cards. NORIF additional Cards, one for each orifice (card format for ith orifice).

Card 12	1	2	3	4	5	6	7	8
Variable	NIDi	ANi	VDi	CAi	INFOi	IMOM	IANG	CHM_ID
Type	I	F	I	F	I	I	I	I
Default	none	none	none	30 Deg	1	0	0	0

VARIABLE	DESCRIPTION
SID1	Part or part set ID defining the complete airbag.
STYPE1	Set type: EQ.0: Part EQ.1: Part set
SID2	Part or part set ID defining the internal parts of the airbag.
STYPE2	Set type: EQ.0: Part EQ.1: Part set EQ.2: Number of parts to read (Not recommended for general use)

VARIABLE **DESCRIPTION**

BLOCK

Blocking. Block must be set to a two-digit number

$$\text{BLOCK} = \text{M} \times 10 + \text{N},$$

The 10′s digit controls the treatment of particles that escape due to deleted elements (particles are always tracked and marked).

M.EQ.0: Active particle method for which causes particles to be put back into the bag.

M.EQ.1: Particles are leaked through vents. See Remark 3.

The 1′s digit controls the treatment of leakage.

N.EQ.0: Always consider porosity leakage without considering blockage due to contact.

N.EQ.1: Check if airbag node is in contact or not. If yes, 1/4 (quad) or 1/3 (tri) of the segment surface will not have porosity leakage due to contact.

N.EQ.2: Same as 1 but no blockage for external vents

N.EQ.3: Same as 1 but no blockage for internal vents

N.EQ.4: Same as 1 but no blockage for all vents.

NPDATA

Number of parts or part sets data.

FRIC

Friction factor. See Remark 2.

IRPD

Dynamic scaling of particle radius.

EQ.0: Off

EQ.1: On

NP

Number of particles. (Default = 200,000)

UNIT

Unit system:

EQ.0: kg-mm-ms-K

EQ.1: SI

EQ.2: tonne-mm-s-K

EQ.3: User defined units (see Remark 11)

VARIABLE **DESCRIPTION**

VISFLG

Visible particles. This field affects only the CPM database. See Remark 5.

EQ.0: Default to 1

EQ.1: Output particle's coordinates, velocities, mass, radius, spin energy, translational energy

EQ.2: Output reduce data set with coordinates only

EQ.3: Suppress CPM database

TATM

Atmospheric temperature. (Default = 293K)

PATM

Atmospheric pressure. (Default = 1 ATM)

NVENT

Number of vent hole parts or part sets

TEND

Time when all (NP) particles have entered bag. (Default = 10^{10})

TSW

Time at which algorithm switches to control volumes. (Default = 10^{10})

TSTOP

Time at which front tracking switches from IAIR = 4 to IAIR = 2.

TSMTH

To avoid sudden jumps in the pressure signal during switching there is a transition period during which the front tracking is tapered. The default time of 1 millisecond will be applied if this value is set to zero.

OCCUP

Particles occupy OCCUP percent of the airbag's volume. The default value of OCCUP is 10%. This field can be used to balance computational cost and signal quality. OCCUP ranges from 0.001 to 0.1.

REBL

If the option is ON, all energy stored from damping will be evenly distributed as vibrational energy to all particles. This improves the pressure calculation in certain applications.

EQ.0: Off (Default)

EQ.1: On

SIDSV

Part set ID for internal shell part. The volume occupied by this part is excluded from the bag volume. These internal parts must be consistently orientated for the excluded volume to be correctly calculated.

VARIABLE **DESCRIPTION**

PSID1

Part set ID for external parts which have normal pointed outward. This option is usually used with airbag integrity check while there are two CPM bags connected with bag interaction. Therefore, one of the bag can have the correct shell orientation but the share parts for the second bag will have wrong orientation. This option will automatically flip the parts defined in this set in the second bag during integrity checking.

Mass, Time, Length

Conversion factor from current unit to MKS unit. For example, if the current unit is using kg-mm-ms, the input should be 1.0, 0.001, 0.001

IAIR

Initial gas inside bag considered:

EQ.0: No

EQ.1: Yes, using control volume method.

EQ.-1: Yes, using control volume method. In this cake ambient air enters the bag when PATM is greater than bag pressure.

EQ.2: Yes, using the particle method.

EQ.4: Yes, using the particle method. Initial air particles are used for the gas front tracking algorithm but they do not apply forces when they collide with a segment. Instead, a uniform pressure is applied to the airbag based on the ratio of air and inflator particles. In this case NPRLX must be negative so that forces are not applied by the initial air.

NGAS

Number of gas components

NORIF

Number of orifices

NID1 - NID3

Three nodes defining a moving coordinate system for the direction of flow through the gas inlet nozzles (Default = fixed system)

CHM

Chamber ID used in *DEFINE_CPM_CHAMBER. See Remark 7.

CD_EXT

Drag coefficient for external air. If the value is not zero, the inertial effect from external air will be considered and forces will be applied in the normal direction on the exterior airbag surface.

VARIABLE **DESCRIPTION**

SIDUP

Part or part set ID defining the internal parts that pressure will be applied to. This internal structure acts as a valve to control the external vent hole area. Pressure will be applied only after switch to UP (uniform pressure) using TSW.

STYUP

Set type:

EQ.0: Part

EQ.1: Part set

PFRAC

Fraction of pressure to be applied to the set (0.0 to 1.0). If PFRAC = 0, no pressure is applied to internal parts.

LINKING

Part ID of an internal part that is coupled to the external vent definition. The minimum area of this part or the vent hole will be used for actual venting area.

SIDH

Part or part set ID defining part data.

STYPEH

Set type:

EQ.0: Part

EQ.1: Part set

HCONV

Heat convection coefficient used to calculate heat loss from the airbag external surface to ambient (W/K/m^2). See *AIRBAG_HYBRID developments (Resp. P.O. Marklund).

LT.0: |HCONV| is a load curve ID defines heat convection coefficient as a function of time.

PFRIC

Friction factor. PFRIC Defaults to FRIC from 1st card 7th field.

SDFBLK

Scale down factor for blockage factor (Default = 1, no scale down). The valid factor will be (0,1]. If 0, it will set to 1.

KP

Thermal conductivity of the part. See Remark 9.

INIP

Place initial air particles on surface.

EQ.0: yes (default)

EQ.1: no

This feature exclude surfaces from initial particle placement. This option is useful for preventing particles from being trapped between adjacent fabric layers.

VARIABLE **DESCRIPTION**

SID3

Part or part set ID defining vent holes.

STYPE3

Set type:

EQ.0: Part

EQ.1: Part set which each part being treated separately

EQ.2: Part set and all parts are treated as one vent. See Remark 13.

C23

GE..0: Vent hole coefficient, a parameter of Wang-Nefske leakage. A value between 0.0-1.0 can be input, default = 1.0. See Remark 1.

LT.0: ID for *DEFINE_CPM_VENT

LCTC23

Load curve defining vent hole coefficient as a function of time. LCTC23 can be defined through *DEFINE_CURVE_FUNCTION. If omitted a curve equal to 1 used. See Remark 1.

LCPC23

Load curve defining vent hole coefficient as a function of pressure. If omitted a curve equal to 1 is used. See Remark 1.

ENH_V

Enhanced venting option. See Remark 8.

EQ.0: Off (default)

EQ.1: On

EQ.2: Two way flow for internal vent; treated as hole for external vent (See Remark 8)

PPOP

Pressure difference between interior and ambient pressure (PATM) to open the vent holes. Once the vents are open, they will stay open.

PAIR

Initial pressure inside bag. (Default PAIR = PATM)

TAIR

Initial temperature inside bag. (Default, TAIR = TATM)

XMAIR

Molar mass of gas initially inside bag.

AAIR - CAIR

Constant, linear, and quadratic heat capacity parameters.

NPAIR

Number of particle for air. See Remark 6.

VARIABLE	DESCRIPTION
NPRLX	Number of cycles to reach thermal equilibrium. See Remark 6. LT.0: If more than 50% of the collision to fabric is from initial air particle, the contact force will not apply to the fabric segment in order to keep its original shape. If the number contains ".", "e" or "E", NPRLX will treated as an end *time* rather than as a cycle count.
LCMASS	Total mass flow rate curve for the MOLEFRACTION option.
LCM*i*	Mass flow rate curve for gas component *i*, unless the MOLEFRACTION option is used, then it is the time dependent molar fraction of the total flow for gas component *i*.
LCT*i*	Temperature curve for gas component *i*.
XM*i*	Molar mass of gas component *i*.
A*i* - C*i*	Constant, linear, and quadratic heat capacity parameters for gas component *i*.
INFG*i*	Inflator ID for this gas component (Defaults to 1).
NID*i*	Node ID/Shell ID defining the location of nozzle *i*. See Remark 12.
AN*i*	Area of nozzle *i*. (Default: all nozzles are assigned the same area)
VD*i*	GT.0: Vector ID. Initial direction of gas inflow at nozzle *i*. LT.0: Values in the NID*i* fields are interpreted as shell IDs. See Remark 12. EQ.-1: direction of gas inflow is using shell normal EQ.-2: direction of gas inflow is in reversed shell normal
CA*i*	Cone angle in degrees (Defaults to 30 degrees). This option is used only when IANG equal to 1.
INFO*i*	Inflator ID for this orifice. (Default = 1)
IMOM	Inflator reaction force (R5.1.1 release and later). EQ.0: Off EQ.1: On

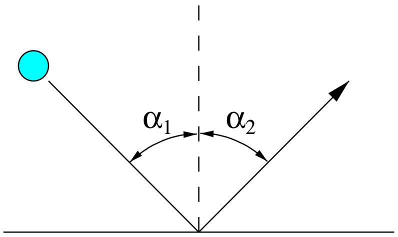
VARIABLE	DESCRIPTION
IANG	Activation for cone angle to use for friction calibration(not normally used; eliminates thermal energy of particles from inflator). EQ.0: Off (Default) EQ.1: On
CHM_ID	Chamber ID where the inflator node resides. Chambers are defined using the *DEFINE_CPM_CHAMBER keyword.

Remarks:

1. **Formula for Total Vent Hole Coefficient.** The value must be between 0 and 1.

$$\text{Total vent hole coefficient} = \min(\max(\text{C23} \times \text{LCTC23} \times \text{LCPC23}, 0), 1)$$

2. **Surface Roughtness.** Friction factor to simulate the surface roughness. If the surface is frictionless the particle incoming angle $\alpha_1$is equal to the deflection angle α_2(see figure below).



Considering surface roughness F_r and the total angle α will have the following relationships:

$$0 \le F_\mathrm{r} \le 1$$

$$\alpha = \alpha_1 + \alpha_2$$

For the special case when,

$$F_\mathrm{r} = 1$$

the incoming particle will bounce back from its incoming direction,

$$\alpha = 0.$$

For, $-1 \le F_r < 0$, particles will bounce towards the surface by the following relationship

$$\alpha = 2\left[\alpha_1 - F_r\left(\frac{\pi}{4} - \frac{\alpha_1}{2}\right)\right].$$

3. **Blocking and BLOCK Field.** Setting the 10's digit to 1 allows for physical holes in an airbag. In this case, particles that are far away from the airbag are

disabled. In most case, these are particles that have escaped through unclosed surfaces due to physical holes, failed elements, etc. This reduces the bucket sort search distance.

4. **Convection Energy Balance.** The change in energy due to convection is given by

$$\frac{\mathrm{d}E}{\mathrm{d}t} = A \times \mathrm{HCONV} \times \left(T_{\mathrm{bag}} - T_{\mathrm{atm}}\right).$$

Where,

A = is part area.

HCONV = user defined heat convection coefficient.

T_{bag} = the weighted average temperature impacting particles.

T_{atm} = aambient temperature.

5. **Output Files.** Particle time history data is always output to d3plot database now. LS-PrePost 2.3 and later can display and fringe this data. In order to reduce runtime memory requirements, VISFLG should be set to 0 (disabled). For LS-DYNA 971 R6.1 and later, VISFLG only affects Version 4 CPM output (see *DATABASE_CPM).

6. **Spatial Distribution Equilibration for Airbag Particles.** Total number of particles used in each card is NP + NPAIR. Since the initial air particles are placed at the surface of the airbag segments with correct velocity distribution initially, particles are not randomly distributed in space. It requires a finite number of relaxation cycles, NPRLX, to allow particles to move and produce better spatial distribution.

   Since the momentum and energy transfer between particles are based on perfect elastic collision, CPM solver would like to keep similar mole per particle between inflator and initial air particles. CPM solver will check the following factor.

$$\mathrm{factor} = \left|1 - \frac{\text{mole per particle of initial air}}{\text{mole per particle of inflator gas}}\right|$$

   If the factor is more than 10% apart, code will issue the warning message with the tag, (SOL+1232) and provide the suggested NPAIR value. User needs make decision to adjust the NPAIR value based on the application. For example, user setup only initial air without any inflator gas for certain impact analysis should ignore this warning message.

7. **Remark Concerning *DEFINE_CPM_CHAMBER.** By default initial air particles will be evenly placed on airbag segments which cannot sense the local volume. This will create incorrect pressure field if the bag has several distinct pockets. *DEFINE_CPM_CHAMBER allows the user to initialize air particles

by volume ratios of regions of airbag. The particles will be distributed proportional to the defined chamber volume to achieve better pressure distribution.

8. **Enhanced Venting.** When enhanced venting is on, the vent hole's equivalent radius R_{eq} will be calculated. Particles within R_{eq} on the high pressure side from the vent hole geometry center will be moved toward the hole. This will increase the collision frequency near the vent for particles to detect small structural features and produce better flow through the vent hole.

   ENH_V equals 1, particles are flow from high to low pressure only. EHN_V equals 2, particles can flow in both directions for internal vent.

   Particles encountering external vents are released. The ambient pressure is not taken into account and the particle will be released regardless the value of the pressure in the bag/chamber. Therefore, the vent rate will be sensitive to the vent location.

9. **Effective Convection Heat Transfer Coefficient.** If the thermal conductivity, KP, is given, then the effective convection heat transfer coefficient is given by

$$H_{\mathrm{eff}} = \left(\frac{1.0}{\mathrm{HCONV}} + \frac{\text{shell thickness}}{\mathrm{KP}}\right)^{-1},$$

   where the part thickness comes from the SECTION database. If KP is not given, H_{eff} defaults to HCONV.

10. **MOLEFRACTION Option.** Without the MOLEFRACTION option, a flow rate is specified for each species on the LCM*i* fields of Card 11. With the MOLEFRACTION option the total mass flow rate is specified in the LCMASS field on Card 10 and the molar fractions are specified in the LCM*i* fields of Card 11.

11. **User Defined Units.** If UNIT = 3 is used, there is no default value for TATM and PATM and user should provide the proper values. User also needs to provide unit conversion factors for code to set correct universal gas constant and some other variables used in the code.

12. **Shell Based Nozzle.** Node ID and shell ID based nozzle should not be used in the same airbag definition. The nozzle location is taken to be at the center of the shell and the initial nozzle direction can be defined by shell's normal or by its reversed normal. This vector is transforms with the moving coordinate system defined by NID1 - NID3. The nozzle area is set on the AN*i* fields and shell area is not taken into account; therefore, the mass flowrate distribution with shells is determined in the same way as it is with nozzles defined by nodes.

13. **Merge Part Set for Vent.** The first part in the set is designated as the master. All remaining parts are merged into the master so that enhanced venting is

treated correctly. ABSTAT_CPM output will be associated with the master part. This option has the same effect as manually merging elements into the master part.

AIRBAG_REFERENCE_GEOMETRY_*{OPTION}_{OPTION}_{OPTION}

Available options include:

<BLANK>

BIRTH

RDT

ID

Purpose: If the reference configuration of the airbag is taken as the folded configuration, the geometrical accuracy of the deployed bag will be affected by both the stretching and the compression of elements during the folding process. Such element distortions are very difficult to avoid in a folded bag. By reading in a reference configuration such as the final unstretched configuration of a deployed bag, any distortions in the initial geometry of the folded bag will have no effect on the final geometry of the inflated bag. This is because the stresses depend only on the deformation gradient matrix:

$$F_{ij} = \frac{\partial x_i}{\partial X_j}$$

where the choice of X_j may coincide with the folded or unfold configurations. It is this unfolded configuration which may be specified here.

Note that a reference geometry which is smaller than the initial airbag geometry will not induce initial tensile stresses.

If a liner is included and the parameter LNRC set to 1 in *MAT_FABRIC, compression is disabled in the liner until the reference geometry is reached, i.e., the fabric element becomes tensile.

When the BIRTH option is specified an additional card setting the BIRTH parameter is activated. The BIRTH parameter specifies a critical time value before which the reference geometry is *not used*. Until the BIRTH time is reach the input geometry is used for (1) inflating the airbag and for (2) determining the time step size, even when the RDT option is set.

NOTE: This card does not support multiple birth times. The last BIRTH value read will be used for *all* preceding *AIRBAG_REFERENCE_GEOMETRY_BIRTH definitions. RGBRTH in *MAT_FABRIC supports a material dependent birth time.

When the RDT option is active the time step size will be based on the reference geometry once the solution time exceeds the birth time. This option is useful for shrunken bags where the bag does not carry compressive loads and the elements can freely expand before stresses develop. If this option is not specified, the time step size will be based on the current configuration and will increase as the area of the elements increase. The default may be much more expensive but possibly more stable.

ID card. Additional card for keyword option ID.

ID	1	2	3	4	5	6	7	8
Variable	ID	SX	SY	SZ	NIDO			
Type	I	F	F	F	I			
Default	none	1.0	1.0	1.0	1st NID			

Birth card. Additional card for keyword option BIRTH.

Birth	1	2	3	4	5	6	7	8
Variable	BIRTH							
Type	F							
Default	0.0							

Node Cards. For each node ID having an associated reference position include an additional card in format 2. The next "*" keyword card terminates this input.

Card 2	1	2	3	4	5	6	7	8	9	10
Variable	NID	X		Y		Z				
Type	I	F		F		F				
Default	none	0.		0.		0.				
Remark										

VARIABLE	DESCRIPTION
ID	Card ID
SX, SY, SZ	Scale factor in each direction
NIDO	Node ID for origin. Default is the first node ID defined in this keyword.
BIRTH	Time at which the reference geometry activates (default = 0.0)
NID	Node ID for which a reference configuration is defined. Nodes defined in this section must also appear under the *NODE input. It is only necessary to define the reference coordinates of nodal points, if their coordinates are different than those defined in the *NODE section.
X	*x* coordinate
Y	*y* coordinate
Z	*z* coordinate

*AIRBAG_SHELL_REFERENCE_GEOMETRY_*{OPTION}*_*{OPTION}*

Available options include:

<BLANK>

RDT

ID

Purpose: Usually, the input in this section is not needed; however, sometimes it is convenient to use disjoint pre-cut airbag parts to define the reference geometries. If the reference geometry is based only on nodal input, this is not possible since in the assembled airbag the boundary nodes are merged between parts. By including the shell connectivity with the reference geometry, the reference geometry can be based on the pre-cut airbag parts instead of the assembled airbag. The elements, which are defined in this section, must have identical element ID's as those defined in the *ELEMENT_SHELL input, but the nodal ID's, which may be unique, are only used for the reference geometry. These nodes are defined in the *NODE section but can be additionlly defined in *AIRBAG_REFERENCE_GEOMETRY. The element orientation and n1-n4 ordering must be identical to the *ELEMENT_SHELL input.

When the RDT option is active the time step size will be based on the reference geometry once the solution time exceeds the birth time which can be defined by RGBRTH of *MAT_FABRIC.

ID card. Additional card for keyword option ID.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	SX	SY	SZ	NID			
Type	I	F	F	F	I			
Default	none	1.0	1.0	1.0	See List			

Card 2	1	2	3	4	5	6	7	8	9	10
Variable	EID	PID	N1	N2	N3	N4				
Type	I	I	I	I	I	I				
Default	none	none	none	none	none	none				

VARIABLE	DESCRIPTION
ID	Card ID
SX, SY, SZ	Scale factor in each direction
NID	Node ID for origin. Default is the first node ID defined in this keyword.
EID	Element ID
PID	Optional part ID, see *PART, the part ID is not used in this section.
N1	Nodal point 1
N2	Nodal point 2
N3	Nodal point 3
N4	Nodal point 4

*ALE

ALE *does not* support implicit time integration nor does it support dynamic relaxation. Furthermore, except for ALE formulation 5, which *does* support contact, ALE *does not*, in general, support contact.

In three dimensions, ALE supports *only* one-point solid elements. These solid elements can either be hexahedral, pentahedral, or tetrahedral. Pentahedrons and tetrahedrons are treated as degenerate hexahedron elements. For each ALE multi-material, strain and stress is evaluated in each solid element at a single integration point. In this sense, the ALE element formulation is equivalent to ELEFORM 1 solid formulation.

Keywords for the ALE structured solver.

*ALE_STRUCTURED_MESH

*ALE_STRUCTURED_MESH_CONTROL_POINTS

Input required for LS-DYNA's Arbitrary-Lagrangian-Eulerian capability is defined using *ALE cards. The keyword cards in this section are defined in alphabetical order:

*ALE_AMBIENT_HYDROSTATIC

*ALE_COUPLING_NODAL_CONSTRAINT

*ALE_COUPLING_NODAL_DRAG

*ALE_COUPLING_NODAL_PENALTY

*ALE_COUPLING_RIGID_BODY

*ALE_ESSENTIAL_BOUNDARY

*ALE_FAIL_SWITCH_MMG

*ALE_FRAGMENTATION

*ALE_FSI_PROJECTION

ALE_FSI_SWITCH_MMG_*{OPTION}

*ALE_FSI_TO_LOAD_NODE

*ALE_MULTI-MATERIAL_GROUP

*ALE_REFERENCE_SYSTEM_CURVE

*ALE_REFERENCE_SYSTEM_GROUP

*ALE_REFERENCE_SYSTEM_NODE

*ALE_REFERENCE_SYSTEM_SWITCH

*ALE_REFINE

*ALE_SMOOTHING

*ALE_TANK_TEST

*ALE_UP_SWITCH

For other input information related to the ALE capability, see keywords:

*ALE_TANK_TEST

*BOUNDARY_AMBIENT_EOS

*CONSTRAINED_EULER_IN_EULER

*CONSTRAINED_LAGRANGE_IN_SOLID

*CONTROL_ALE

*DATABASE_FSI

*INITIAL_VOID

*INITIAL_VOLUME_FRACTION

*INITIAL_VOLUME_FRACTION_GEOMETRY

*SECTION_SOLID

*SECTION_POINT_SOURCE_FOR_GAS_ONLY

*SECTION_POINT_SOURCE_MIXTURE

*SET_MULTIMATERIAL_GROUP_LIST

*CONSTRAINED_EULER_IN_EULER

For a single gaseous material:

*EOS_LINEAR_POLYNOMIAL

*EOS_IDEAL_GAS

*MAT_NULL

For multiple gaseous materials:

*MAT_GAS_MIXTURE

*INTIAL_GAS_MIXTURE

*ALE_AMBIENT_HYDROSTATIC

Purpose: When an ALE model contains one or more ambient (or reservoir-type) ALE parts (ELFORM = 11 and AET = 4), this command may be used to initialize the hydrostatic pressure field in the ambient ALE domain due to gravity. The *LOAD_-BODY_{OPTION} keyword must be defined. The associated *INITIAL_HYDROSTAT-IC_ALE keyword may be used to define a similar initial hydrostatic pressure field for the regular ALE domain (not reservoir-type region).

Card 1	1	2	3	4	5	6	7	8
Variable	ALESID	STYPE	VECID	GRAV	PBASE	RAMPTLC		
Type	I	I	I	I	I	I		
Default	none	0	none	0	0	0		

Card 2	1	2	3	4	5	6	7	8
Variable	NID	MMGBL						
Type	I	I						
Default	None	none						

VARIABLE	DESCRIPTION
ALESID	ALESID defines the reservoir-type ALE domain/mesh whose hydrostatic pressure field due to gravity is being initialized by this keyword. See Remark 4.
STYPE	ALESID set type. See Remark 4. EQ.0: Part set ID (PSID), EQ.1: Part ID (PID)), EQ.2: Solid set ID (SSID).
VECID	Vector ID of a vector defining the direction of gravity.

VARIABLE	DESCRIPTION
GRAV	Magnitude of the Gravitational acceleration. For example, in metric units the value is usually set to 9.80665 m/s^2.
PBASE	Nominal or reference pressure at the top surface of all fluid layers. By convention, the gravity direction points from the top layer to the bottom layer. Each fluid layer must be represented by an ALE multi-material group ID (AMMGID or MMG). See Remark 1.
RAMPTLC	A ramping time function load curve ID. This curve (via *DEFINE_CURVE) defines how gravity is ramped up as a function of time. Given GRAV value above, the curve's ordinate varies from 0.0 to 1.0, and its abscissa is the (ramping) time. See Remark 2.
NID	Node ID defining the top of an ALE fluid (AMMG) layer.
MMGBL	AMMG ID of the fluid layer immediately below this NID. Each node is defined in association with one AMMG layer below it. See Remark 4.

Remarks:

1. **Pressure in Multi-Layer Fluids.** For models using multi-layer ALE Fluids the pressure at the top surface of the top fluid layer is set to PBASE and the hydrostatic pressure is computed as following

$$P = P_{\text{base}} + \sum_{i=1}^{N_{\text{layers}}} \rho_i g h_i \, .$$

2. **Hydrostatic Pressure Ramp Up.** If RAMPTLC is activated (i.e. not equal to "0"), then the hydrostatic pressure is effectively ramped up over a user-defined duration and kept steady. When this load curve is defined, do not define the associated *INITIAL_HYDROSTATIC_ALE card to initialize the hydrostatic pressure for the non-reservoir ALE domain. The hydrostatic pressure in the regular ALE region will be initialized indirectly as a consequence of the hydrostatic pressure generated in the reservoir-type ALE domain. The same load curve should be used to ramp up gravity in a corresponding *LOAD_BODY_(OPTION) card. Via this approach, any submerged Lagrangian structure coupled to the ALE fluids will have time to equilibrate to the proper hydrostatic condition.

3. **Limitation on EOS Model.** The keyword only supports *EOS_GRUNEISEN and *EOS_LINEAR_POLYNOMIAL, but only inthe following two cases,

$$c_3 = c_4 = c_5 = c_6 = 0, \qquad E_0 = 0$$

$$c_4 = c_5 > 0, \qquad c_1 = c_2 = c_3 = c_6 = 0, \qquad V_0 = 0.$$

4. **Structured ALE usage.** When used with structured ALE, PART and PART set options might not make too much sense. This is because all elements inside a structured ALE mesh are assigned to one single PART ID. In the Structured ALE case, we should generate a solid set which contains those ALE boundary elements we want to prescribe hydrostatic pressures on. It is done by using the *SET_SOLID_GENERAL keyword with SALECPT option. And then use the STYPE=2 option (Solid element set ID).

Example:

Model Summary: Consider a model consisting of 2 ALE parts, air on top of water.

$$\text{H3} = \text{AMMG1} = \text{Air part above}$$

$$\text{H4} = \text{AMMG2} = \text{Water part below}$$

```
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
$ ALE materials (fluids) listed from top to bottom:
$
$ NID AT TOP OF A LAYER SURFACE         ALE MATERIAL LAYER BELOW THIS NODE
$ TOP OF 1st LAYER -------> 1681        ----------------------------------------
$                                       Air above   = PID 3 = H3 = AMMG1 (AET=4)
$ TOP OF 2nd LAYER -------> 1671        ----------------------------------------
$                                       Water below = PID 4 = H4 = AMMG2 (AET=4)
$ BOTTOM -------------------->          ----------------------------------------
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
*ALE_AMBIENT_HYDROSTATIC
$   ALESID     STYPE     VECID      GRAV     PBASE   RAMPTLC
        34         0        11   9.80665  101325.0         9
$      NID    MMGBL
      1681         1
      1671         2
*SET_PART_LIST
        34
         3         4
*ALE_MULTI-MATERIAL_GROUP
         3         1
         4         1
*DEFINE_VECTOR
$      VID        XT        YT        ZT        XH        YH        ZH       CID
        11       0.0       1.0       0.0       0.0       0.0       0.0
*DEFINE_CURVE
         9
               0.000               0.000
               0.001               1.000
              10.000               1.000
*LOAD_BODY_Y
$     LCID        SF    LCIDDR        XC        YC        ZC
         9   9.80665         0       0.0       0.0       0.0
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
```

ALE_COUPLING_NODAL_CONSTRAINT_*{OPTION}

Available options include:

<BLANK>

ID

TITLE

Purpose: This keyword activates constraint coupling between ALE materials (master) and non-ALE nodes. The slave nodes may belong to Lagrangian solid, shell, beam, thick shell, or discrete sphere (see *ELEMENT_DISCRETE_SPHERE) elements. In contrast to *ALE_COUPLING_NODAL_PENALTY, caution should be exercised in connection with EFG and SPH nodes, as meshless methods generally do not satisfy essential boundary conditions, leading to energy dissipation.

> This keyword requires a 3D ALE formulation. It is, therefore, incompatible with parts defined using *SECTION_ALE2D or *SECTION_ALE1D.

If a title is not defined LS-DYNA will automatically create an internal title for this coupling definition.

Title Card. Additional card for TITLE and ID keyword options.

Title	1	2	3	4	5	6	7	8
Variable	COUPID	TITLE						
Type	I	A70						

Card 1	1	2	3	4	5	6	7	8
Variable	SLAVE	MASTER	SSTYP	MSTYP	CTYPE	MCOUP		
Type	I	I	I	I	I	I		
Default	none	none	0	0	1	0		

Card 2	1	2	3	4	5	6	7	8
Variable	START	END				FRCMIN		
Type	F	F				F		
Default	0	1.0E10				0.5		

VARIABLE	DESCRIPTION
COUPID	Coupling (card) ID number (I10). If not defined, LSDYNA will assign an internal coupling ID based on the order of appearance in the input deck.
TITLE	A description of this coupling definition (A70).
SLAVE	Slave set ID defining a part, part set or segment set ID of the Lagrangian or slave structure (see *PART, *SET_PART or *SET_-SEGMENT). See Remark 1.
MASTER	Master set ID defining a part or part set ID of the ALE or master solid elements (see *PART or *SET_PART).
SSTYP	Slave set type of “SLAVE”: EQ.0: part set ID (PSID). EQ.1: part ID (PID). EQ.2: segment set ID (SGSID). EQ.3: node set ID (NSID).
MSTYP	Master set type of “MASTER”: EQ.0: part set ID (PSID). EQ.1: part ID (PID).
CTYPE	Coupling type: EQ.1: constrained velocity only. EQ.2: constrained acceleration and velocity.
MCOUP	Multi-material option (see Remark). EQ.0: couple with all multi-material groups,

VARIABLE	DESCRIPTION
	LT.0: MCOUP must be an integer. –MCOUP refers to a set ID of an ALE multi-material groups defined in *SET_MULTI-MATERIAL_GROUP.
START	Start time for coupling.
END	End time for coupling.
FRCMIN	Only to be used with nonzero MCOUP. Minimum volume fraction of the fluid materials included in the list of AMMGs to activate coupling. Default value is 0.5.

Remarks:

When MCOUP is a negative integer, say for example MCOUP = -123, then an ALE multi-material set-ID (AMMSID) of 123 must exist. This is an ID defined by a *SET_-MULTI-MATERIAL_GROUP_LIST card.

*ALE_COUPLING_NODAL_DRAG

Available options include:

<BLANK>

ID

TITLE

Purpose: This command provides a coupling mechanism to model the drag interaction between ALE fluids, which provide the master elements, and discrete element forms, which comprise the slave nodes. The slave nodes are assumed to be of spherical shape being either SPH elements, or discrete elements (*ELEMENT_DISCRETE_SPHERE). The coupling forces are proportional to the relative speed between the fluid and particles plus the buoyancy force due to gravitational loading.

> This keyword requires a 3D ALE formulation. It is, therefore, incompatible with parts defined using *SECTION_ALE2D or *SECTION_ALE1D.

If a title is not defined, LS-DYNA will automatically generate an internal title for this coupling definition.

Title Card. Additional card for TITLE and ID keyword options.

Title	1	2	3	4	5	6	7	8
Variable	COUPID	TITLE						
Type	I	A70						

Card 1	1	2	3	4	5	6	7	8
Variable	SLAVE	MASTER	SSTYP	MSTYP				
Type	I	I	I	I				
Default	none	none	0	0				

Card 2	1	2	3	4	5	6	7	8
Variable	START	END		FCOEF			DIRECG	GRAV
Type	F	F		F			I	F
Default	0	1.0E10		1.0			none	0.0

VARIABLE	DESCRIPTION
COUPID	Coupling (card) ID number (I10). If not defined, LSDYNA will assign an internal coupling ID based on the order of appearance in the input deck.
TITLE	A description of this coupling definition (A70).
SLAVE	Slave set ID defining a part, part set or segment set ID of the Lagrangian or slave structure (see *PART, *SET_PART or *SET_-SEGMENT).
MASTER	Master set ID defining a part or part set ID of the ALE or master solid elements (see *PART or *SET_PART, and see Remark 1).
SSTYP	Slave set type of “SLAVE”: EQ.0: part set ID (PSID). EQ.1: part ID (PID). EQ.2: segment set ID (SGSID). EQ.3: node set ID (NSID).
MSTYP	Master set type of “MASTER”: EQ.0: part set ID (PSID). EQ.1: part ID (PID).
START	Start time for coupling.
END	End time for coupling.

VARIABLE **DESCRIPTION**

FCOEF

Drag coefficient scale factor or function ID to calculate drag coefficient

GT.0: Drag coefficient scale factor.

LT.0: The absolute value of FCOEF is the Function ID of the user provided function to calculate drag coefficient; See Remark 1.

DIRECG

Gravity force direction.

EQ.1: Global x direction

EQ.2: Global y direction

EQ.3: Global z direction

GRAV

Gravity value. This value is used to calculate buoyance force.

Remarks:

1. **Drag Coupling Force.** The drag coupling force in between the particles and ALE fluids takes the following form.

$$F_{\mathrm{drag}} = c_{\mathrm{drag}} \times \tfrac{1}{2}\rho v^2 \times \tfrac{1}{4}\pi d^2$$

where c_{drag} is the drag coefficient, ρ the fluid density in which the particle is submerged, v the relative velocity between the particle and the fluid, d the diameter of the particle.

The default drag coefficient is a function of Reynolds's number and calculated by using the following formula:

$$c_{\mathrm{drag}} = \left(0.63 + \frac{4.8}{\sqrt{\mathrm{Re}}}\right)^2 .$$

Users can define their own function of drag coefficient. To do that, one needs to define a function using *DEFINE_FUNCTION and assign the negative function ID to FCOEF flag.

An example is provided below to illustrate the setup. It is equivalent to the default drag coefficient calculation.

```
*ALE_COUPLING_NODAL_DRAG
     10001         1         3         1
                                     -10                             3      9.81
*DEFINE_FUNCTION
        10
float cd(float re)
{
  float cd;
  cd=(0.63+4.8/sqrt(re))*(0.63+4.8/sqrt(re));
  if (cd > 2.0) cd = 2.0;
  return cd;
}
```

*ALE_COUPLING_NODAL_PENALTY

Available options include:

<BLANK>

ID

TITLE

Purpose: This command provides a penalty coupling mechanism between ALE materials (master) and non-ALE nodes (slave). The slave nodes may belong to Lagrangian solid, shell, beam, thick shell, or discrete (*ELEMENT_DISCRETE_SPHERE) elements. In contrast to *ALE_COUPLING_NODAL_CONSTRAINT, SPH and EFG nodes *are* supported.

> This keyword is incompatible with parts that use *SECTION_ALE2D or *SECTION_ALE1D, i.e., it requires a 3D ALE formulation.

If a title is not defined LS-DYNA will automatically create an internal title for this coupling definition.

Title Card. Additional card for TITLE and ID keyword options.

Title	1	2	3	4	5	6	7	8
Variable	COUPID	TITLE						
Type	I	A70						

Card 1	1	2	3	4	5	6	7	8
Variable	SLAVE	MASTER	SSTYP	MSTYP		MCOUP		
Type	I	I	I	I		I		
Default	none	none	0	0		0		

Card 2	1	2	3	4	5	6	7	8
Variable	START	END	PFORM	PFAC		FRCMIN		
Type	F	F	I	F		F		
Default	0	1.0E10	0	0.1		0.5		

VARIABLE	DESCRIPTION

COUPID Coupling (card) ID number (I10). If not defined, LSDYNA will assign an internal coupling ID based on the order of appearance in the input deck.

TITLE A description of this coupling definition (A70).

SLAVE Slave set ID defining a part, part set or segment set ID of the Lagrangian or slave structure (see *PART, *SET_PART or *SET_SEGMENT). See Remark 1.

MASTER Master set ID defining a part or part set ID of the ALE or master solid elements (see *PART or *SET_PART).

SSTYP Slave set type of “SLAVE”:

EQ.0: part set ID (PSID).

EQ.1: part ID (PID).

EQ.2: segment set ID (SGSID).

EQ.3: node set ID (NSID).

MSTYP Master set type of “MASTER”:

EQ.0: part set ID (PSID).

EQ.1: part ID (PID).

MCOUP Multi-material option (see Remark).

EQ.0: couple with all multi-material groups,

LT.0: MCOUP must be an integer. –MCOUP refers to a set ID of an ALE multi-material groups defined in *SET_MULTI-MATERIAL_GROUP.

START Start time for coupling.

VARIABLE **DESCRIPTION**

END End time for coupling.

PFORM Penalty stiffness formulations.

EQ.0: mass based penalty stiffness.

EQ.1: bulk modulus based penalty stiffness.

EQ.2: penalty stiffness is determined by the user-provided load curve between penetration and penalty pressure.

PFAC Penalty stiffness factor (PFORM = 0 or 1) for scaling the estimated stiffness of the interacting (coupling) system or Load Curve ID (PFORM = 2).

FRCMIN Only to be used with nonzero MCOUP. Minimum volume fraction of the fluid materials included in the list of AMMGs to activate coupling. Default value is 0.5.

Remarks:

When MCOUP is a negative integer, say for example MCOUP = -123, then an ALE multi-material set-ID (AMMSID) of 123 must exist. This is an ID defined by a *SET_-MULTI-MATERIAL_GROUP_LIST card.

*ALE_COUPLING_RIGID_BODY

Purpose: This command serves as a simplified constraint type coupling method between ALE fluids and a Lagrange rigid body.

In certain FSI simulations structure deformation is either small or not of the interest. Often times these structures are modeled as rigid bodies to shorten the simulation time and reduce the complexity. For such kind of problems, a full scale ALE/FSI simulation is costly in both simulation time and memory. This keyword provides a light-weight alternative FSI method for systems with minimal structural response.

It has a similar input format to *ALE_ESSENTIAL_BOUNDARY and maybe regarded as being an extension of the essential boundary feature. The documentation for *ALE_-ESSENTIAL_BOUNDARY_BODY applies, in large part, to this card also.

Card 1	1	2	3	4	5	6	7	8
Variable	IPID	NSID						
Type	I	I						
Default	none	none						

ALE Coupling Interfaces Cards. Include one card for each part, part set or segment to define ALE coupling interface. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	ID	IDTYPE	ICTYPE	IEXCL				
Type	I	I	I	I				
Default	none	none	1	none				

VARIABLE	DESCRIPTION
IPID	Rigid body part ID.
NSID	Node set ID defining ALE boundary nodes to follow Rigid body motion.

VARIABLE **DESCRIPTION**

ID

Set ID defining a part, part set or segment set ID of the ALE coupling interface.

IDTYPE

Type of set ID:

EQ.0: part set ID (PSID).

EQ.1: part ID (PID).

EQ.2: segment set ID (SGSID).

ICTYPE

Constraint type:

EQ.1: No flow through all directions.

EQ.2: No flow through normal direction. (slip condition)

IEXCL

Segment Set ID to be excluded from applying ALE essential boundary condition. For example, inlet/outlet segments.

Remarks:

For ICTYPE = 2, the constrained direction(s) at each surface node comes in part from knowing whether the node is a surface node, an edge node, or a corner node. If the ALE mesh boundary is identified by part(s) (IDTYPE = 0/1), edge and corner nodes are automatically detected during the segment generation process. However, this automatic detection is not foolproof for complicated geometries. Identifying the ALE mesh boundary using segment sets (IDTYPE = 2) is generally preferred for complicated geometries in order to avoid misidentification of edge and corner nodes. When segment sets are used, the edge and corner nodes are identified by their presence in multiple segment sets where each segment set describes a more or less smooth, continuous surface. The intersections of these surfaces are used to identify edge/corner nodes.

*ALE_ESSENTIAL_BOUNDARY

Purpose: This command applies and updates essential boundary conditions on ALE boundary surface nodes. Updating the boundary conditions is important if the ALE mesh moves according to *ALE_REFERENCE_SYSTEM_GROUP. If the mesh does not move, it's more correct to call it an Eulerian mesh rather than an ALE mesh, but *ALE_-ESSENTIAL_BOUNDARY can be applied nonetheless.

Certain engineering problems need to constrain the flow along the ALE mesh boundary. A simple example would be water flowing in a curved tube. Using the *ALE_ESSENTIAL_BOUNDARY approach, the tube material is not modeled and there is no force coupling between the fluid and the tube, rather the interior volume of the tube is represented by the location of the ALE mesh. Defining SPC boundary conditions with a local coordinate system at each ALE boundary node would be extremely inconvenient in such a situation. The *ALE_ESSENTIAL_BOUNDARY command applies the desired constraints along the ALE surface mesh automatically. The user only needs to specify the part(s) or segment set(s) corresponding to the ALE boundary surfaces and the type of constraint desired.

Boundary Condition Cards. Include one card for each part, part set or segment on which essential boundary conditions are applied. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	IDTYPE	ICTYPE	IEXCL				
Type	I	I	I	I				
Default	none	none	1	none				

VARIABLE	DESCRIPTION
ID	Set ID defining a part, part set or segment set ID of the ALE mesh boundary.
IDTYPE	Type of set ID: EQ.0: part set ID (PSID). EQ.1: part ID (PID). EQ.2: segment set ID (SGSID).

VARIABLE	DESCRIPTION
ICTYPE	Constraint type: EQ.1: No flow through all directions. EQ.2: No flow through normal direction. (slip condition)
IEXCL	Segment Set ID to be excluded from applying ALE essential boundary condition. For example, inlet/outlet segments.

Remarks:

For ICTYPE = 2, the constrained direction(s) at each surface node comes in part from knowing whether the node is a surface node, an edge node, or a corner node. If the ALE mesh boundary is identified by part(s) (IDTYPE = 0/1), edge/corner nodes are automatically detected during the segment generation process. However, this automatic detection is not foolproof for complicated geometries. Identifying the ALE mesh boundary using segment sets (IDTYPE = 2) is generally preferred for complicated geometries in order to avoid misidentification of edge/corner nodes. When segment sets are used, the edge/corner nodes are identified by their presence in multiple segment sets where each segment set describes a more or less smooth, continuous surface. In short, the junctures or intersections of these surfaces identify edge/corner nodes.

ALE_FAIL_SWITCH_MMG_*{OPTION}

Purpose: This card is used to allow the switching of an ALE multi-material-group ID (AMMGID) if a failure criteria is reached. If this card is not used and *MAT_VACUUM has a multi-material group in the input deck, failed ALE groups are replaced by the group for *MAT_VACUUM.

Available options include:

<BLANK>

ID

TITLE

A title for the card may be input between the 11th and 80th character on the title-ID line. The optional title line precedes all other cards for this command.

The user can explicitly define a title for this coupling.

Title Card. Additional card for the ID or TITLE options to keyword.

Title	1	2	3	4	5	6	7	8
Variable	ID	TITLE						
Type	I10	A70						

Card 1	1	2	3	4	5	6	7	8
Variable	FR_MMG	TO_MMG						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
FR_MMG	This is the AMMG-SID before the switch. The AMMG-SID corresponds to the SID defined under the *SET_MULTI-MATERIAL_GROUP_LIST (SMMGL) card. This SID points to one or more AMMGs. See Remark 1.

VARIABLE	DESCRIPTION
TO_MMG	This is the AMMG-SID after the switch. The AMMG-SID corresponds to the SID defined under the *SET_MULTI-MATERIAL_GROUP_LIST card. This SID points to one or more AMMGs. See Remark 1.

Remarks:

1. There is a correspondence between the FR_MMG and TO_MMG. Consider an example where:

   a) The FR_MMG SID points to a SID = 12 (the SID of its SMMGL card is 12, and this SID contains AMMG 1 and AMMG 2)

   b) The TO_MMG points to a SID = 34 (the SID of the SMMGL card is 34, and this SID contains AMMG 3 and AMMG 4)

   Then, AMMG 1, if switched, will become AMMG 3, and AMMG 2, if switched, will become AMMG 4.

*ALE_FRAGMENTATION

Purpose: When a material fails, this card is used to switch the failed material to vacuum. When used with FRAGTYP = 2, it can be used to model material fragmentation.

Card 1	1	2	3	4	5	6	7	8
Variable	FR_MMG	TO_MMG	FRAGTYP					
Type	I	I	I					
Default	none	none	none					

VARIABLE	DESCRIPTION
FR_MMG	This is the AMMGID of the material that just fails, before the switch.
TO_MMG	This is the AMMGID of the vacuum that the failed material is being switched to.
FRAGTYP	Flag defining whether the failed material is completely or partially switched to vacuum (see remark 1). EQ.1: Fully switch; all failed material is switched to vacuum. EQ.2: Partially switch; only the volume expansion from the last time step is switched to vacuum.

Remarks:

The Lagrange element contains only one material. Once the failure criterion is met in a Lagrange element, the whole element is marked as "failed" and either deleted or kept from further element force calculation.

However, for multi-material ALE elements, such approach is not practical as these elements are occupied by multiple materials. Failure, therefore, cannot be adequately modeled at the element level. Instead we convert the failed material inside an ALE element to vacuum. The effect is similar to element deletion in Lagrange simulations. The failed material, once switched to vacuum, is excluded from any future element force calculation.

1. **Switch to Vacuum, (FRAGTYPE = 1).** By default multi-material elements switch failed materials to vacuum. This switch involves assigning the full volume fraction of the failed material, say AMMG 1, in an element to vacuum, say AMMG 2.

   FRAGTYP = 1 is equivalent to the default treatment. However, with this card the vacuum AMMG can be explicitly specified. In the case that more than one vacuum AMMG exist, it is strongly recommended to use the FRAGTYP = 1 approach to eliminate ambiguity. It is also helpful during post-processing since it is possible to see the material interface of the switched material by assigning a dedicated vacuum AMMG to the switched material.

2. **Fragmentation, (FRAGTYPE = 2).** FRAGYP = 2 models material fragmentation. Note that the FRAGTYP = 1 approach leads to loss of mass and, consequently, dissipates both momentum and energy. With FRAGTYP = 2, instead of converting the full volume of the failed material to vacuum, LS-DYNA only converts the material expansion to vacuum. This approach conserves mass and, therefore, momentum and energy.

   To illustrate how this fragmentation model works, consider a tension failure example. At the time step when the material fails, LS-DYNA calculates the material expansion in the current step and converts this volume to vacuum. The stresses and other history variables are left unchanged, so that in the next time step it will again fail. The expansion in the next time step will be also converted to vacuum. This process continues until maybe at a later time the gap stops growing or even starts to close due to compression.

Example:

Consider a simple bar extension example:

FR_MMG:	H5	=	AMMG1	=	Metal bar
	H6	=	AMMG2	=	Ambient air
TO_MMG:	H7	=	AMMG3	=	Dummy vacuum part

```
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
*ALE_FRAGMENTATION
$   FR_MMG    TO_MMG   FRAGTYP
         1         3         2
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
```

*ALE_FSI_PROJECTION

Purpose: This card provides a coupling method for simulating the interaction between a Lagrangian material set (structure) and ALE material set (fluid). The nearest ALE nodes are projected onto the Lagrangian structure surface at each time step. This method does not conserve energy, as mass and momentum are transferred via constrained based approach.

Card 1	1	2	3	4	5	6	7	8
Variable	LAGSID	ALESID	LSIDTYP	ASIDTYP	SMMGID	ICORREC	INORM	
Type	I	I	I	I	I	I	I	
Default	0	0	0	0	0	0	0	

Card 2	1	2	3	4	5	6	7	8
Variable	BIRTH	DEATH						
Type	F	F						
Default	0.0	1.E+10						

VARIABLE	DESCRIPTION
LAGSID	A set ID defining the Lagrangian part(s) for this coupling (structures).
ALESID	A set ID defining the ALE part(s) for this coupling (fluids).
LSIDTYP	Lagrangian set ID type EQ.0: Part set ID (PSID), EQ.1: Part ID (PID).
ASIDTYP	ALE set ID type EQ.0: Part set ID (PSID), EQ.1: Part ID (PID).

VARIABLE	DESCRIPTION
SMMGID	A set ID referring to a group of one or more ALE-Multi-Material-Group (AMMG) IDs which represents the ALE materials interacting with the Lagrangian structure. This SMMGID is a set ID defined by *SET_MULTI-MATERIAL_GROUP_LIST.
ICORREC	Advection error correction method (See Remark 1). EQ.1: ALE mass is conserved. Leaked mass is moved, EQ.2: ALE mass is almost conserved, EQ.3: No correction performed (default). ALE mass is conserved. Some leakage may occur. This may be the best solution.
INORM	Type of coupling. EQ.0: Couple in all directions, EQ.1: Couple in compression and tension (free sliding), EQ.2: Couple in compression only (free sliding). This choice requires ICORREC = 3.
BIRTH	Start time for coupling.
DEATH	End time for coupling.

Remarks:

1. As the ALE nodes are projected onto the closest Lagrangian surface, there may be some advection errors introduced. These errors may result in a small element mass fraction being present on the "wrong" side of the coupled Lagrangian surface. There are 3 possible scenarios:

   a) Mass on the wrong side of the Lagrangian structure may be moved to the right side. This may cause P oscillations. No leakage will occur.

   b) Mass on the wrong side is deleted. Mass on the right side is scaled up to compensate for the lost mass. No leakage will occur.

   c) Mass on the wrong side is allowed (no correction performed). Some leakage may occur. This may be the most robust and simplest approach.

Example:

Model Summary:

H1 = AMMG1 = background air mesh
H2 = AMMG1 = background air mesh
S3 = cylinder containing AMMG2
S4 = dummy target cylinder for impact

The gas inside S3 is AMMG2. S3 is given an initial velocity and it will impact S4.

```
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
*ALE_MULTI-MATERIAL_GROUP
         1         1
         2         1
*SET_MULTI-MATERIAL_GROUP_LIST
        22
         2
*ALE_FSI_PROJECTION
$   LAGSID    ALESID   LSIDTYP   ASIDTYP    SMMGID   ICORREC     INORM
         3         1         1         1        22         3         2
$    BIRTH     DEATH
       0.0      20.0
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
```

ALE_FSI_SWITCH_MMG_*{OPTION}

Purpose: This card is used to allow the switching of an ALE multi-material-group ID (AMMGID) of a fluid as that fluid passes across a monitoring surface. This monitoring surface may be a Lagrangian shell structure, or a segment set. It does not have to be included in the slave set of the coupling card: *CONSTRAINED_LAGRANGE_IN_SOLID. However, at least one coupling card must be present in the model.

Available options include:

<BLANK>

ID

TITLE

An ID number (up to 8 digits) may be defined for this switch command in the first 10-character space. A title for the card may be input between the 11th and 80th character on the title-ID line. The optional title line precedes all other cards for this command.

The user can explicitly define a title for this coupling.

Title Card. Additional card for the Title or ID keyword options.

Title	1	2	3	4	5	6	7	8
Variable	ID	TITLE						
Type	I10	A70						

Card 1	1	2	3	4	5	6	7	8
Variable	SID	STYPE	NQUAD	XOFF	BTIME	DTIME	NFREQ	NFOLD
Type	I	I	I	F	F	F	I	I
Default	none	0	1	0.0	0.0	1.0E20	1	0

Card 2	1	2	3	4	5	6	7	8
Variable	FR_MMG	TO_MMG	XLEN					
Type	I	I	F					
Default	none	none	0.0					

VARIABLE	DESCRIPTION
SID	A set ID defining a monitoring surface over which an ALE fluid flows across, and its ALE multi-material-group-ID (AMMGID) is switched. The monitoring surface may be a Lagrangian shell structure, or a segment set. This surface, if Lagrangian, does not have to be included in the coupling definition (see remark 4).
STYPE	Set ID type of the above SID. EQ.0: Part set ID (PSID) (default). EQ.1: Part ID (PID). EQ.2: Segment set ID (SGSID).
NQUAD	The number of flow-sensor points to be distributed over each monitoring surface/segment. There should be enough sensor points distributed to monitor the flow in each ALE element intersected by this monitoring surface (default = 1, see remark 3).
XOFF	An offset distance away from the monitoring surface, beyond which the AMMGID switching occurs. The direction of XOFF is defined by the normal vector of the monitoring segment. This offset distance, in general, should be at least 2 ALE element widths away from, and beyond the monitoring interface (default = 0.0).
BTIME	Start time for the AMMGID switch to be activated (default = 0.0).
DTIME	Ending time for the AMMGID switch (default = 1.0E20).
NFREQ	Number of computational cycles between ALE switch check (default = 1).

VARIABLE	DESCRIPTION
NFOLD	Flag for checking folding logic (default = 0, ⇒ off). If NFOLD = 1 (⇒ on), then LS-DYNA will check if the monitoring segment is in the fold, applicable to airbag. If the monitoring segment is still located within a folded (shell) region, then no switching is allowed yet until it has unfolded.
FR_MMG	This is the AMMG-SID before the switch. The AMMG-SID corresponds to the SID defined under the *SET_MULTI-MATERIAL_GROUP_LIST (SMMGL) card. This SID points to one or more AMMGs. See Remark 1.
TO_MMG	This is the AMMG-SID after the switch. The AMMG-SID corresponds to the SID defined under the *SET_MULTI-MATERIAL_GROUP_LIST card. This SID points to one or more AMMGs. See Remark 1.
XLEN	This is an absolute distance for distributing the flow sensor points over the ALE elements. To make sure that at least 1 sensor point, defined on each Lagrangian segment, is present in each ALE element to track the flow of an AMMG, XLEN may be estimated as roughly half the length of the smallest ALE element in the mesh. See Remark 3.

Remarks:

1. There is a correspondence between the FR_MMG and TO_MMG. Consider an example where:

   a) The FR_MMG SID points to a SID = 12 (the SID of its SMMGL card is 12, and this SID contains AMMG 1 and AMMG 2)

   b) The TO_MMG points to a SID = 34 (the SID of the SMMGL card is 34, and this SID contains AMMG 3 and AMMG 4)

   Then, AMMG 1, if switched, will become AMMG 3, and AMMG 2, if switched, will become AMMG 4.

2. The ID option must be activated if the parameter SWID is used in the *DATABAS_FSI card. Then the accumulated mass of an AMMG that goes through a tracking surface, and being switched, will be reported via the parameter "mout" in the "dbfsi" ASCII output file (or equivalently the "POROSITY" parameter inside LS-PrePost ASCII plotting option).

3. When both NQUAD and XLEN are defined, whichever gives smaller sensor-point interval distance will be used. XLEN may give better control as in the case of a null shell acting as the monitoring surface. As this null shell is stretched, NQUAD distribution of sensor-points may not be adequate, but XLEN would be.

4. The monitoring surface does not have to be included in the slave set of the coupling card. However, at least one coupling card must be present in the model. The monitoring segment set can be made up of Lagrangian or ALE nodes.

Example:

Consider a simple airbag model with 3 part IDs:

H25 (AMMG1) = Inflator gas injected into the airbag.
H24 (AMMG2) = Air outside the airbag (background mesh).
H26 (AMMG3) = Dummy AMMG of inflator gas after it passes through a vent hole.
S9 = A Lagrangian shell part representing a vent hole.
S1 = A Lagrangian shell part representing the top half of an airbag.
S2 = A Lagrangian shell part representing the bottom half of an airbag.

The inflator gas inside the airbag is distinguished from the inflator gas that has passed through the monitoring surface (vent hole) to the outside of the airbag by assigning different ALE multi-material group set ID to each. The dummy fluid part (H26) should have the same material and EOS model IDs as the before-switched fluid (H25).

TO_MMG = 125
⇒ AMMGID (before switch) = *SET_MULTI-MATERIAL_GROUP_LIST(125) = 1
⇒ PART = PART(AMMGID1) = H25

FR_MMG = 126
⇒ AMMGID (before switch) = *SET_MULTI-MATERIAL_GROUP_LIST(126) = 3
⇒ PART = PART(AMMGID3) = H26

```
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
*ALE_MULTI-MATERIAL_GROUP
        25         1
        24         1
        26         1
*DATABASE_FSI
$     TOUT                                  [STYPE: 0=PSID ; 1=PID ; 2=SGSID]
    0.1000
$ DBFSI_ID       SID     STYPE  AMMGSWID  LDCONVID
         1         1         1
         2         2         1
         3         9         1     90000
```

```
*SET_MULTI-MATERIAL_GROUP_LIST
       125
         1
*SET_MULTI-MATERIAL_GROUP_LIST
       126
         3
*ALE_FSI_SWITCH_MMG_ID
     90000
$      SID   SIDTYPE     NQUAD      XOFF     BTIME     DTIME     NFREQ      FOLD
         9         1         3     -20.0       5.0       0.0         1         1
$   Fr_MMG    To_MMG     XCLEN
       125       126        5.
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
```

Note:

1. The *DATABASE_FSI card tracks 3 surface entities: (a) top half of an airbag, (b) bottom half of an airbag, and (c) the vent hole monitoring surface where the AMMGID of the inflator gas is switched.

2. The amount of mass passing through the vent hole during the switch is output to a parameter called “pleak” in a “dbfsi” ASCII file. See *DATABASE_FSI.

3. The *ALE_FSI_SWITCH_MMG_ID card track any flow across S9 and switch the AMMGSID from 125 (AMMG 1) to 126 (AMMG 3).

*ALE_FSI_TO_LOAD_NODE

Purpose: Output the ALE coupling forces from *CONSTRAINED_LAGRANGE_IN_SOLID, CTYPE = 4 in keyword format, so they may be applied directly in another run.

Card 1	1	2	3	4	5	6	7	8
Variable	DT	NSID	IOPT					
Type	I	I	I					
Default	none	none	0					

VARIABLE **DESCRIPTION**

DT Output intervals

NSID Node Set ID. See *SET_NODE.

IOPT Options to map the coupling data between 2 runs:

EQ.0: The keyword alefsiloadnode.k (See Remark 1) is created at the end of the run by LS-DYNA.

EQ.1: A database of coupling forces is dumped without the conversion in keyword file at the end of the run (See Remark 2). The database can be treated by a program (alefsiloadnode.exe) to write alefsiloadnode.k.

EQ.2: The database of coupling forces created by IOPT = 1 (See Remark 2) is read back. The structure meshes should be identical. The forces are directly applied the nodes without using *LOAD_NODE. The parameters DT and NSID are not read.

EQ.3: A database of coupling accelerations is dumped at the end of the run (See Remark 3).

EQ.4: The database of coupling accelerations created by IOPT = 3 (See Remark 3) is read back. The structure meshes can be different. The accelerations are interpolated at the nodes provided by NSID. The parameters DT and NSID are read.

Remarks:

1. The name of the output keyword file is alefsiloadnode.k. For each node, this file contains three *LOAD_NODE for each global direction and three *DEFINE_CURVE for the coupling force histories.

2. The name of the database is alefsi2ldnd.tmp (or alefsi2ldnd.tmp00… in MPP). It should be in the directory of the 2nd run for IOPT = 2. The database lists the coupling forces by node. The structure meshes (and their MPP decomposition) for the IOPT = 1 and IOPT = 2 runs should be the same.

3. The names of the databases are alefsi2ldnd.tmp (or alefsi2ldnd.tmp00… in MPP) and alefsi2lndndx.tmp. They should be in the directory of the 2nd run for IOPT = 4. The file alefsi2ldnd.tmp lists the coupling accelerations by node (coupling acceleration = coupling force / nodal mass). The file alefsi2lndndx.tmp lists the initial nodal coordinates and coupling segment connectivities (see *CONSTRAINED_LAGRANGE_IN_SOLID for coupling segment). The structure meshes for the IOPT = 3 and IOPT = 4 runs can be different. The IOPT = 3 initial geometry stored in alefsi2lndndx.tmp is used to interpolate the coupling accelerations (saved in alefsi2ldnd.tmp) at the nodes provided by NSID. For the interpolation to work, these nodes should be within the IOPT = 3 coupling segment initial locations.

*ALE_MULTI-MATERIAL_GROUP

Purpose: This command defines the appropriate ALE material groupings for interface reconstruction when two or more ALE Multi-Material Groups (AMMG) are present in a model. This card is required when ELFORM = 11 in the *SECTION_SOLID card or when ALEFORM = 11 in *SECTION_ALE1D or *SECTION_ALE2D. Each data line represents one ALE multi-material group (AMMG), with the first line referring to AMMGID 1, second line AMMGID 2, etc. Each AMMG represents one unique "fluid" which may undergo interaction with any Lagrangian structure in the model.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	IDTYPE						
Type	I	I						
Default	none	0						
Remarks	1							

VARIABLE	DESCRIPTION
SID	Set ID.
IDTYPE	Set type: EQ.0: Part set, EQ.1: Part.

Remarks:

1. When ELFORM = 12 in the *SECTION_SOLID card (single material and void), this card should not be used. In one model, ELFORM = 12 cannot be used together with ELFORM = 11. If possible, it is recommended that ELFORM = 11 be used as it is the most robust and versatile formulation for treating multi-material ALE parts.

2. Each AMMG is automatically assigned an ID (AMMGID), and consists of one or more PART ID's. The interface of each AMMGID is reconstructed as it evolves dynamically. Each AMMGID is represented by one material contour color in LS-PrePost.

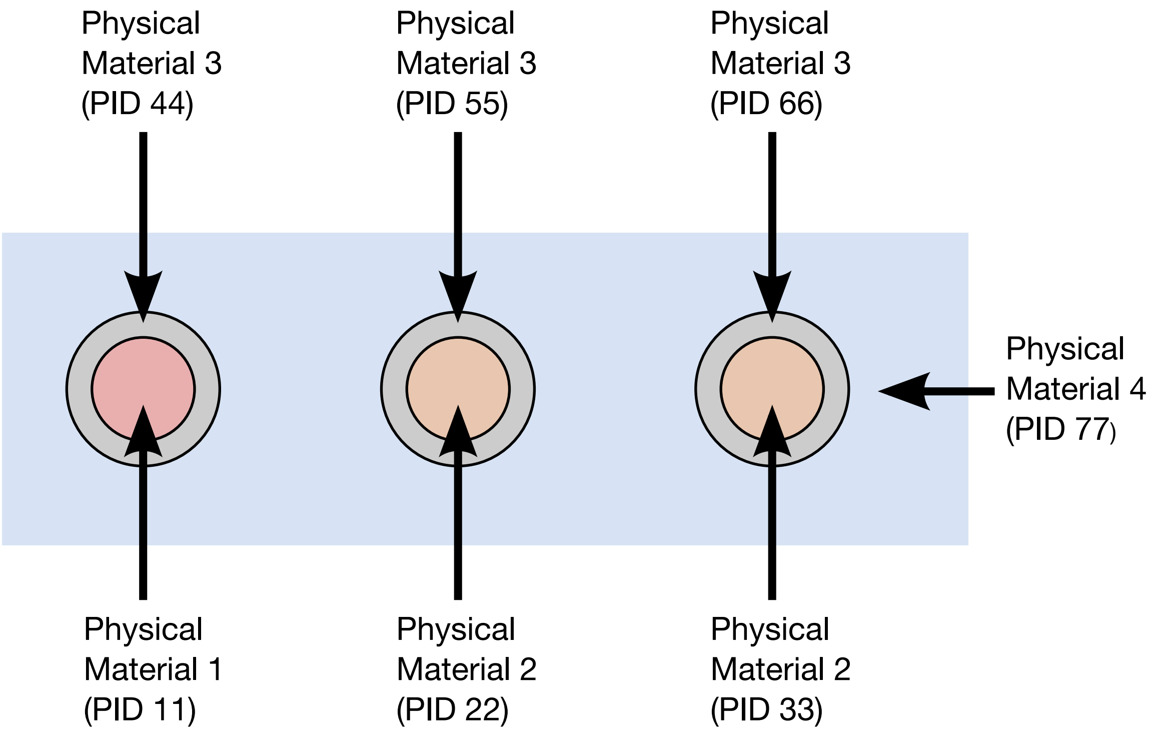


Figure 4-1. Schematic illustration of Example 1.

3. The maximum number of AMMGIDs allowed has been increased to 20. However, there may be 2, at most 3, AMMGs inside an ALE element at anytime. If there are more than 3 AMMGs inside any 1 ALE element, the ALE mesh needs refinement. Better accuracy is obtained with 2 AMMGs in mixed elements.

4. To plot these AMMGIDs in LS-PrePost:

   [FCOMP] ⇒ [MISC] ⇒ [VOLUME FRACTION OF AMMGID #] ⇒ [APPLY]

   (Note: Contour definitions maybe different for gas mixture application)

5. It is very important to distinguish among the

   a) Physical materials,

   b) PART IDs, and

   c) AMMGIDs.

   A *PART may be any mesh component. In ALE formulation, it is simply a geometric entity and a time = 0 concept. This means a *PART may be a mesh region that can be filled with one or more AMMGIDs at time zero, via a volume filling command (*INITIAL_VOLUME_FRACTION_GEOMETRY). An AMMGID represents a physical material group which is treated as one material entity (represented by 1 material color contour in LS-PrePost plotting). AMMGID is used in dealing with multiple ALE or Eulerian materials. For example, it can be used to specify a master ALE group in a coupling card.

Example 1:

Consider a purely Eulerian model containing 3 containers containing 2 different physical materials (fluids 1 and 2). All surrounded by the background material (maybe air). The containers are made of the same material, say, metal. Assume that these containers explode and spill the fluids. We want to track the flow and possibly mixing of the various materials. Note that all 7 parts have ELFORM = 11 in their *SECTION_-SOLID cards. So we have total of 7 PIDs, but only 4 different physical materials. See Figure 4-1.

Approach 1: If we want to track only the interfaces of the **physical** materials.

```
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
*SET_PART
         1
        11
*SET_PART
         2
        22        33
*SET_PART
         3
        44        55        66
*SET_PART
         4
        77
*ALE_MULTI-MATERIAL_GROUP
         1         0   ←  1st line  =  1st AMMG  ⇒ AMMGID = 1
         2         0   ←  2nd line  =  2nd AMMG  ⇒ AMMGID = 2
         3         0   ←  3rd line  =  3rd AMMG  ⇒ AMMGID = 3
         4         0   ←  4th line  =  4th AMMG  ⇒ AMMGID = 4
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
```

With this approach, we define only 4 AMMGs (NALEGP = 4). So in LS-PrePost, when plotting the material-group (history variable) contours, we will see 4 colors, one for each material group. One implication is that when the fluids from part 22 and part 33 flow into the same element, they will coalesce and no boundary distinction between them is maintained subsequently. While this may be acceptable for fluids at similar thermodynamic states, this may not be intuitive for solids. For example, if the solid container materials from parts 44, 55 and 66 flow into one element, they will coalesce "like a single fluid", and no interfaces among them are tracked. If this is undesirable, an alternate approach may be taken. It is presented next.

Approach 2: If we want to reconstruct as many interfaces as necessary, in this case, we follow the interface of each part.

```
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
*ALE_MULTI-MATERIAL_GROUP
         1         1  ←  1st line  =  1st AMMG  ⇒ AMMGID = 1
         2         1  ←  2nd line  =  2nd AMMG  ⇒ AMMGID = 2
         3         1  ←  3rd line  =  3rd AMMG  ⇒ AMMGID = 3
         4         1  ←  4th line  =  4th AMMG  ⇒ AMMGID = 4
         5         1  ←  4th line  =  5th AMMG  ⇒ AMMGID = 5
         6         1  ←  4th line  =  6th AMMG  ⇒ AMMGID = 6
         7         1  ←  4th line  =  7th AMMG  ⇒ AMMGID = 7
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
```

There are 7 AMMGs in this case (NALEGP = 7). This will involve more computational cost for the additional tracking. Realistically, accuracy will be significantly reduced if there are more than 3 or 4 materials in any one element. In that case, higher mesh resolution may be required.

Example 2:

Oil	Water	Air
Group 1	Group 2	Group 3
Part IDs 1 and 2	Part ID 3	Part IDs 5, 6, and 7

The above example defines a mixture of three groups of materials (or “fluids”), oil, water and air, that is, the number of ALE multi-material groups (AMMGs) NALEGP = 3.

The first group contains two parts (materials), part ID's 1 and 2.

The second group contains one part (material), part ID 3.

The third group contains three parts (materials), part ID's 5, 6 and 7.

*ALE_REFERENCE_SYSTEM_CURVE

Purpose: This command defines a motion and/or a deformation prescribed for a geometric entity, where a geometric entity may be any part, part set, node set, or segment set. The motion or deformation is completely defined by the 12 parameters shown in the equation below. These 12 parameters are defined in terms of 12 load curves. This command is required only when PRTYPE = 3 in the *ALE_REFERENCE_-SYSTEM_GROUP command.

Card 1	1	2	3	4	5	6	7	8
Variable	ID							
Type	I							
Default	none							

Card 2	1	2	3	4	5	6	7	8
Variable	LCID1	LCID2	LCID3	LCID4	LCID5	LCID6	LCID7	LCID8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

Card 3	1	2	3	4	5	6	7	8
Variable	LCID9	LCID10	LCID11	LCID12				
Type	I	I	I	I				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
ID	Curve group ID.
LCID1, …, LCID12	Load curve IDs.

Remarks:

1. The velocity of a node at coordinate (x, y, z) is defined as:

$$\begin{Bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_5 \\ f_9 \end{Bmatrix} + \begin{bmatrix} f_2 & f_3 & f_4 \\ f_6 & f_7 & f_8 \\ f_{10} & f_{11} & f_{12} \end{bmatrix} \begin{Bmatrix} x - \mathrm{XC} \\ y - \mathrm{YC} \\ z - \mathrm{ZC} \end{Bmatrix}$$

where $f_1(t)$ is the value of load curve LCID1 at time t, $f_2(t)$ is the value of load curve LCID2 at time t and so on. The functions $f_1(t)$, $f_5(t)$, and $f_9(t)$ respectively correspond to the translation components in global x, y, and z direction, while the functions $f_2(t)$, $f_7(t)$, and $f_{12}(t)$ correspond to and expansion or contraction along the x, y, and z axes.

The parameters XC, YC and ZC from the second data card of *ALE_REFERENCE_SYSTEM_GROUP define the center of rotation and expansion of the mesh. If the mesh translates, the center position is updated with $f_1(t)$, $f_5(t)$, and $f_9(t)$.

If LCID8, LCID10, LCID3 are equal to -1, their corresponding values $f_8(t)$, $f_{10}(t)$, and $f_3(t)$ will be equal to $-f_{11}(t)$, $-f_4(t)$, and $-f_6(t)$ so as to make a skew symmetric matrix thereby inducing a rigid rotation of the mesh about the axis **v** defined by the triple,

$$\mathbf{v} = \big(f_{11}(t), f_4(t), f_6(t)\big)$$

Example:

Consider a motion that consists of translation in the x and y direction only. Thus only $f_1(t)$ and $f_5(t)$ are required. Hence only 2 load curve ID's need be defined:

```
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|...8
*ALE_REFERENCE_SYSTEM_GROUP
$      SID     STYPE     PRTYP      PRID    BCTRAN     BCEXP     BCROT    ICOORD
         1         0         3        11         0         7         0
$       XC        YC        ZC    EXPLIM
         0         0         0         0
*ALE_REFERENCE_SYSTEM_CURVE
$ CURVESID
        11
$    LCID1     LCID2     LCID3     LCID4     LCID5     LCID6     LCID7     LCID8
       111         0         0         0       222         0         0         0
$    LCID9    LCID10    LCID11    LCID12
         0         0         0         0
*DEFINE_CURVE
$     lcid      sidr       sfa       sfo      offa      offo    dattyp
       111
$                 a1                  o1
                0.00                 5.0
                0.15                 4.0
*DEFINE_CURVE
$     lcid      sidr       sfa       sfo      offa      offo    dattyp
       222
$                 a1                  o1
                0.00                -1.0
```

```
                    0.15                     -5.0
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
```

*ALE_REFERENCE_SYSTEM_GROUP

Purpose: This card is used to associate a geometric entity to a reference system type. A geometric entity may be any part, part set, node set, or segment set of a model (or a collection of meshes). A reference system type refers to the possible transformation allowed for a geometric entity (or mesh). This command defines the type of reference system or transformation that a geometric entity undergoes. In other words, it prescribes how certain mesh can translate, rotate, expand, contract, or be fixed in space, etc.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	STYPE	PRTYPE	PRID	BCTRAN	BCEXP	BCROT	ICR/NID
Type	I	I	I	I	I	I	I	I
Default	none	0	0	0	0	0	0	0

Card 2	1	2	3	4	5	6	7	8
Variable	XC	YC	ZC	EXPLIM	EFAC		FRCPAD	IEXPND
Type	F	F	F	F	F		F	I
Default	0.0	0.0	0.0	inf.	0.0		0.1	0

Remaining cards are optional.†

Card 3	1	2	3	4	5	6	7	8
Variable	IPIDXCL	IPIDTYP						
Type	I	I						
Default	0	0						

VARIABLE **DESCRIPTION**

SID Set ID.

STYPE Set type:

EQ.0: part set,

EQ.1: part,

EQ.2: node set,

EQ.3: segment set.

PRTYPE Reference system type (See Remark 1):

EQ.0: Eulerian,

EQ.1: Lagrangian,

EQ.2: Normal ALE mesh smoothing,

EQ.3: Prescribed motion following load curves, see *ALE_REFERENCE_SYSTEM_CURVE,

EQ.4: Automatic mesh motion following mass weighted average velocity in ALE mesh,

EQ.5: Automatic mesh motion following a local coordinate system defined by three user defined nodes, see *ALE_REFERENCE_SYSTEM_NODE,

EQ.6: Switching in time between different reference system types, see *ALE_REFERENCE_SYSTEM_SWITCH,

EQ.7: Automatic mesh expansion in order to enclose up to twelve user defined nodes, see *ALE_REFERENCE_SYSTEM_NODE.

EQ.8: Mesh smoothing option for shock waves, where the element grid contracts in the vicinity of the shock front: this may be referred to as the Delayed-ALE option. It controls how much the mesh is to be moved during the remap step. This option requires the definition of the 5th parameter in the 2nd card, EFAC; see below for definition.

EQ.9: Allowing the ALE mesh(es) to:

1. Translate and/or rotate to follow a local Lagrangian reference coordinate system (whose *ALE_REFERENCE_SYSTEM_NODE card ID is defined by the BCTRAN parameter)

VARIABLE	DESCRIPTION
	2. Expand or contract to enclose a Lagrangian part-set ID defined by the PRID parameter. 3. Has a Lagrangian node ID be defined by the ICR/NID parameter to be the center of the ALE mesh expansion.
PRID	A parameter giving additional information depending on the reference system (PRTYPE) choice: PRTYPE.EQ.3: PRID defines a load curve group ID specifying an *ALE_REFERENCE_SYSTEM_CURVE card for mesh translation. This defines up to 12 curves which prescribe the motion of the system. PRTYPE.EQ.4: PRID defines a node set ID (*SET_NODE), for which a mass average velocity is computed. This velocity controls the mesh motion. PRTYPE.EQ.5: PRID defines a node group ID specifying an *ALE_REFERENCE_SYSTEM_NODE card, via which, three nodes forming a local coordinate system are defined. PRTYPE.EQ.6: PRID defines a switch list ID specifying an *ALE_REFERENCE_SYSTEM_SWITCH card. This defines the switch times and the reference system choices for each time interval between the switches. PRTYPE.EQ.7: PRID defines a node group ID specifying an *ALE_REFERENCE_SYSTEM_NODE card. Up to 12 nodes in space forming a region to be enveloped by the ALE mesh are defined. PRTYPE.EQ.9: PRID defines a Lagrangian part set ID (PSID) defining the Lagrangian part(s) whose range of motion is to be enveloped by the ALE mesh(es). This is useful for airbag modeling.

If PRTYPE.EQ.4 or PRTYPE.EQ.5, then

BCTRAN	BCTRAN is a translational constraint (Remark 3). EQ.0: no constraints, EQ.1: constrained x translation,

VARIABLE **DESCRIPTION**

EQ.2: constrained y translation,

EQ.3: constrained z translation,

EQ.4: constrained x and y translation,

EQ.5: constrained y and z translation,

EQ.6: constrained z and x translation,

EQ.7: constrained x, y, and z translation.

Else If PRTYPE.EQ.9, then

BCTRAN BCTRAN is a node group ID from a *ALE_REFERENCE_SYSTEM_NODE card prescribing a local coordinate system (3 node IDs) whose motion is to be followed by the ALE mesh(es).

Else

BCTRAN Ignored

End if

BCEXP For PRTYPE = 4 & 7 BCTRAN is an expansion constraint, otherwise it is ignored (Remark 3).

EQ.0: no constraints,

EQ.1: constrained x expansion,

EQ.2: constrained y expansion,

EQ.3: constrained z expansion,

EQ.4: constrained x and y expansion,

EQ.5: constrained y and z expansion,

EQ.6: constrained z and x expansion,

EQ.7: constrained x, y, and z expansion.

BCROT BCROT is a rotational constraint (Remark 3). Otherwise, BCROT is ignored.

EQ.0: no constraints,

EQ.1: constrained x rotation,

EQ.2: constrained y rotation,

VARIABLE DESCRIPTION

EQ.3: constrained z rotation,

EQ.4: constrained x and y rotation,

EQ.5: constrained y and z rotation,

EQ.6: constrained z and x rotation,

EQ.7: constrained x, y, and z rotation.

If PRTYPE.EQ.4

ICR/(NID) ICR is a flag the specifies the method LS-DYNA uses for determining the center point for expansion and rotation (Remark 3).

EQ.0: The center is at center of gravity of the ALE mesh.

EQ.1: The center is at (XC, YC, ZC), just a point in space (it does not have to be a defined node)

Else if PRTYPE.EQ.9

(ICR)/NID NID sets the Lagrangian node ID for the node that anchors the center of ALE mesh expansion (Remark 2).

End if

XC, YC, ZC Center of mesh expansion and rotation for PRTYPE = 4 and 5, otherwise ignored. See ICR above.

EXPLIM Limit ratio for mesh expansion and contraction. Each Cartesian direction is treated separately. The distance between the nodes is not allowed to increase by more than a factor EXPLIM, or decrease to less than a factor 1/EXPLIM. This flag applies only for PRTYPE = 4, otherwise it is ignored.

EFAC Mesh remapping factor for PRTYPE = 8 only, otherwise it is ignored. EFAC is allowed to range between 0.0 and 1.0. When EFAC approaches 1.0, the remapping approaches the Eulerian behavior.

The smaller the value of EFAC, the closer the mesh will follow the material flow in the vicinity of a shock front, i.e. approaching Lagrangian behavior.

Note that excessively small values for EFAC can result in severe

VARIABLE **DESCRIPTION**

mesh distortions as the mesh follows the material flow. As time evolves, the mesh smoothing behavior will approach that of an Eulerian system.

FRCPAD

For PRTYPE = 9 this is an ALE mesh padding fraction, otherwise it is ignored.

FRCPAD is allowed to range from 0.01 to 0.2. If the characteristic Lagrange mesh dimension, dL_1, exceeds

$$(1 - 2 \times \text{FRCPAD}) \times dL_A,$$

where dL_A is the characteristic length of the ALE mesh, then the ALE mesh is expanded so that

$$dL_A = \frac{dL_1}{1 - 2 \times \text{FRCPAD}}.$$

This provides an extra few layers of ALE elements beyond the maximum Lagrangian range of motion.

EQ.0.01: $dL_A = {}^{dL_L}/_{0.98} = dL_L \times 1.020408$

EQ.0.20: $dL_A = {}^{dL_L}/_{0.60} = dL_L \times 1.666667$

IEXPND

For PRTYPE = 9 this is an ALE mesh expansion control flag, otherwise it is ignored.

EQ.0: Both mesh expansion and contraction are allowed.

EQ.1: Only mesh expansion is allowed:

IPIDXCL

An ALE set ID to be excluded from the expansion and/or contraction only. Translation and rotation are allowed. For example, this may be used to prevent the ALE mesh (or part) at the inflator gas inlet region from expanding too much. High ALE mesh resolution is usually required to resolve the high speed flow of the gas into the airbag via point sources (See Remark 2).

IPIDTYPE

Set ID type of IPIDXCL:

EQ.0: PSID

EQ.1: PID

Remarks:

1. **Required Associated Cards.** Some PRTYP values may require a supplemental definition defined via corresponding PRID. For example, PRTYP = 3 requires a *ALE_REFERENCE_SYSTEM_CURVE card. If PRID = n, then in the corresponding *ALE_REFERENCE_SYSTEM_CURVE card, ID = n. Similar association applies for any PRTYP (i.e. 3, 5, 6, or 7) which requires a definition for its corresponding PRID parameter.

2. **Mesh Centering.** For PRTYPE = 9: ICR/NID can be useful to keep a high density ALE mesh centered on the region of greatest interest, (such as the inflator orifices region in an airbag model). For example, in the case of nonsymmetrical airbag deployment, assuming that the ALE mesh is initially finer near the inlet orifices, and gradually coarsened away from it. Defining an "anchor node" at the center of the orifice location will keep the fine ALE mesh region centered on the orifice region. So that this fine ALE mesh region will not be shifted away (from the point sources) during expansion and translation. The ALE mesh can move and expand outward to envelop the Lagrangian airbag in such a way that the inlet is well resolved throughout the deployment.

3. **Additional Constraints.** The table below shows the applicability of the various choices of PRTYPE. Simple deductions from the functional definitions of the PRTYPE choices will clarify the applications of the various constraints. For example, when PRTYP = 3, nodal motion of the ALE mesh is completely controlled by the 12 curves. Therefore, no constraints are needed.

PRTYPE	ICR/NID	BCTRAN	BCROT	BCEXP
3	NO	NO	NO	NO
4	YES (ICR)	YES	YES	YES
5	NO	YES	NO	NO
6	NO	NO	NO	NO
7	NO	NO	NO	YES
8	NO	NO	NO	NO
9	YES (NID)	YES (NGID)	NO	NO

Example 1:

Consider a bird-strike model containing 2 ALE parts: a bird is surrounded by air (or void). A part-set ID 1 is defined containing both parts. To allow for the meshes of these

2 parts to move with their combined mass-weighted-average velocity, PRTYPE = 4 is used. Note that BCEXP = 7 indicating mesh expansion is constrained in all global directions.

```
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|...8
*ALE_REFERENCE_SYSTEM_GROUP
$      SID     STYPE     PRTYP      PRID    BCTRAN     BCEXP     BCROT    ICOORD
         1         0         4         0         0         7         0
$       XC        YC        ZC    EXPLIM
         0         0         0         0
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|...8
```

Example 2:

Consider a bouncing ball model containing 2 ALE parts: a solid ball (PID 1) is surrounded by air or void (PID 2). A part-set ID 1 is defined containing both parts. To allow for the meshes of these 2 parts to move with 2 reference system types: (a) first, they move with their combined mass-weighted-average velocity between 0.0 and 0.01 second; and subsequently (between 0.01 and 10.0 seconds) their reference system is switched to (b) an Eulerian system (thus the mesh is fixed in space), a reference system "SWITCH" is required. This is done by setting PRTYPE = 6. This PRTYPE requires a corresponding *ALE_REFERENCE_SYSTEM_SWITCH card. Note that PRID = 11 in the *ALE_REFERENCE_SYSTEM_GROUP card corresponds to the SWITCHID = 11 in *ALE_REFERENCE_SYSTEM_SWITCH card.

```
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
*ALE_REFERENCE_SYSTEM_GROUP
$      SID     STYPE     PRTYP      PRID    BCTRAN     BCEXP     BCROT    ICOORD
         1         0         6        11         0         7         7
$       XC        YC        ZC    EXPLIM   EULFACT SMOOTHVMX
         0         0         0         0       0.0
*ALE_REFERENCE_SYSTEM_SWITCH
$ SWITCHID
        11
$       t1        t2        t3        t4        t5        t6        t7
      0.01      10.0
$    TYPE1     TYPE2     TYPE3     TYPE4     TYPE5     TYPE6     TYPE7     TYPE8
         4         0
$      ID1       ID2       ID3       ID4       ID5       ID6       ID7       ID8
         0         0         0         0         0         0         0         0
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
```

*ALE_REFERENCE_SYSTEM_NODE

Purpose: This command defines a group of nodes that control the motion of an ALE mesh. It is used only when PRTYPE = 5 or 7 in a corresponding *ALE_REFERENCE_SYSTEM_GROUP card.

Card 1	1	2	3	4	5	6	7	8
Variable	ID							
Type	I							
Default	none							

Card 2	1	2	3	4	5	6	7	8
Variable	NID1	NID2	NID3	NID4	NID5	NID6	NID7	NID8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

Card 3	1	2	3	4	5	6	7	8
Variable	NID9	NID10	NID11	NID12				
Type	I	I	I	I				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
ID	Node group ID for PRTYPE 5 or 7, see *ALE_REFERENCE_SYSTEM_GROUP.
NID1, …, NID12	User specified nodes.

Remarks:

1. For PRTYPE = 5 the ALE mesh is forced to follow the motion of a coordinate system, which is defined by three nodes (NID1, NID2, NID3). These nodes are located at x_1, x_2 and x_3, respectively. The axes of the coordinate system, $x^{'}$, $y^{'}$, and $z^{'}$, are defined as:

$$x' = \frac{x_2 - x_1}{|x_2 - x_1|}$$

$$y' = z' \times x'$$

$$z' = x' \times \frac{x_3 - x_1}{|x' \times (x_3 - x_1)|}$$

   Note that $x_1 \rightarrow x_2$ is the local x'axis, $x_1 \rightarrow x_3$ is the local y' axis and x' crosses y' gives the local z' axis. These 3 nodes are used to locate the reference system at any time. Therefore, their positions relative to each other should be as close to an orthogonal system as possible for better transformation accuracy of the ALE mesh.

2. For PRTYPE = 7, the ALE mesh is forced to move and expand, so as to enclose up to twelve user defined nodes (NID1, …, NID12). This is a rarely used option.

Example 1:

Consider modeling sloshing of water inside a rigid tank. Assuming there are 2 ALE parts, the water (PID 1) and air or void (PID 2) contained inside a rigid (Lagrangian) tank (PID 3). The outer boundary nodes of both ALE parts are merged with the inner tank nodes. A part-set ID 1 is defined containing both ALE parts (PIDs 1 and 2). To allow for the meshes of the 2 ALE parts to move with the rigid Lagrangian tank, PRTYPE = 5 is used. The motion of the ALE parts then follows 3 reference nodes on the rigid tank. These 3 reference nodes must be defined by a corresponding *ALE_REFERENCE_SYSTEM_NODE card. In this case the reference nodes have the nodal IDs of 5, 6 and 7. Note that PRID = 12 in the

*ALE_REFERENCE_SYSTEM_GROUP card corresponds to the SID = 12 in the *ALE_-REFERENCE_SYSTEM_NODE card.

```
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|...8
*ALE_REFERENCE_SYSTEM_GROUP
$      SID     STYPE     PRTYP      PRID    BCTRAN     BCEXP     BCROT    ICOORD
         1         0         5        12
$       XC        YC        ZC    EXPLIM
         0         0         0         0
*ALE_REFERENCE_SYSTEM_NODE
$     NSID
        12
$       N1        N2        N3        N4        N5        N6        N7        N8
         5         6         7
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|...8
```

*ALE_REFERENCE_SYSTEM_SWITCH

Purpose: The PRTYPE parameter in the *ALE_REFERENCE_SYSTEM_GROUP (ARSG) card allows many choices of the reference system types for any ALE geometric entity. This command allows for the time-dependent switches between these different types of reference systems, i.e., switching to multiple PRTYPEs at different times during the simulation. This command is required only when PRTYPE = 6 in ARSG card. Please see example 2 in the ARSG section.

Card 1	1	2	3	4	5	6	7	8
Variable	ID							
Type	I							
Default	none							

Card 2	1	2	3	4	5	6	7	8
Variable	T1	T2	T3	T4	T5	T6	T7	
Type	F	F	F	F	F	F	F	
Default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

Card 3	1	2	3	4	5	6	7	8
Variable	TYPE1	TYPE2	TYPE3	TYPE4	TYPE5	TYPE6	TYPE7	TYPE8
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

Card 4	1	2	3	4	5	6	7	8
Variable	ID1	ID2	ID3	ID4	ID5	ID6	ID7	ID8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
ID	Switch list ID, see *ALE_REFERENCE_SYSTEM_GROUP,
T1, …, T7	Times for switching reference system type. By default, the reference system TYPE1 occurs between time = 0 and time = T1, and TYPE2 occurs between time = T1 and time = T2, etc.
TYPE1, …, TYPE8	Reference system types (also see PRTYPE under ARSG): EQ.0: Eulerian, EQ.1: Lagrangian, EQ.2: Normal ALE mesh smoothing, EQ.3: Prescribed motion following load curves, see *ALE_REFERENCE_SYSTEM_CURVE, EQ.4: Automatic mesh motion following mass weighted average velocity in ALE mesh, EQ.5: Automatic mesh motion following a local coordinate system defined by three user defined nodes, see *ALE_-REFERENCE_SYSEM_NODE,
ID1, …, ID8	The corresponding PRID parameters supporting each PRTYPE used during the simulation.

Remarks:

1. The beginning time is assumed to be t = 0, and the starting PRTYPE is TYPE1. So at T1, the 1st switching time, PRTYPE is switched from TYPE1 to TYPE2, and so forth. This option can be complex in nature so it is seldom applied.

*ALE_REFINE

See *CONTROL_REFINE_ALE.

*ALE_SMOOTHING

Purpose: This smoothing constraint keeps an ALE slave node at its initial parametric location along a line between two other ALE nodes. If these nodes are not ALE nodes, the slave node has to follow their motion (see Remark 1). This constraint is active during each mesh smoothing operation. This keyword can be used with ALE solids, ALE shells and ALE beams.

Card 1	1	2	3	4	5	6	7	8
Variable	SNID	MNID1	MNID2	IPRE	XCO	YCO	ZCO	
Type	I	I	I	I	F	F	F	
Default	none	none	none	0	0.0	0.0	0.0	

VARIABLE	DESCRIPTION
SNID	Slave ID, see Figure 4-2. GT.0: SNID is an ALE node, EQ.0: the slaves are the nodes of an ALE mesh connected to the first master nodes (MNID1). See Remarks 2 and 4. LT.0: |SNID| is a ID of ALE node set. See Remark 2.

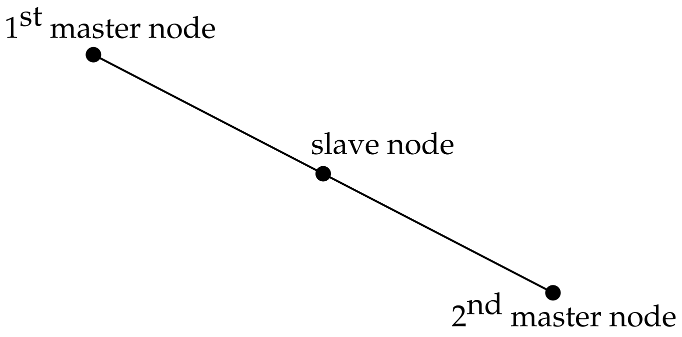


Figure 4-2. This simple constraint, which ensures that a slave node remains on a straight line between two master nodes, is sometimes necessary during ALE smoothing.

VARIABLE	DESCRIPTION
MNID1	First master ID. GT.0: MNID1 is a node, LT.0: |MNID1| is a segment set ID if XCO = YCO = ZCO = 0.0. Otherwise, |MNID1| is a node set ID. See Remarks 2 and 3.
MNID2	Second master ID. GT.0: MNID2 is a node, EQ.0: the slave motion is solely controlled by MNID1. See Remark 5. LT.0: |MNID2| is a node set ID. See Remark 2.
IPRE	EQ.0: smoothing constraints are performed after mesh relaxation, EQ.1: smoothing constraints are performed before mesh relaxation.
XCO	*x*-coordinate of constraint vector
YCO	*y*-coordinate of constraint vector
ZCO	*z*-coordinate of constraint vector

Remarks:

1. **When Master Nodes Are Not ALE Nodes.** If SNID, MNID1 and MNID2 are ALE nodes, the positions of MNID1 and MNID2 are interpolated to position SNID. If MNID1 and MNID2 are not ALE nodes, the motions of MNID1 and MNID2 are interpolated to move SNID.

2. **Node Sets for Constraint Generation.** If MNID1 is a set, SNID and MNID2 should be node sets or zeros. In such a case, the constraints are created during the initialization and printed out in a file called alesmoothingenerated.k for the user's convenience.

3. **Constraint Generation Algorithm.** The constraints for a given master node in MNID1 are generated by finding the closest slave nodes to an axis passing through the master node and oriented by the constraint vector (XCO,YCO,ZCO) if MNID1 is a node set. If MNID1 is a segment set, the nor-

mals of the segments connected to the master node are averaged to give a direction to the axis.

4. **Automatic Identification of Slave Nodes.** If SNID=0, MNID1 should be a set of nodes or segments along the boundary of an ALE mesh. For a given master node in MNID1, the constraints are created for all the nodes of the mesh found the closest to the axis described previously. The search of slaves starts with nodes of elements connected to the master node and stops when a boundary node with an element connectivity similar to the master node's one is reached or when a node in the set MNID2 (if defined) is found.

5. **MNID2 = 0.** If MNID2=0 and SNID is defined, MNID1 should not be ALE. Otherwise SNID and MNID1 positions would match and the element volumes between them could be zero or negative. Only SNID and MNID1 motion should match in such a case.

*ALE_STRUCTURED_MESH

Purpose: This keyword generates a structured 3D mesh and invokes the Structured ALE (S-ALE) solver. Spacing parameters are input through one or more of the *ALE_-STRUCTURED_MESH_CONTROL_POINTS cards. The local coordinate system is defined using the *ALE_STRUCTURED_MESH card.

In certain contexts it is advantageous to use structured meshes. With structured meshes the element and node connectivity are straightforward and the searching algorithm used for ALE coupling is greatly simplified. Also numerous checks are avoided because these meshes include only HEX elements.

This new S-ALE solver supports SMP, MPP and MPP hybrid configurations. All three implementations require less simulation time and memory usage than the regular ALE solver. We, therefore, encourage using the S-ALE solver when the ALE mesh is structured.

The S-ALE solver uses the same set of keyword cards as the regular ALE solver with the only exception being this keyword. Once an ALE mesh is generated using *ALE_-STRUCTURED_MESH card this card invokes the S-ALE and performs the ALE advection timestep. For fluid structured interaction using the *CONSTRAINED_-LARGANGE_IN_SOLID card S-ALE uses a much faster searching algorithm that takes advantage of the mesh structure.

Card 1	1	2	3	4	5	6	7	8
Variable	MSHID	DPID	NBID	EBID				
Type	I	I	I	I				
Default	0	none	0	0				

Card 2	1	2	3	4	5	6	7	8
Variable	CPIDX	CPIDY	CPIDZ	NID0	LCSID			
Type	I	I	I	I	I			
Default	none	none	none	none	none			

VARIABLE	DESCRIPTION
MSHID	S-ALE Mesh ID. A unique number must be specified.
DPID	Default Part ID. The elements generated will assigned to DPID. This part contains no material including only the mesh. This part definition is automatically generated during the input phase and contains neither material nor element formulation information. Please see Remark 1.
NBID	Nodes are generated and assigned with node IDs starting from NBID.
EBID	Elements are generated and assigned with element IDs starting from EBID.
CPIDX, CPIDY, CPIDZ	Control point IDs defining node ID/value pairs along each local axis. See *ALE_STRUCTURED_MESH_CONTROL_POINTS.
NID0	NID0 sets the mesh's origin node. During the simulation, prescribed motion applied to this node applies to the entire structure S-ALE mesh.
LCSID	Local coordinate system ID. Please see Remark 2.

Remarks:

1. **DPID.** The part specific by ID DPID wholey consists of elements and nodes. It does not include material properties or integration rules. The requirmenet that a part ID be specified for these automatically generated S-ALE solid elements exusts only to satisfy the legacy rule that every element must be associated with a part. Users do not need to set up the *PART card for DPID. All PART definitions used in this card only refer to mesh, not material.

2. **LCSID.** The local coordinate system is defined on the data cards associated with the *DEFINE_COORDINATE keyword. This local coordinate cordinate system specifies the three cardinal directions used for generating the structured ALE mesh. The structured mesh can be made to rotate by specifying a rotating local coordinate system. To define a rotating local coordinate system, use the *DEFINE_COORDINATE_NODES keyword with FLAG = 1 and then apply prescribed motion to the three coordinate nodes.

3. **ALE→S-ALE Converter.** For existing ALE models with rectilinear mesh, we could use *ALE_STRUCTURED_MESH card to invoke the ALE→ S-ALE con-

verter. To invoke this feature, add a *ALE_STRUCTURED_MESH card in the model input with CPIDX=-1/0 and all other fields blank. It will then convert all ALE keywords to be of S-ALE format and write the modified input in a file named "saleconvrt.inc". The solver used to perform the analysis depends on the value of CPIDX. If -1, S-ALE solver is used; if 0, ALE solver is used.

Example:

The following example generates a regular evenly distributed 0.2 by 0.2 by 0.2 box mesh having 22 nodes along each direction. The generated mesh is aligned to the local coordinate system spefied by nodes 1, 2, 3, and 4 originating from node 1.

All the elements inside the mesh are assigned to part 1. Note that part 1 is not explicitly defined in the input. The necessary part definition is automatically generated and contains neither material definitions nor integration rules.

```
*ALE_STRUCTURED_MESH
$    mshid      dpid      nbid      ebid
         1         1    200001    200001
$    cpidx     cpidy     cpidz      nid0     lcsid
      1001      1001      1001         1       234
*DEFINE_COORDINATE_NODES
$      cid      nid1      nid2      nid3      flag
       234         2         3         4         1
*ALE_STRUCTURED_MESH_CONTROL_POINTS
      1001
$                   x1                  x2
                     1                  .0
                    22                  .2
*NODE
       1   0.0000000e+00   0.0000000e+00   0.0000000e+00
       2   0.0000000e+00   0.0000000e+00   0.0000000e+00
       3   0.1000000e+00   0.0000000e+00   0.0000000e+00
       4   0.0000000e+00   0.1000000e+00   0.0000000e+00
*END
```

*ALE_STRUCTURED_MESH_CONTROL_POINTS

Purpose: The purpose of this keyword is to provide spacing information used by the *ALE_STRUCTURED_MESH keyword to generate a 3D structured ALE mesh.

Each instance of the *ALE_STRCUTURED_MESH_CONTROL_POINTS card defines a one-dimensional mesh using control. Each control point consists of a node number (see N, card 2) and of a coordinate (see X, card 2). The first control point *must* be node 1, and the node number of the last point defines the total number of nodes. Between and two control points the mesh is uniform. The *ALE_STRUCTURED_MESH card, in turn, defines a simple three dimensional mesh from the triple product of three *ALE_STRUCTURED_MESH_CONTROL_POINT one-dimensional meshes.

Card 1	1	2	3	4	5	6	7	8
Variable	CPID	Not used	Not used	SFO	Not used	OFFO		
Type	I			F		F		
Default	None			1.		0.		

Point Cards. Put one pair of points per card (2E20.0). Input is terminated at the next keyword ("*") card. *At least two cards are required, one of which, having* N = 1 *is required*.

Card 2	1	2	3	4	5	6	7	8
Variable	N		X		RATIO			
Type	I20		E20.0		E20.0			
Default	none		none		0.0			

VARIABLE	DESCRIPTION
CPID	Control Points ID. A unique number must be specified. This ID is to be referred in the three fields marked up CPIDX, CPIDY, CPIDZ in *ALE_STRUCTURED_MESH.
SFO	Scale factor for ordinate value. This is useful for simple modifications. EQ.0.0: default set to 1.0.

VARIABLE	DESCRIPTION
OFFO	Offset for ordinate values. See Remark 1.
N	Control point node number.
X	Control point position.
RATIO	Ratio for progressive mesh spacing. Progressively larger or smaller mesh will be generated between the control point that has nonzero ratio specified and the control point following it. See remark 2. GT.0.0: mesh size increases; $dl_{n+1} = dl_n * (1 + ratio)$ LT.0.0: mesh size decreases; $dl_{n+1} = dl_n/(1 - ratio)$

Remarks:

1. **Coordinates scaling.** The ordinate values are scaled after the offsets are applied, i.e.:

   $$\text{Ordinate value} = \text{SFO} \times (\text{Defined value} + \text{OFFO})$$

2. **Progressive mesh spacing.** The formula used to calculate element length is as follows: $dl_{base} = |x_{end} - x_{start}| * (factor - 1)/(factor^n - 1)$ in which dl_{base} is the smallest base length; x_{start} and x_{end} are the coordinate of the start and end point respectively; $factor = 1 + ratio\ (ratio > 0)\ or\ 1/(1 - ratio)\ (ratio < 0)$. Please note here element size either increases by $ratio$ $(ratio > 0)$ or decreases by $-ratio/(1 - ratio)$ $(ratio < 0)$ each time. But the overall effect is the same: starting from the smallest element, each time the element size is increased by $|ratio|$.

Example:

1. This example below generates a regular box mesh. Its size is 0.2 by 0.2 by 0.2. It is generated in a local coordinate system defined by three nodes 2, 3, 4 and originates from node 1.

The local x-axis mesh spacing is defined by control points ID 1001. It has 21 nodes evenly distributed from 0.0 to 0.2. The local y-axis is defined by ID 1002 and has twice the elements of 1001. It has 41 nodes evenly distributed from 0.0 to 0.2. The local z-axis is defined by ID 1003. It has 31 nodes and covers from 0.0 to 0.2. The mesh is two times finer in the region between node 6 and node 26.

```
*ALE_STRUCTURED_MESH
$    mshid      dpid      nbid      ebid
         1         1    200001    200001
$    cpidx     cpidy     cpidz      nid0     lcsid
      1001      1002      1003         1       234
*DEFINE_COORDINATE_NODES
$      cid      nid1      nid2      nid3
       234         2         3         4
*ALE_STRUCTURED_MESH_CONTROL_POINTS
      1001
$                 x1                  x2
                   1                  .0
                  21                  .2
*ALE_STRUCTURED_MESH_CONTROL_POINTS
      1002
$                 x1                  x2
                   1                  .0
                  41                  .2
*ALE_STRUCTURED_MESH_CONTROL_POINTS
      1003
$                 x1                  x2
                   1                  .0
                   6                 .05
                  26                 .15
                  31                  .2
*NODE
       1   0.0000000e+00   0.0000000e+00   0.0000000e+00
       2   0.0000000e+00   0.0000000e+00   0.0000000e+00
       3   0.1000000e+00   0.0000000e+00   0.0000000e+00
       4   0.0000000e+00   0.1000000e+00   0.0000000e+00
*END
```

2. This example shows how to generate a progressive larger/smaller mesh spacing. The mesh geometry is the same as the example above. At x-direction the mesh is progressively smaller between node 1 and 8. For these 7 elements, each element is 0.1/1.1 = 9.09% smaller than its left neighbor. Between node 15 and node 22, the mesh is progressively larger by 10% each time in those 7 elements. The 7 elements in the middle between node 8 and 15 are of equal length.

```
*ALE_STRUCTURED_MESH
$    mshid      dpid      nbid      ebid
         1         1    200001    200001
$    cpidx     cpidy     cpidz      nid0     lcsid
      1001      1002      1003         1       234
*DEFINE_COORDINATE_NODES
$      cid      nid1      nid2      nid3
       234         2         3         4
*ALE_STRUCTURED_MESH_CONTROL_POINTS
      1001
$                 x1                  x2               ratio
                   1                  .0                -0.1
                   8          0.06666667
                  15          0.13333333                 0.1
                  22                  .2
```

*ALE_STRUCTURED_MESH_REFINE

Purpose: This keyword is to provide a convenient utility to refine existing meshes generated by *ALE_STRUCTURED_MESH card. All the NODESET, ELEMENTSET and SEGMENTSET defined using SALECPT and SALEFAC options in *SET_cards will be automatically updated. This way, this card is the only modification in the input deck for users to define a refined S-ALE mesh.

Card 1	1	2	3	4	5	6	7	8
Variable	MSHID	IFX	IFY	IFZ				
Type	I	I	I	I				
Default	none	1	1	1				

VARIABLE	DESCRIPTION
MSHID	S-ALE Mesh ID. The ID of the Structured ALE mesh to be refined.
IFX, IFY, IFZ	Refinement factor at each local direction. Please see remark 1.

Remarks:

1. IFX, IFY, IFZ prescribe how many times to refine the grid along each direction. They have to be integers.

2. This keyword provides a new modeling technique to handle the multi-material ALE problems. Compared to pure Lagrange problems, models contain multi-material ALE fluids are often time consuming and memory demanding. So it is better to construct a concept model with much coarse mesh to get an estimate of the computational resources needed and refine the concept model mesh gradually until convergence is achieved. This keyword minimized the user effort following such procedure.

Example:

This example below generates two regular evenly distributed box mesh. Each has 22 nodes along each direction and the overall size is 0.2 by 0.2 by 0.2. S-ALE mesh 1 is generated in a local coordinate system defined by three nodes 2,3,4 and originated from node 1.

If at later times, we decided to make the mesh finer, we can simply add the following card. Now the solid element set 100 would contain elements ranging between nodes (1,1,23) and (45,45,45) instead of the original (1,1,11) and (22,22,22).

```
*ALE_STRUCTURED_MESH_REFINE
$    mshid       ifx       ify       ifz
         1         2         2         2
*ALE_STRUCTURED_MESH
$    mshid      dpid      nbid      ebid
         1         1    200001    200001
$    cpidx     cpidy     cpidz      nid0     lcsid
      1001      1001      1001         1       234
*DEFINE_COORDINATE_NODES
$      cid      nid1      nid2      nid3
       234         2         3         4
*SET_SOLID_GENERAL
$      SID
       100
$    OPTION      MSHID         XMN         XMX         YMN         YMX         ZMN
ZMX
   SALECPT           1           1          22           1          22          11       22
*ALE_STRUCTURED_MESH_CONTROL_POINTS
      1001
$                 x1                  x2
                   1                  .0
                  22                  .2
*NODE
       1   0.0000000e+00   0.0000000e+00   0.0000000e+00
       2   0.0000000e+00   0.0000000e+00   0.0000000e+00
       3   0.1000000e+00   0.0000000e+00   0.0000000e+00
       4   0.0000000e+00   0.1000000e+00   0.0000000e+00
       5   0.0000000e+00   0.0000000e+00   0.0000000e+00
*END
```

*ALE_SWITCH_MMG

Purpose: This card changes a fraction of an ALE multi-material-group (AMMGID) into another group. The fraction is to be specified by a *DEFINE_FUNCTION function. The function take as many arguments as there are fields specified on the cards in format 2.

Card 1	1	2	3	4	5	6	7	8
Variable	FR_MMG	TO_MMG	IDFUNC	IDSEGSET	IDSLDSET	NCYCSEG	NCYCSLD	
Type	I	I	I	I	I	I	I	
Default	none	none	None	0	0	50	50	

Variable Cards. Cards defining the function arguments. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	VAR	VAR	VAR	VAR	VAR	VAR	VAR	VAR
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

VARIABLE	DESCRIPTION
FR_MMG	This is the AMMG-SID before the switch. The AMMG-SID corresponds to the SID defined on a *SET_MULTI-MATERIAL_-GROUP_LIST (SMMGL) card. This SID refers to one or more AMMGs. See Remark 1.
TO_MMG	This is the AMMG-SID after the switch. The AMMG-SID corresponds to the SID defined on a *SET_MULTI-MATERIAL_-GROUP_LIST card. This SID refers to one or more AMMGs. See Remark 1.
IDFUNC	ID of a *DEFINE_FUNCTION function. This function determines the material fraction to be switched. See Example 1.

VARIABLE **DESCRIPTION**

IDSEGSET

ID of *SEGMENT_SET that is used to pass geometric properties to the function specified by IDFUNC. *This field is optional*.

The segment center positions and normal vectors are computed. For each ALE element, this data is passed to the function IDFUNC for the segment the closest to the element center. See Example 2.

IDSLDSET

The ID of a *SOLID_SET specifying which elements are affected by this particular instance of the *ALE_SWITCH_MMG keyword. *This field is optional*. If undefined, *ALE_SWITCH_MMG affects all ALE elements. The element centers are computed and can be used as variables in the function IDFUNC.

NCYCSEG

Number of cycles between each update of the segment centers and normal vectors (if a segment set is defined). For each update, a bucket sort is applied to find the closest segment to each ALE element. If the segment nodes are fully constrained, the segment centers and normal vectors are computed only one time.

NCYCSLD

Number of cycles between each update of the ALE element centers. For each update, a bucket sort is applied to find the closest segment to each ALE element. If the element nodes does not move (as with AFAC = -1 in *CONTROL_ALE) the element centers are computed exactly once.

VAR

Variable rank in the following list (See Remark 2):

EQ.0:	See Remark 3
EQ.1:	*xx*-stress for FR_MMG
EQ.2:	*yy*-stress for FR_MMG
EQ.3:	*zz*-stress for FR_MMG
EQ.4:	*xy*-stress for FR_MMG
EQ.5:	*yz*-stress for FR_MMG
EQ.6:	*zx*-stress for FR_MMG
EQ.7:	plastic strain for FR_MMG
EQ.8:	internal energy for FR_MMG
EQ.9:	bulk viscosity for FR_MMG
EQ.10:	volume from previous cycle for FR_MMG
GE.11 and LE.20:	other auxiliary variables for FR_MMG

VARIABLE **DESCRIPTION**

GE.21 and LE.40:	auxiliary variables for TO_MMG (xx-stress, …)
EQ.41:	mass for FR_MMG
EQ.42:	mass for TO_MMG
EQ.43:	volume fraction for FR_MMG
EQ.44:	volume fraction for TO_MMG
EQ.45:	material volume for FR_MMG
EQ.46:	material volume for TO_MMG
EQ.47:	time
EQ.48:	cycle
EQ.49:	x-position of the ALE element center
EQ.50:	y-position of the ALE element center
EQ.51:	z-position of the ALE element center
EQ.52:	x-position of the segment center
EQ.53:	y-position of the segment center
EQ.54:	z-position of the segment center
EQ.55:	x-component of the segment normal
EQ.56:	y-component of the segment normal
EQ.57:	z-component of the segment normal
GE.58 and LE.65:	x-positions of the ALE nodes
GE.66 and LE.69:	x-positions of the segment nodes
GE.70 and LE.77:	y-positions of the ALE nodes
GE.79 and LE.81:	y-positions of the segment nodes
GE.83 and LE.89:	z-positions of the ALE nodes
GE.90 and LE.93:	z-positions of the segment nodes
GE.94 and LE.101:	x-velocities of the ALE nodes
GE.102 and LE.105:	x-velocities of the segment nodes
GE.106 and LE.113:	y-velocities of the ALE nodes
GE.114 and LE.117:	y-velocities of the segment nodes
GE.118 and LE.125:	z-velocities of the ALE nodes
GE.126 and LE.129:	z-velocities of the segment nodes

VARIABLE	DESCRIPTION	
	GE.130 and LE.137:	x-accelerations of the ALE nodes
	GE.138 and LE.141:	x-accelerations of the segment nodes
	GE.142 and LE.149:	y-accelerations of the ALE nodes
	GE.150 and LE.153:	y-accelerations of the segment nodes
	GE.154 and LE.161:	z-accelerations of the ALE nodes
	GE.162 and LE.165:	z-accelerations of the segment nodes
	GE.166 and LE.173:	masses of the ALE nodes
	GE.174 and LE.177:	masses of the segment nodes
	EQ.178:	rank of the variable updated by the function (See Remark 4)
	EQ.179:	rank of the multi-material group in the set
	EQ.180:	time step

Remarks:

1. **Mapping.** The multi-material group sets that are specified by the fields FR_MMG and TO_MMG must be of the same length. Multi-material groups are switched so that, for instance, the fourth multi-material group in the set FR_MMG is mapped to the fourth multi-material group in the set TO_MMG. (see Example 1).

2. **Variable Specification.** The variables are presented to the function IDFUNC as floating point data. The order of the arguments appearing in the *DEFINE_-FUNCTION should match the order of variable ranks VAR specified on Card 2 (for this keyword). For example, when there is one card in format 2 containing "47, 48, 41, 42", then the time (47), the cycle (48), and the masses (41 & 42) should be the first, second, third, and fourth arguments to the function defined on the *DEFINE_FUNCTION keyword.

   If there is a blank column between 2 variable ranks, the list between these 2 ranks is specified. For example, if the card contains "1, ,6", then the 6 stresses (1 through 6) are selected as arguments (see Example 2). In the case that there are several groups in the sets, if a variable rank VAR is repeated, the corresponding variable will be defined in the function for each group. For instance, if the sets have 3 groups and the volume fractions of the 2 first groups in the set TO_MMG are required as arguments of the function, a card in format 2 should have "44,44".

3. **Variable Update for Several Groups.** If there is more than one group in the set, the function is called for each group. For a given group with a rank in the set > 1 (VAR=179), some variables including the volume fraction, mass, internal energy may have been updated during the previous switches. If their original values are required, they can be obtained by setting the first field (VAR) to 0.

4. **Variable Update by the User.** The variables can be updated by the user. If VAR < 0 for some variables, the function is called again (after the switch) for each of these variables. VAR = 178 gives the rank of the variable for which the function is called. The function's return value is taken as the new value for this variable (instead of the fraction of material to switch). If the rank given by VAR = 178 is zero, it means that the function is called for the switch. Only the 46 first variables 1 < VAR < 46, 58 < VAR < 165 and VAR = 180 can be modified.

Example 1:

The first example switches the material if the pressure is lower than a given value.

```
*comment
units: mks

Switch from the 3rd group to the 5th one if the pressure of the 3rd
      group
is lower than pc : pres < pc
Do the same for the switch from 4th to 7th
If the switch occurs, the function frac returns 1.0.  So the whole
material is permuted.

xxsig   :   xx-stress   of   the   groups   in   the   1st   *set_multi-
      material_group_list
yysig   :   yy-stress   of   the   groups   in   the   1st   *set_multi-
      material_group_list
zzsig   :   zz-stress   of   the   groups   in   the   1st   *set_multi-
      material_group_list
pres  : pressure
pc    : pressure cutoff

*ALE_SWITCH_MMG
$#  fr_mmg    to_mmg    idfunc  idsegset  idsldset   ncycseg   ncycsld
         1         2        10
         1         2         3
*set_multi-material_group_list
1
3,4
*set_multi-material_group_list
2
5,7
*DEFINE_FUNCTION
10
float frac(float xxsig, float yysig, float zzsig)
{
  float pc;
  pres = -(xxsig+yysig+zzsig)/3.0;
```

```
 pc = -1000;
 if (pres < pc) {
   return 1.0;
 } else {
   return 0.0;
 }
}
```

Example 2:

The second example switches the material if it goes through a segment.

```
*comment
units: mks

Switch the 1st group to the 2nd group if the ALE element center goes
through a segment of the set defined by idsegset = 1.
The segment position is updated every cycle
A fraction of the material is switched.  This fraction depends on the
distance between the segment and element centers

time    : 47th variable
cycle   : 48th variable
xsld    : 49th variable (x-position of the element center)
ysld    : 50th variable (y-position of the element center)
zsld    : 51th variable (z-position of the element center)
xseg    : 52th variable (x-position of the segment center)
yseg    : 53th variable (y-position of the segment center)
zseg    : 54th variable (z-position of the segment center)
xn      : 55th variable (x-component of the segment normal)
yn      : 56th variable (y-component of the segment normal)
zn      : 57th variable (z-component of the segment normal)
volmat1 : 43th variable (material volume of the 1st group)
volfrac1: 45th variable (volume fraction of the 1st group)
segsurf : segment surface (given by 0.5*sqrt(xn*xn+yn*yn+zn*zn))
sldvol  : ALE element volume (given by volmat1/volfrac1)
segcharaclen: characteristic length for the segment
sldcharaclen: characteristic length for the solid
xseg2sld: x-component of the vector segment center to element center
yseg2sld: y-component of the vector segment center to element center
zseg2sld: z-component of the vector segment center to element center
distnormseg2sld: Distance segment-element projected on the normal
disttangseg2sld: Distance segment-element projected on the segment plane

*ALE_SWITCH_MMG
$#  fr_mmg    to_mmg    idfunc  idsegset  idsldset   ncycseg   ncycsld
         1         2        11         1                   1
        47                  57        43        45
*set_multi-material_group_list
1
1
*set_multi-material_group_list
2
2
*DEFINE_FUNCTION
11
float switchmmg(float time, float cycle,
                float xsld, float ysld, float zsld,
                float xseg, float yseg, float zseg,
                float xn, float yn,  float zn,
                float volmat1, float volfrac1)
{
  float segsurf, sldvol, segcharaclen, sldcharaclen;
  float xseg2sld, yseg2sld, zseg2sld, distnormseg2sld;
  float xtangseg2sld, ytangseg2sld, ztangseg2sld, disttangseg2sld;
  float frac;
  segsurf = sqrt(xn*xn+yn*yn+zn*zn);
  if (segsurf != 0.0) {
    xn = xn/segsurf;
    yn = yn/segsurf;
    zn = zn/segsurf;
  }
  segsurf = 0.5*segsurf;
  sldvol = volmat1/volfrac1;
  segcharaclen = 0.5*sqrt(segsurf);
  sldcharaclen = 0.5*sldvol**(1.0/3.0);
  xseg2sld = xsld-xseg;
  yseg2sld = ysld-yseg;
  zseg2sld = zsld-zseg;
  distnormseg2sld = xseg2sld*xn+yseg2sld*yn+zseg2sld*zn;
  xtangseg2sld = xseg2sld-distnormseg2sld*xn;
  ytangseg2sld = yseg2sld-distnormseg2sld*yn;
  ztangseg2sld = zseg2sld-distnormseg2sld*zn;
  disttangseg2sld = xtangseg2sld*xtangseg2sld+
                  ytangseg2sld*ytangseg2sld+
                  ztangseg2sld*ztangseg2sld;
  disttangseg2sld = sqrt(disttangseg2sld);
  if (disttangseg2sld <= segcharaclen &&
      distnormseg2sld <= sldcharaclen) {
    sldcharaclen = 2.0*sldcharaclen;
    frac = distnormseg2sld/sldcharaclen;
    frac = 0.5-frac;
    return frac;
  } else {
    return 0.0;
  }
}
```

*ALE_TANK_TEST

Purpose: Control volume airbags (*AIRBAG_) only require two engineering curves to define gas inflator, i.e. $\dot{m}(t)$ and $\overline{T}_{\mathrm{gas}}(t)$; those two curves can be experimentally measured. However, the ALE inflator needs one additional state variable - the inlet gas velocity which is impractical to obtain. This keyword is to provide such curve through an engineering approximation.

It takes two curves from the accompanying *SECTION_POINT_SOURCE as input. It assumes inflator gas under choking condition to generate velocity curve. During this process, the original curves, $\dot{m}(t)$ and $\overline{T}_{\mathrm{gas}}(t)$, are modified accordingly.

It complements and must be used together with the*SECTION_POINT_SOURCE command. Please see *SECTION_POINT_SOURCE for additional information.

Card 1	1	2	3	4	5	6	7	8
Variable	MDOTLC	TANKV	PAMB	PFINAL	MACHL	VELMAX	AORIF	
Type	I	I	I	I	F	F	F	
Default	0	0.0	0.0	0.0	0.0	0.0	0.0	

Card 2	1	2	3	4	5	6	7	8
Variable	AMGIDG	AMGIDA	NUMPNT					
Type	I	I	I					
Default	0	0	50					

VARIABLE	DESCRIPTION
MDOTLC	LCID for mass flow rate as a function of time. This may be obtained directly from the control-volume type input data.
TANKV	Volume of the tank used in a tank test from which the tank pressure is measured, and $\dot{m}(t)$ and $\overline{T}_{\mathrm{gas}}(t)$ are computed from this tank pressure data.

VARIABLE	DESCRIPTION
PAMB	The pressure inside the tank before jetting (usually 1bar).
PFINAL	The final equilibrated pressure inside the tank from the tank test.
MACHL	A limiting MACH number for the gas at the throat (MACH = 1 preferred).
VELMAX	Maximum allowable gas velocity across the inflator orifice (not preferred).
AORIF	Total inflator orifice area (optional, only needed if the *SECTION_POINT_SOURCE card is not used).
AMGIDG	The ALE multi-material group ID (AMMGID) of the gas.
AMGIDA	The ALE multi-material group ID (AMMGID) of the air.
NUMPNT	The number of points in $\dot{m}(t)$ and $\overline{T}_{\text{gas}}(t)$ curves. If NUMPNT = 0, defaults to 50 points.

Remarks:

In an airbag inflator tank test, the tank pressure data is measured. This pressure is used to derive $\dot{m}(t)$ and to estimate $\overline{T}_{\text{gas}}(t)$, the stagnation temperature of the inflator gas. This is done by applying a lumped-parameter method to the system of conservation equations using an equation of state.

Together $\dot{m}(t)$ and $\overline{T}_{\text{gas}}(t)$ provide enough information to model an airbag with the control volume method (see *AIRBAG cards). However, for an ALE or Eulerian fluid-structure interaction analysis, the gas velocity, $v(t)$, and density, $\rho(t)$, at the inlet must be computed. But, since only $\dot{m}(t)$ is known, additional assumptions must be made about the inlet conditions. If $v(t)$ and $\rho(t)$ are calculated outside of LS-DYNA, then LS-DYNA combines them with $\dot{m}(t)$ and $\overline{T}_{\text{gas}}(t)$ to obtain $\overline{T}_{\text{gas corrected}}(t)$, $v(t)$ and $\rho(t)$ which are sufficient input for an ALE calculation.

The curves $v(t)$ and $\rho(t)$ need not be calculated outside of LS-DYNA as LS-DYNA features a method for calculating them itself. This card, *ALE_TANK_TEST, activates this capability. Thus, with the combination of this card and the *SECTION_POINT_-SOURCE card, LS-DYNA can proceed directly from the control volume method input, $\dot{m}(t)$ and $\overline{T}_{\text{gas}}(t)$, to an ALE or Eulerian fluid-structure interaction analysis. The user does not have to do the conversion himself.

If the *ALE_TANK_TEST card is present:

1. The definitions of the relative volume, $V(t)$, and the velocity, $v(t)$, curves in the *SECTION_POINT_SOURCE card will be ignored in favor of those computed by LS-DYNA.

2. The $\dot{m}(t)$curve is read in on *ALE_TANK_TEST card.

3. The $\overline{T}_{\text{gas}}(t)$ curve (stagnation temperature), as opposed to $\overline{T}_{\text{gas corrected}}(t)$, is read in on *SECTION_POINT_SOURCE card.

   There is a subtle, but important, distinction between the two temperatures. $\overline{T}_{\text{gas}}(t)$ is derived directly from the tank pressure data based on a lump-parameter approach, whereas $\overline{T}_{\text{gas corrected}}(t)$ is computed from $\dot{m}(t)$ and $\overline{T}_{\text{gas}}(t)$ with additional isentropic and sonic flow assumptions for the maximum velocity at an orifice. $\overline{T}_{\text{gas corrected}}(t)$ is most appropriately interpreted as the static temperature. These assumptions provide a necessary and physically reasonable supplement to the governing equation,

   $$\dot{m}(t) = \rho(t)v(t)A$$

   in which only$\dot{m}(t)$ and A are known leaving two parameters: $\rho(t)$, and $v(t)$ as unkown.

4. The inflator area is computed from the *SECTION_POINT_SOURCE card that has the AMMGID of the inflator gas in the *ALE_TANK_TEST card. If the *BOUNDARY_AMBIENT_EOS card is used instead of the *SECTION_POINT_SOURCE card, then the area may be input in this *ALE_TANK_TEST card.

5. The reference density of the propellant "gas", ρ_0, is computed internally and automatically used for the calculation. The ρ_0 value from the *MAT_NULL card is ignored.

Example:

Consider a tank test model consists of the inflator gas (PID 1) and the air inside the tank (PID 2). The following information from the control volume model is available:

- $\dot{m}(t)$ (LCID 1 is from control volume model input).
- $\overline{T}_{\text{gas}}(t)$ (LCID 2 is from control volume model input).
- Volume of the tank used in the inflator tank test.
- Final equilibrated pressure inside the tank.
- Ambient pressure in the air.

Also available are:

- The nodal IDs of the nodes defining the orifice holes through which the gas flows into the tank.
- The area associated with each hole (the node is assumed to be at the center of this area).
- The vector associated with each hole defining the direction of flow.

In the input below LCID 1 and 2 are $\dot{m}(t)$ and $\bar{T}_{\text{gas}}(t)$, respectively. LCID 4 and 5 will be ignored when the *ALE_TANK_TEST card is present. If it is not present, all 3 curves in the *SECTION_POINT_SOURCE card will be used. When the *SECTION_POINT_SOURCE card is present, the element formulation is equivalent to an ELFORM = 11.

```
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|...8
*PART
inflator gas
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
         1         1         1         0         0         0         0         0
*PART
air inside the tank
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
         2         2         2         0         0         0         0         0
*SECTION_SOLID
$    SECID    ELFORM       AET
         2        11         0
*ALE_MULTI-MATERIAL_GROUP
$      SID   SIDTYPE
         1         1
         2         1
*SECTION_POINT_SOURCE
$    SECID     LCIDT  LCIDVOLR   LCIDVEL      <= 3 curves in tempvolrvel.k file
         1         2         4         5
$   NODEID    VECTID      AREA
     24485         3    15.066
       ...
     24557         3    15.066
*ALE_TANK_TEST
$   MDOTLC     TANKV      PAMB    PFINAL     MACHL    VELMAX     AORIF
         1     6.0E7    1.0E-4  5.288E-4       1.0       0.0
$   AMGIDG    AMGIDA    NUMPNT
         1         2        80
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|...8
```

*ALE_UP_SWITCH

Purpose: For the simulation of airbag inflation process, this card allows the switching from an ALE computation to a control volume (CV) or uniform pressure (UP) method at a user-defined switch time.

Card 1	1	2	3	4	5	6	7	8
Variable	UPID	SWTIME						
Type	I	F						
Default	0	1.0e+16						
Remark	1							

Card 2	1	2	3	4	5	6	7	8
Variable	FSI_ID1	FSI_ID2	FSI_ID3	FSI_ID4	FSI_ID5	FSI_ID6	FSI_ID7	FSI_ID8
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

Additional card for UPID = 0 (or not defined).

Optional 3	1	2	3	4	5	6	7	8
Variable	SID	SIDTYPE	MMGAIR	MMGGAS				
Type	I	I	I	I				
Default	0	0	0	0				

VARIABLE DESCRIPTION

UPID

An ID defines a corresponding *AIRBAG_HYBRID_ID card for use in an ALE-method-switching-to-CV-method simulation. The simulation starts with ALE computational method, then switches to a CV (or UP) method at some given time.

EQ.0: (or blank) The code will construct an equivalent *AIRBAG_HYBRID_ID card automatically internally, (default). The 3rd optional line is then a required input.

NE.0: An ID points to a corresponding *AIRBAG_HYBRID_ID card which must be defined for use after the switch. If UPID is defined, do not define the 3rd optional card.

SWTIME

The time at which the computation does a switch from an ALE-method-to-CV-method.

FSI_ID1, …, FSI_ID8

Coupling IDs for one or more ALE fluid-structure-interaction (FSI) *CONSTRAINED_LAGRANGE_IN_SOLID_ID cards. These couplings are deleted during the 2nd, CV computational phase.

SID

A set ID defines the Lagrangian parts which make up the airbag.

SIDTYPE

Set ID type for the above SETID (following the conventions in *AIRBAG_HYBRID card).

EQ.0: SID is a segment set ID (SGSID).

NE.0: SID is a part set ID (PSID).

MMGAIR

The AMMG (ALE multi-material group) ID of surrounding air.

MMGGAS

The AMMG ID of inflator gas injected into the airbag.

Remarks:

1. If UPID is zero or blank, optional card 3 must be defined. LSDYNA will construct an equivalent *AIRBAG_HYBRID_ID card automatically.

Example 1:

Consider an airbag model with a 2-phase simulation: an ALE calculation being switched to a CV method. During the CV phase, the simulation is defined by an *AIRBAG_HYBRID_ID card.

```
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
*ALE_UP_SWITCH
$    UP_ID   SW_time
   100000    2.0000
$ FSI_ID_1  FSI_ID_2  FSI_ID_3  FSI_ID_4  FSI_ID_5  FSI_ID_6  FSI_ID_7  FSI_ID_8
         1         2
$-------------------------------------------------------------------------------
*AIRBAG_HYBRID_ID
$       ID
   100000
$      SID    SIDTYP      RBID      VSCA      PSCA      VINI       MWD      SPSF
         2         1         0       1.0       1.0       0.0       0.0       0.0
$ 2   ATMT      ATMP      ATMD        GC        CC
      293.  1.0130e-4  1.200E-9    8.3143        1.
$      C23     LCC23       A23     LCA23      CP23     LCP23      AP23    LCAP23

$      OPT     PVENT      NGAS
                             4
$bac LCIDM     LCIDT   NOTUSED        MW     INITM         A         B         C
      1001      1002           0.0288691       1.0     28.98
$    FMASS

$air LCIDM     LCIDT   NOTUSED        MW     INITM         A         B         C
      1600      1603            28.97E-3       0.0     26.38  8.178e-3 -1.612e-6
$    FMASS

$pyroLCIDM     LCIDT   NOTUSED        MW     INITM         A         B         C
      1601      1603            43.45E-3       0.0     32.87  2.127e-2 -5.193E-6
$    FMASS

$sto_LCIDM     LCIDT   NOTUSED        MW     INITM         A         B         C
      1602      1603            39.49E-3       0.0     22.41  2.865e-3 -6.995e-7
$    FMASS
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
```

Example 2:

Consider the same airbag model with the same 2-phase simulation. However, all the *AIRBAG_HYBRID_ID card definitions are extracted automatically from the ALE model. There is no need to define the *AIRBAG_HYBRID_ID card. The 3rd optional card is required.

```
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
*ALE_UP_SWITCH
$    UP_ID   SW_time
$   100000    2.0000
         0    2.0000
$ FSI_ID_1  FSI_ID_2  FSI_ID_3  FSI_ID_4  FSI_ID_5  FSI_ID_6  FSI_ID_7  FSI_ID_8
         1         2
$    SETID    SETYPE   MMG_AIR   MMG_GAS
         2         1         2         1
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
```

*BOUNDARY

The keyword *BOUNDARY provides a way of defining imposed motions on boundary nodes. The keyword control cards in this section are defined in alphabetical order:

*BOUNDARY_ACOUSTIC_COUPLING

*BOUNDARY_ACOUSTIC_IMPEDANCE

*BOUNDARY_ACOUSTIC_MAPPING

*BOUNDARY_ALE_MAPPING

*BOUNDARY_AMBIENT

*BOUNDARY_AMBIENT_EOS

BOUNDARY_CONVECTION_*OPTION

*BOUNDARY_COUPLED

*BOUNDARY_CYCLIC

*BOUNDARY_DE_NON_REFELECTING

BOUNDARY_ELEMENT_METHOD_*OPTION

BOUNDARY_FLUX_*OPTION

*BOUNDARY_MCOL

*BOUNDARY_NON_REFLECTING

*BOUNDARY_NON_REFLECTING_2D

*BOUNDARY_PAP

BOUNDARY_PORE_FLUID_*OPTION

*BOUNDARY_PRECRACK

*BOUNDARY_PRESCRIBED_ACCELEROMETER_RIGID

*BOUNDARY_PRESCRIBED_FINAL_GEOMETRY

BOUNDARY_PRESCRIBED_MOTION_*{OPTION1}_{OPTION2}

BOUNDARY_PRESCRIBED_ORIENTATION_RIGID_*OPTION

BOUNDARY_PRESSURE_OUTFLOW_*OPTION

BOUNDARY_PWP_*OPTION

BOUNDARY_RADIATION_*OPTION

*BOUNDARY_SLIDING_PLANE

BOUNDARY_SPC_*{OPTION1}_{OPTION2}_{OPTION3}

*BOUNDARY_SPC_*SYMMETRY*_PLANE_*OPTION*

*BOUNDARY_SPH_FLOW

*BOUNDARY_SPH_NON_REFLECTING

*BOUNDARY_SPH_SYMMETRY_PLANE

*BOUNDARY_SYMMETRY_FAILURE

BOUNDARY_TEMPERATURE_*OPTION

BOUNDARY_THERMAL_BULKFLOW_*{OPTION1}_{OPTION2}

*BOUNDARY_THERMAL_BULKNODE

*BOUNDARY_THERMAL_WELD

*BOUNDARY_THERMAL_WELD_TRAJECTORY

*BOUNDARY_USA_SURFACE

BOUNDARY_ACOUSTIC_COUPLING_*{OPTION}

There are two forms of this keyword command:

1. for coupling of surfaces with coincident nodes

   *BOUNDARY_ACOUSTIC_COUPLING

2. for coupling surfaces without coincident nodes

   *BOUNDARY_ACOUSTIC_COUPLING_MISMATCH

Purpose: Define a segment set for acoustic coupling of structural element faces and acoustic volume elements (type 8 and type 14 solid elements.)

If the mismatch option is not used, then this command couples either one side of a shell or solid element structure or both sides of a shell structure to acoustic elements. The segments in the segment set should define the structural surface for which coupling is intended. The nodal points of the structural segments must be coincident with the nodal points for the fluid element faces on either side of the structural segments. If fluid exists on just one side of the structural segments, and the nodes are merged, then the input data in this section is not required. The coupling will happen automatically. However, if fluid is on both sides of the structural segments, then this input data is required and the nodes should not be merged; two-sided coupling will not properly apply loads when the interface nodes are merged out.

If the mismatch option is used, then this command permits the coupling of acoustic fluid volume elements with one side of a structural element when the meshes of the fluid and structural models are moderately mismatched. In this case, it is possible that most fluid and structural nodes will not be coincident. None of the fluid and structural nodes at the interface should be merged together. The segments in the segment set should define the structural surface and, following a right hand rule, the normal vector for the segments should point at the fluid volume elements with which coupling is intended. If coupling is required on both sides of a structural shell element, duplicate segments with opposite normal vectors should be defined. Every segment in the segment set must couple with the fluid volume at some integration point, but it is not necessary that all integration points on the segment couple with the fluid. The meshes do not have to be mismatched to use mismatched coupling, as long as the fluid and structural nodes are not merged.

Card 1	1	2	3	4	5	6	7	8
Variable	SSID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
SSID	Segment set ID, see *SET_SEGMENT

Remarks:

1. For the stability of the acoustic-structure coupling, the following condition must be satisfied:

$$\frac{2\rho_a D}{\rho_s t_s} < 5$$

where ρ_a is the density of the acoustic medium, D is the total thickness of the acoustic elements adjacent to the structural element, ρ_s is the density, and t_s is the thickness of the structural shell element. If the structural element is a solid or thick shell element, then t_s should be half the thickness of the element. If coupling is on both sides of the structural elements, then t_s should also be half the thickness of the structural element.

2. In mismatched coupling, free fluid faces are considered for coupling with the structural segments if they are near one another and if they face each other. Faces and segments that differ in orientation by more than 45 degrees are excluded. In regions of high curvature the surfaces therefore need to be more similar than when the surfaces are flat. If a fluid face couples with any structural segment, then all four integration points on the fluid face must couple with some structural segment. Fluid faces may not be partially coupled. Structural segments are allowed to be partially coupled.

3. The mismatched coupling process dumps two LS-DYNA files that can be imported into LS-PrePost for review of the results of the coupling process. File “bac_str_coupling.dyn” contains shell elements where structural segments have coupled with the fluid and mass elements at structural integration points with coupling. When the messag file indicates that some structural segments have partial coupling, this file can be used to check the unconnected segment integration points. File “bac_flu_coupling.dyn” contains shell elements where free

fluid faces have coupled with the structural segments and mass elements at free fluid face integration points with coupling. These files are only for visualization of the coupling and serve no other purpose.

*BOUNDARY_ACOUSTIC_IMPEDANCE

Purpose: Define a segment set to prescribe the acoustic impedance of acoustic volume element (type 8 and type 14 solid elements) faces.

Card 1	1	2	3	4	5	6	7	8
Variable	SSID	ZEE						
Type	I	F						
Default	none	none						

VARIABLE	DESCRIPTION
SSID	Segment set ID, see *SET_SEGMENT
ZEE	Value of the acoustic impedance ρc

Remarks:

1. The effect of the boundary impedance on the acoustic cavity response is incorporated in the forcing vector. Solutions are conditionally stable, with low values of impedance relative to the impedance of the *MAT_ACOUSTIC elements causing instabilities. Reducing the factor of safety on the time step extends the range of applicability, however it is recommended that pressure release conditions be handled by leaving the boundary free rather than by providing a relatively low boundary acoustic impedance value. A warning is issued if the boundary impedance value is less than 25 percent of the *MAT_ACOUSTIC impedance. A value less than 1 percent of the *MAT_ACOUSTIC impedance is considered to be an error.
2. Special allowance is made for cases when both *LOAD_SEGMENT set pressures and the *BOUNDARY_ACOUSTIC_IMPEDANCE are defined on the same segments. In this event a nonreflecting entrant boundary condition is assumed. The pressures in the LOAD_SEGMENT_SET definition are treated as incoming incident pressure. Pressure waves within the *MAT_ACOUSTIC domain striking this boundary will exit the model. In contrast, a *LOAD_SEGMENT_SET on *MAT_ACOUSTIC volume faces in the absence of *BOUNDARY_ACOUSTIC_IMPEDANCE acts as a time-dependent, total pressure constraint and pressure waves within the *MAT_ACOUSTIC domain striking this boundary will be reflected back into the model.

*BOUNDARY_ACOUSTIC_MAPPING

Purpose: Define a set of elements or segments on structure for mapping structural nodal velocity to acoustic volume boundary.

Card 1	1	2	3	4	5	6	7	8
Variable	SSID	STYP						
Type	I	I						
Default	none	0						

VARIABLE	DESCRIPTION
SSID	Set or part ID
STYP	Set type: EQ.0: part set ID, see *SET_PART, EQ.1: part ID, see *PART, EQ.2: segment set ID, see *SET_SEGMENT.

Remarks:

1. If acoustic elements are not overlapping with structural elements, this keyword passes structural velocity to acoustic volume boundary, for subsequent frequency domain acoustic computation.

*BOUNDARY_ALE_MAPPING

Purpose: This card maps ALE data histories from a previous run to a region of elements. Data are read or written in a mapping file called by the prompt "map=" on the command line (see Remarks 4 and 5). To map data at the initial time (not the histories) to all the ALE domain (not just a region of elements) see *INITIAL_ALE_MAPPING.

The following transitions are allowed:

1D → 2D	2D → 2D	3D → 3D
1D → 3D	2D → 3D	

Card 1	1	2	3	4	5	6	7	8
Variable	ID	TYP	AMMSID	IVOLTYP	BIRTH	DEATH	DTOUT	INI
Type	I	I	I	I	F	F	F	I
Default	none	none	none	none	0.0	10^{20}	time step	0

Card 2	1	2	3	4	5	6	7	8
Variable	THICK	RADIUS	X1	Y1	Z1	X2	Y2	Z2
Type	F	F	F	F	F	F	F	F
Default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Card 3	1	2	3	4	5	6	7	8
Variable	X0	Y0	Z0	VECID				
Type	F	F	F	I				
Default	0.0	0.0	0.0	none				

VARIABLE **DESCRIPTION**

ID
Part ID or part set ID or element set ID

TYP
Type of "ID" (see Remark 1):

EQ.0: part set ID.

EQ.1: part ID.

EQ.2: shell set ID.

EQ.3: solid set ID.

AMMSID
Set ID of ALE multi-material groups defined in *SET_MULTI-MATERIAL_GROUP. See Remark 1.

IVOLTYP
Type of volume containing the selected elements for the mapping. The absolute value of IVOLTYPE indicates the type of volume and the sign indicates whether the data is being read of written.

Volume Type

|IVOLTYP|.EQ.1: Spherical surface with thickness (THICK).

|IVOLTYP|.EQ.2: Box.

|IVOLTYP|.EQ.3: Cylindrical surface with thickness (THICK)

|IVOLTYP|.EQ.4: All the elements defined by ID.

Read/Write

IVOLTYP.LT.0: data from the mapping file are read for the elements of this volume.

IVOLTYP.GT.0: data from the elements of this volume are written in the mapping file.

BIRTH
Birth time to write or read the mapping file. If a mapping file is written, the next run reading this file will begin at time BIRTH if this parameter for this next run is not larger.

DEATH
Death time to write or read the mapping file. If a mapping file is written, the next run will stop to read this file at time DEATH if this parameter for this next run is not smaller.

DTOUT
Time interval between outputs in the mapping file. This parameter is only used to write in the mapping file.

VARIABLE	DESCRIPTION
INI	Flag to initialize all the ALE domain of the next run: EQ.0: No initialization EQ.1: Initialization. *INITIAL_ALE_MAPPING will have to be in the input deck of the next run to read the data from the mapping file. The initial time of the next run will be BIRTH.
THICK	Thickness for the element selection using surfaces.
RADIUS	Radius for abs(IVOLTYP) = 1 and abs(IVOLTYP) = 3.

If abs(IVOLTYP).EQ.1:

X1	X1 is the *x*-coordinate of the sphere center.
Y1	Y1 is the *y*-coordinate of the sphere center.
Z1	Z1 is the *z*-coordinate of the sphere center.
X2, Y2, Z2	Ignored

If abs(IVOLTYP).EQ.2:

X1	X1 is the *x*-coordinate of the box's minimum point.
Y1	Y1 is the *y*-coordinate of the box's minimum point.
Z1	Z1 is the *z*-coordinate of the box's minimum point.
X2	X2 is the *x*-coordinate of the box's maximum point.
Y2	Y2 is the *y*-coordinate of the box's maximum point.
Z2	Z2 is the *z*-coordinate of the box's maximum point.

VARIABLE	DESCRIPTION
If abs(IVOLTYP).EQ.3:	
X1	X1 is the *x*-coordinate of a point on the cylinder's axis.
Y1	Y1 is the *y*-coordinate of a point on the cylinder's axis.
Z1	Z1 is the *z*-coordinate of a point on the cylinder's axis.
X2	X2 is the *x*-coordinate of a vector parallel to the cylinder's axis.
Y2	Y2 is the *y*-coordinate of a vector parallel to the cylinder's axis.
Z2	Z2 is the *z*-coordinate of a vector parallel to the cylinder's axis.
If abs(IVOLTYP).EQ.4:	
X1, Y1, Z1	ignored
X2, Y2, Z2	ignored
End if	
X0	Origin position in global *x*-direction. See Remark 2.
Y0	Origin position in global *y*-direction. See Remark 2.
Z0	Origin position in global *z*-direction. See Remark 2.
VECID	ID of the symmetric axis defined by *DEFINE_VECTOR. See Remark 3.

Remarks:

1. **Mapping of Multi-Material Groups.** The routines of this card need to know which mesh will be initialized with the mapping data and more specifically which multi-material groups. The first 2 parameters (ID and TYP) defines the mesh and the third one (AMMSID) refer to the *SET_MULTI-MATERIAL_-GROUP_LIST card. This card will define a list of material groups in the current run. The rank in this list should match the rank of the multi-material groups from the previous run (as a reminder the ranks of multi-material groups are defined by *ALE_MULTI-MATERIAL_GROUP). For instance, if the previous model has 3 groups, the current one has 5 groups, and the following mapping is wanted:

The 1st group (previous) ⇒ the 3rd group (current),

The 2nd group (previous) ⇒ the 5th group (current) and,

The 3rd group (previous) ⇒ the 4th group (current).

Then, the *SET_MULTI-MATERIAL_GROUP_LIST card should be as follows:

```
*SET_MULTI-MATERIAL_GROUP_LIST
300
3,5,4
```

2. **Origin.** The data can be mapped in different parts of the mesh by defining the origin of the coordinate system (X0, Y0, Z0).

3. **Orientation Vector: VECID.** For a mapping file created by a previous asymmetric model, the symmetric axis orientation in the current model is specified by VECID. For a mapping file created by a 3D or 1D spherical model, the vector VECID is read but ignored. The definitions of X0, Y0, Z0 and VECID change in the case of the following mappings:

   a) plain strain 2D (ELFORM = 13 in *SECTION_ALE2D) to plain strain 2D

   b) plain strain 2D to 3D

   While, VECID still defines the y-axis in the 2D domain, the 3 first parameters in *DEFINE_VECTOR, additionally, define the location of the origin. The 3 last parameters defines a position along the y-axis. For this case when 2D data is used in a 3D calculation the point X0, Y0, Z0 together with the vector, VECID, define the plane.

4. **Mapping File.** To make one mapping: only the command-line argument “map=” is necessary. If IVOLTYP is positive, the mapping file will be created and ALE data histories will be written in this file. If IVOLTYP is negative the mapping file will be read and ALE data histories will be used to interpolate the ALE variables of the selected elements. This file contains the following nodal and element data:

   - nodal coordinates
   - nodal velocities
   - part ids
   - element connectivities
   - element centers
   - densities
   - volume fractions

- stresses
- plastic strains
- internal energies
- bulk viscosities
- relative volumes

5. **Successive Mappings.** To make several successive mapping: the prompt “map1=” is necessary. If IVOLTYP is positive and the prompt “map1=” is in the command line, the ALE data are written to the mapping file given by “map1=”. If IVOLTYP is negative and the prompt “map=” is in the command line, ALE data are read from the mapping file given by “map=”.

*BOUNDARY_AMBIENT

Purpose: This command defines ALE “ambient” type element formulations (please see Remarks 1, 2 and 5).

Card 1	1	2	3	4	5	6	7	8
Variable	SETID	MMG	AMBTYP					
Type	I	I	I					
Default	none	none	none					

Optional Card. Additional optional card for AMBTYP = 4 with curves

Card 3	1	2	3	4	5	6	7	8
Variable	LCID1	LCID2						
Type	I	I						
Default	none	none						

VARIABLE **DESCRIPTION**

SETID The ambient element set ID for which the thermodynamic state is being defined. The element set can be *SET_SOLID for a 3D ALE model, *SET_SHELL for a 2D ALE model or *SET_BEAM for a 1D ALE model.

MMG ALE multi-material group ID.

AMBTYP Ambient element type:

EQ.4: Pressure inflow/outflow (see Remarks 1 and 2)

EQ.5: Receptor for blast load (See *LOAD_BLAST_ENHANCED)

VARIABLE	DESCRIPTION
LCID1	A load curve ID for internal energy per unit reference volume (Please see Remark 4 and read the beginning of the EOS section for details). If *EOS_IDEAL_GAS is being used, this ID then refers to a temperature load curve ID.
LCID2	Load curve ID for relative volume, $v_r = \left(\frac{v}{v_0} = \frac{\rho_0}{\rho}\right)$. (Please see Remark 3 and read the beginning of the EOS section for details).

Remarks:

1. The term "ambient" refers to a medium that has predetermined thermodynamic state throughout the simulation. All "ambient" elements will have its thermodynamic state reset back to this predetermined state every cycle. If this state is defined via the *EOS card, then this predetermined thermodynamic state is constant throughout the simulation. If it is defined via the curves of the 2nd line for AMBTYP = 4, its thermodynamic state will vary according to these defined load curves. "Ambient" elements are sometimes also referred to as "reservoir" elements as they may be used to simulate semi-infinite region.

2. In general, a thermodynamic state of a non-reacting and no-phase-change material may be defined by 2 thermodynamic variables. By defining (a) an internal energy per unit reference volume load curve (or a temperature load curve if using *EOS_IDEAL_GAS) and (b) a relative volume load curve, the pressure as a function of time for this ambient part ID can be computed directly via the equation of state (*EOS_…).

3. A reference specific volume, $v_0 = \frac{1}{\rho_0}$, is the inverse of a reference density, ρ_0. The reference density is defined as the density at which the material is under a reference or nominal state. Please refer to the *EOS section for additional explanation on this.

4. The internal energy per unit reference volume may be defined as

$$e_{\mathrm{ipv0}} = \frac{C_v T}{v_0}.$$

   The specific internal energy (or internal energy per unit mass) is defined as $C_\nu T$.

5. This card does not require AET under *SECTION_SOLID or SECTION_ALE2D or SECTION_ALE1D card

*BOUNDARY_AMBIENT_EOS

Purpose: This command defines the IDs of 2 load curves: (1) internal energy per unit reference volume (or temperature if using *EOS_IDEAL_GAS) and (2) relative volume. These 2 curves completely prescribe the thermodynamic state as a function of time for any ALE or Eulerian part with an "ambient" type element formulation (please see Remark 4).

Card 1	1	2	3	4	5	6	7	8
Variable	PID	LCID1	LCID2					
Type	I	I	I					
Default	none	none	none					

VARIABLE	DESCRIPTION
PID	The ambient Part ID for which the thermodynamic state is being defined.
LCID1	Load curve ID (*DEFINE_CURVE or *DEFINE_CURVE_FUNCTION) for internal energy per unit reference volume (please read the beginning of the EOS section for details). If *EOS_IDEAL_GAS is being used, this ID then refers to a temperature load curve ID.
LCID2	Load curve ID (*DEFINE_CURVE or *DEFINE_CURVE_FUNCTION) for relative volume, $v_r = \left(\frac{v}{v_0} = \frac{\rho_0}{\rho}\right)$. (Please read the beginning of the EOS section for details).

Remarks:

1. The term "ambient" refers to a medium that has predetermined thermodynamic state throughout the simulation. All "ambient" parts/elements will have its thermodynamic state reset back to this predetermined state every cycle. If this state is defined via the *EOS card, then this predetermined thermodynamic state is constant throughout the simulation. If it is defined via this card, *BOUNDARY_AMBIENT_EOS, then its thermodynamic state will vary according to these defined load curves. "Ambient" part is sometimes also referred to as "reservoir" part as it may be used to simulate semi-infinite region.

2. In general, a thermodynamic state of a non-reacting and no-phase-change material may be defined by 2 thermodynamic variables. By defining (a) an internal energy per unit reference volume load curve (or a temperature load curve if using *EOS_IDEAL_GAS) and (b) a relative volume load curve, the pressure as a function of time for this ambient part ID can be computed directly via the equation of state (*EOS_…).

3. A reference specific volume, $v_0 = \frac{1}{\rho_0}$, is the inverse of a reference density, ρ_0. The reference density is defined as the density at which the material is under a reference or nominal state. Please refer to the *EOS section for additional explanation on this.

4. The internal energy per unit reference volume may be defined as

$$e_{\mathrm{ipv0}} = \frac{C_v T}{v_0}.$$

   The specific internal energy (or internal energy per unit mass) is defined as $C_\nu T$.

5. This card is only to be used with "ambient" element type as defined by the parameters under the *SECTION_SOLID card:

   a) ELFORM = 7, or

   b) ELFORM = 11 and AET = 4, or

   c) ELFORM = 12 and AET = 4.

Example:

Consider an ambient ALE part ID 1 which has its internal energy per unit reference volume in a load curve ID 2 and relative volume load curve ID 3:

```
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|...8
*BOUNDARY_AMBIENT_EOS
$      PID  e/T_LCID rvol_LCID
         1         2         3
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|...8
```

BOUNDARY_CONVECTION_*OPTION

Available options include:

SEGMENT

SET

Purpose: Apply a convection boundary condition on a SEGMENT or SEGMENT_SET for a thermal analysis. Two cards are defined for each option.

Card 1 for SET keyword option.

Card 1	1	2	3	4	5	6	7	8
Variable	SSID							
Type	I							
Default	none							

Card 1 for SEGMENT keyword option.

Card 1	1	2	3	4	5	6	7	8
Variable	N1	N2	N3	N4				
Type	I	I	I	I				
Default	none	none	none	none				

Card 2	1	2	3	4	5	6	7	8
Variable	HLCID	HMULT	TLCID	TMULT	LOC			
Type	I	F	I	F	I			
Default	none	0.	none	0.	0			

VARIABLE	DESCRIPTION
SSID	Segment set ID, see *SET_SEGMENT.
N1, N2, ….	Node ID's defining segment.
HLCID	Convection heat transfer coefficient, h. This parameter can reference a load curve ID (see *DEFINE_CURVE) or a function ID (see *DEFINE_FUNCTION and Remark 1). When the reference is to a curve, HLCID has the following interpretation: GT.0: h is given as a function of time, t. The curve consists of $(t, h(t))$ data pairs. EQ.0: h is a constant defined by the value HMULT. LT.0: h is given as a function of temperature, T_{film}. The curve consists of (T_{film}, h) data pairs. Enter $\vert\text{HLCID}\vert$ on the DEFINE_CURVE keyword. (see Remark 1)
HMULT	Convection heat transfer coefficient, h, curve multiplier.
TLCID	Environment temperature, T_∞. This parameter can reference a load curve ID (see *DEFINE_CURVE) or a function ID (see *DEFINE_FUNCTION and Remark 2). When the reference is to a curve, TLCID has the following interpretation: GT.0: T_∞ is defined by a curve indexed by time consisting of $(t, T_\infty(t))$ data pairs. EQ.0: T_∞ is a constant defined by the value TMULT.
TMULT	Environment temperature, T_∞, curve multiplier.
LOC	For a thick thermal shell, the convection will be applied to the surface identified by LOC. See parameter, THSHEL, on the *CONTROL_SHELL keyword. EQ.-1: lower surface of thermal shell element EQ.0: middle surface of thermal shell element EQ.1: upper surface of thermal shell element

Remarks:

1. A convection boundary condition is calculated using $\dot{q}'' = h(T_{\text{surface}} - T_\infty)$ where h is the heat transfer coefficient, and $T_{\text{surface}} - T_\infty$ is a temperature potential. If h is a function of temperature, h is evaluated at the average or "film"

temperature defined by
$T_{film} = (T_{\mathrm{surface}} + T_\infty)/2$..

2. If HLCID references a DEFINE_FUNCTION, the following function arguments are allowed $f(x, y, z, v_x, v_y, v_Z, T, T_\infty, t)$ where:

$$x, y, z = \text{segment centroid coordinates}$$

$$v_x, v_y, v_z = \text{segment centroid velocity component}$$

$$T = \text{segment centroid temperature}$$

$$T_\infty = \text{environment temperature, } T_\infty$$

$$t = \text{solution time}$$

3. If TLCID references a DEFINE_FUNCTION, the following function arguments are allowed $f(x, y, z, v_x, v_y, v_z, t)$ where:

$$x, y, z = \text{segment centroid coordinates}$$

$$v_x, v_y, v_z = \text{segment centroid velocity components}$$

$$t = \text{solution time}$$

*BOUNDARY_COUPLED

Purpose: Define a boundary that is coupled with an external program. Two cards are required for each coupled boundary

Card 1	1	2	3	4	5	6	7	8
Variable	ID	TITLE						
Type	I	A70						

Card 2	1	2	3	4	5	6	7	8
Variable	SET	TYPE	PROG					
Type	I	I	I					
Default	none	none	none					

VARIABLE	DESCRIPTION
ID	ID for this coupled boundary
TITLE	Descriptive name for this boundary
SET	Node set ID
TYPE	Coupling type: EQ.1: node set with force feedback EQ.2: node set for multiscale spotwelds
PROG	Program to couple to EQ.1: MPP-DYNA

Remarks:

This option is only available in the MPP version, and allows for loose coupling with other MPI programs using a “multiple program” execution method. Currently it is only useful when linking with MPP-DYNA for the modeling of multiscale spotwelds (type = 2, prog = 1). See *INCLUDE_MULTISCALE_SPOTWELD for information about using this capability.

*BOUNDARY_CYCLIC_{OPTION}

OPTION allows an optional ID to be given that applies each cyclic definition

ID

Purpose: Define nodes in boundary planes for cyclic symmetry.

These boundary conditions can be used to model a segment of an object that has rotational symmetry such as an impeller, i.e., Figure 5-1. The segment boundary, denoted as a side 1 and side 2, may be curved or planar. In this section, a paired list of points are defined on the sides that are to be joined.

ID Card. Additional card for ID keyword option.

ID	1	2	3	4	5	6	7	8
Variable	ID	HEADING						
Type	I	A70						

Card 1	1	2	3	4	5	6	7	8
Variable	XC	YC	ZC	NSID1	NSID2	IGLOBAL	ISORT	
Type	F	F	F	I	I	I	I	
Default	none	none	none	none	none	0	0	

VARIABLE	DESCRIPTION
XC	x-component axis vector of axis of rotation
YC	y-component axis vector of axis of rotation
ZC	z-component axis vector of axis of rotation
NSID1	Node set ID for first boundary (side 1, see Figure 5-1).

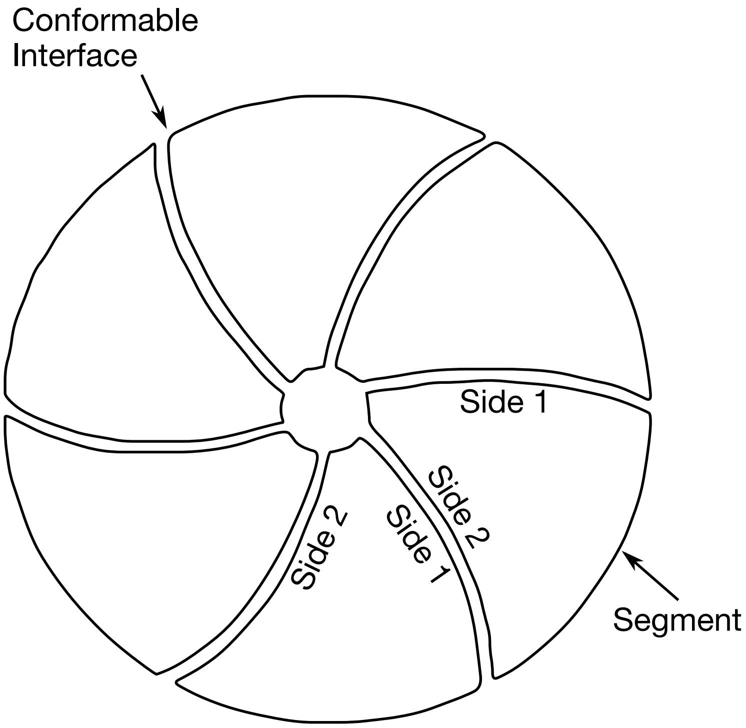


Figure 5-1. With axi-symmetric cyclic symmetry, only one segment is modeled.

VARIABLE	DESCRIPTION
NSID2	Node set ID for second boundary (side 2, see Figure 5-1). Each node in this set is constrained to its corresponding node in the first node set. Node sets NSID1 and NSID2 must contain the same number of nodal points. The shape of the two surfaces formed by the two node sets need not be planar but the shapes should match.
IGLOBAL	Flag for repeating symmetry: EQ.0: Axi-symmetric cyclic symmetry (default) EQ.1: Repeating symmetry in planes normal to global X EQ.2: Repeating symmetry in planes normal to global Y EQ.3: Repeating symmetry in planes normal to global Z
ISORT	Set to 1 for automatic sorting of nodes in node sets. See Remark 2.

Remarks:

1. Each node set should generally be boundaries of the model.

2. Prior to version 970, it was assumed that the nodes are correctly ordered within each set, i.e. the nth node in NSID1 is equivalent to the nth node in NSID2. In version 970 and later versions, if the ISORT flag is active, the nodes in NSID2 are automatically sorted to achieve equivalence, so the nodes can be picked by the quickest available method. However, for axi-symmetric cyclic symmetry (IGLOBAL = 0), it is assumed that the axis passes through the origin, i.e., only globally defined axes of rotation are possible.

*BOUNDARY_DE_NON_REFLECTING

Purpose: Define a non-reflecting boundary for discrete element.

Card 1	1	2	3	4	5	6	7	8
Variable	NSID							
Type	I							
Default	none							
Remarks	1, 2							

VARIABLE	DESCRIPTION
NSID	Node set ID, see *SET_SEGMENT.

Remarks:

1. Non-reflecting boundaries are used on the exterior boundaries of an analysis model of an infinite domain, such as a half-space to prevent artificial stress wave reflections generated at the model boundaries form reentering the model and contaminating the results.

BOUNDARY_FLUX_*OPTION

Available options include:

SEGMENT

SET

Purpose: Apply a flux boundary condition on a SEGMENT or SEGMENT_SET for a thermal analysis. Two or more cards are defined for each option. History variables can be associated with the boundary condition which will invoke a call to a user defined boundary flux subroutine for computing the flux.

Card 1 for SET option.

Card 1	1	2	3	4	5	6	7	8
Variable	SSID							
Type	I							
Default	none							

Card 1 for SEGMENT option.

Card 1	1	2	3	4	5	6	7	8
Variable	N1	N2	N3	N4				
Type	I	I	I	I				
Default	none	none	none	none				

Card 2	1	2	3	4	5	6	7	8
Variable	LCID	MLC1	MLC2	MLC3	MLC4	LOC	NHISV	
Type	I	F	F	F	F	I	I	
Default	none	0.	0.	0.	0.	0	0	

Define as many cards as necessary to initialize NHISV history variables.

Card 3	1	2	3	4	5	6	7	8
Variable	HISV1	HISV2	HISV3	HISV4	HISV5	HISV6	HISV7	HISV8
Type	F	F	F	F	F	F	F	F
Default	0.	0.	0.	0.	0.	0.	0.	0.

VARIABLE	DESCRIPTION
SSID	Segment set ID, see *SET_SEGMENT
N1, N2, …	Node IDs that define the segment

VARIABLE **DESCRIPTION**

LCID This parameter can reference a load curve ID (see *DEFINE_CURVE) or a function ID (see *DEFINE_FUNCTION and Remark 2e) for heat flux. When the reference is to a curve, LCID has the following interpretation:

GT.0: the flux is defined by a curve consisting of (time, flux) data pairs using the DEFINE_CURVE keyword. The flux value applied to the nodal points is the curve value multiplied by the values MLC1, MLC2, MLC3, and MLC4, respectively.

EQ.0: a constant flux is applied to each node defined by the values MLC1, MLC2, MLC3, and MLC4, respectively.

LT.0: the flux is defined by a curve consisting of (temperature, flux) data pairs using the DEFINE_CURVE keyword. The flux value applied to the nodal points is the curve value multiplied by the values MLC1, MLC2, MLC3, and MLC4. Enter |-LCID| on the DEFINE_CURVE keyword.

MLC1 Curve multiplier at node N1.

MLC2 Curve multiplier at node N2.

MLC3 Curve multiplier at node N3.

MLC4 Curve multiplier at node N4.

LOC For a thick thermal shell, the flux will be applied to the surface identified by LOC. See parameter, THSHEL, on the *CONTROL_SHELL keyword.

EQ.-1: lower surface of thermal shell element

EQ.0: middle surface of thermal shell element

EQ.1: upper surface of thermal shell element

NHISV Number of history variables associated with the flux definition:

GT.0: A user defined subroutine will be called to compute the flux. See Remark 1.

HISV1 Initial value of history variable 1

HISV2 Initial value of history variable 2

VARIABLE	DESCRIPTION
⋮	⋮
HISV*n*	Initial value of history variable *n*, where *n* = NHISV

Remarks:

1. The segment normal has no bearing on the flux. A positive flux transfers energy into the volume; a negative flux transfers energy out of the volume.

2. Flux can be defined by:

   a) When LCID = 0, a constant flux is applied to each node defined by the values MLC1, MLC2, MLC3, and MLC4, respectively.

   b) When LCID > 0, the flux is defined by a curve consisting of (time, flux) data pairs using the DEFINE_CURVE keyword. The flux value applied to the nodal points is the curve value multiplied by the values MLC1, MLC2, MLC3, and MLC4, respectively.

   c) When LCID < 0, the flux is defined by a curve consisting of (temperature, flux) data pairs using the DEFINE_CURVE keyword. The flux value applied to the nodal points is the curve value multiplied by the values MLC1, MLC2, MLC3, and MLC4. Enter |LCID| on the DEFINE_-CURVE keyword.

   d) When NHSIV > 0, the user subroutine

   ```
   subroutine usrflux(fl, flp, ...)
   ```

   will be called to compute the heat flux (fl). For more details see Appendix S.

   e) If LCID references a DEFINE_FUNCTION, the following function arguments are allowed $f(x,y,z,v_x,v_y,v_z,T,T_\infty,t)$ where:

   $$x, y, z = \text{segment centroid coordinates}$$
   $$v_x, v_y, v_z = \text{segment centroid velocity components}$$
   $$T = \text{segment centroid temperature}$$
   $$T_\infty = \text{environment temperature, } T_\infty$$
   $$t = \text{solution time}$$

3. This keyword is supported in the SPH elements to define the flux boundary conditions for a thermal or coupled thermal/structural analysis. The values n_1,

n_2, n_3, n_4 from the SPH particles or segments are used to define the flux segments.

*BOUNDARY_MCOL

Purpose: Define parameters for MCOL coupling. The MCOL Program is a rigid body mechanics program for modeling the dynamics of ships. See Remark 1 for more information.

Card 1	1	2	3	4	5	6	7	8
Variable	NMCOL	MXSTEP	ETMCOL	TSUBC	PRTMCOL			
Type	I	I	F	F	F			
Default	2	none	0.0	0.0	none			
Remarks			2					

Ship Card. Include NMCOL cards, one for each ship.

Card 2	1	2	3	4	5	6	7	8
Variable	RBMCOL	MCOLFILE						
Type	I	A60						
Default		none						

VARIABLE	DESCRIPTION
NMCOL	Number of ships in MCOL coupling.
MXSTEP	Maximum of time step in MCOL calculation. If the number of MCOL time steps exceeds MXSTEP, then LS-DYNA will terminate.
ETMCOL	Uncoupling termination time, see Remark 2 below. EQ.0.0: set to LS-DYNA termination time
TSUBC	Time interval for MCOL subcycling. EQ.0.0: no subcycling

VARIABLE	DESCRIPTION
PRTMCOL	Time interval for output of MCOL rigid body data.
RBMCOL	LS-DYNA rigid body material assignment for the ship.
MCOLFILE	Filename containing MCOL input parameters for the ship.

Remarks:

1. The basis for MCOL is a convolution integral approach for simulating the equations of motion. A mass and inertia tensor are required as input for each ship. The masses are then augmented to include the effects of the mass of the surrounding water. A separate program determines the various terms of the damping/buoyancy force formulas which are also input to MCOL. The coupling is accomplished in a simple manner: at each time step LS-DYNA computes the resultant forces and moments on the MCOL rigid bodies and passes them to MCOL. MCOL then updates the positions of the ships and returns the new rigid body locations to LS-DYNA. A more detailed theoretical and practical description of MCOL can be found in a separate report (to appear).

2. After the end of the LS-DYNA/MCOL calculation, the analysis can be pursued using MCOL alone. ETMCOL is the termination time for this analysis. If ETMCOL is lower than the LS-DYNA termination time, the uncoupled analysis will not be activated.

3. The MCOL output is set to the files mcolout (ship position) and mcolenergy (energy breakdown). In LS-PrePost, mcolout can be plotted through the rigid body time history option and MCOLENERGY

*BOUNDARY_NON_REFLECTING

Purpose: Define a non-reflecting boundary. This option applies to continuum domains modeled with solid elements. For geomechanical problems this option is important for limiting the spatial extent of the finite element mesh and thus the number of solid elements.

Card 1	1	2	3	4	5	6	7	8
Variable	SSID	AD	AS					
Type	I	F	F					
Default	none	0.0	0.0					
Remarks	1, 2	3	3					

VARIABLE	DESCRIPTION
SSID	Segment set ID, see *SET_SEGMENT.
AD	Default activation flag for dilatational waves. EQ.0.0: on NE.0.0: off
AS	Default activation flag for shear waves. EQ.0.0: on NE.0.0: off

Remarks:

1. Non-reflecting boundaries defined with this keyword are only used with three-dimensional solid elements. Boundaries are defined as a collection of segments, and segments are equivalent to element faces on the boundary. Segments are defined by listing the corner nodes in either a clockwise or counterclockwise order.

2. Non-reflecting boundaries are used on the exterior boundaries of an analysis model of an infinite domain, such as a half-space to prevent artificial stress wave reflections generated at the model boundaries form reentering the model

and contaminating the results. Internally, LS-DYNA computes an impedance matching function for all non-reflecting boundary segments based on an assumption of linear material behavior. Thus, the finite element mesh should be constructed so that all significant nonlinear behavior is contained within the discrete analysis model.

3. With the two optional switches, the influence of reflecting waves can be studied.

4. During the dynamic relaxation phase (optional), nodes on non-reflecting segments are constrained in the normal direction. Nodal forces associated with these constraints are then applied as external loads and held constant in the transient phase while the constraints are replaced with the impedance matching functions. In this manner, soil can be quasi-statically prestressed during the dynamic relaxation phase and dynamic loads (with non-reflecting boundaries) subsequently applied in the transient phase.

5. In explicit analyses this command has the side effect of reducing the default value for the time step scale factor from 0.9 to 0.667. A nonzero value of TSSFAC in *CONTROL_TIMESTEP will override that default.

*BOUNDARY_NON_REFLECTING_2D

Purpose: Define a non-reflecting boundary. This option applies to continuum domains modeled with two-dimensional solid elements in the xy plane. For geomechanical problems, this option is important for limiting the size of the models.

Card 1	1	2	3	4	5	6	7	8
Variable	NSID							
Type	I							
Default	none							
Remarks	1, 2							

VARIABLE	DESCRIPTION
NSID	Node set ID, see *SET_NODE. See Figure 5-2.

Remarks:

1. Non-reflecting boundaries defined with this keyword are only used with two-dimensional solid elements in either plane strain or axisymmetric geometries. Boundaries are defined as a sequential string of nodes moving counterclockwise around the boundary.

2. Non-reflecting boundaries are used on the exterior boundaries of an analysis model of an infinite domain, such as a half-space to prevent artificial stress wave reflections generated at the model boundaries from reentering the model and contaminating the results. Internally, LS-DYNA computes an impedance matching function for all non-reflecting boundary segments based on an assumption of linear material behavior. Thus, the finite element mesh should be constructed so that all significant nonlinear behavior in contained within the discrete analysis model.

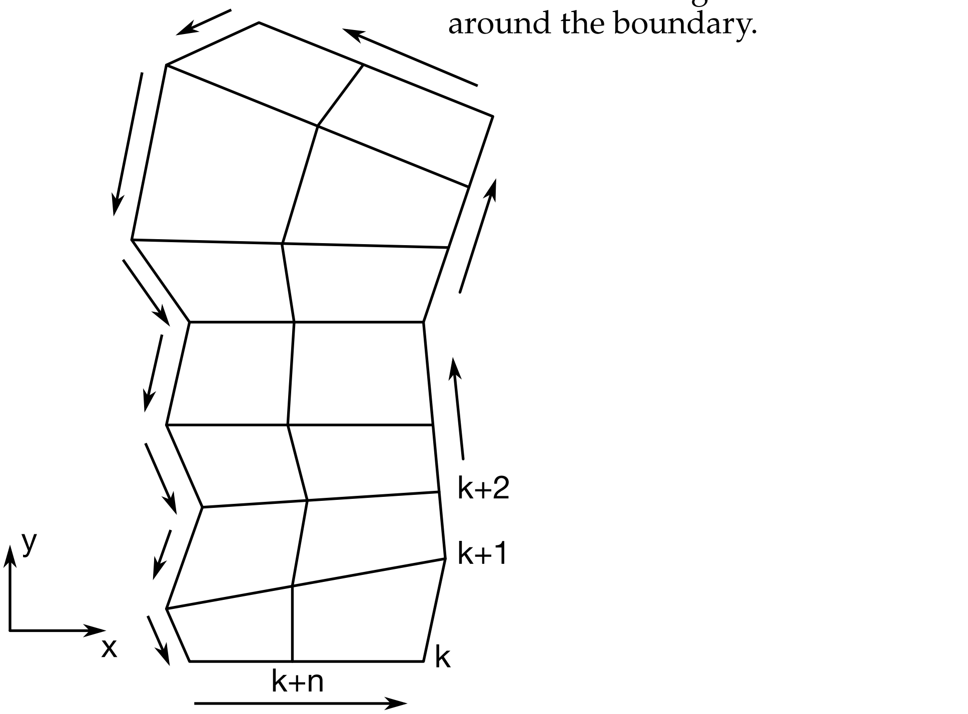

Define the nodes k, k+1, k+2, ...,
k+n while moving counterclockwise
around the boundary.
k+2
k+1
k
k+n
y
x

Figure 5-2. When defining a transmitting boundary in 2D define the node numbers in the node set in consecutive order while moving counterclockwise around the boundary.

*BOUNDARY_PAP

Purpose: Define pressure boundary conditions for pore air flow calculation, e.g. at structure surface exposed to atmospheric pressure.

Card 1	1	2	3	4	5	6	7	8
Variable	SEGID	LCID	CMULT	CVMASS	BLOCK	TBIRTH	TDEATH	CVRPER
Type	I				F	F	F	F
Default	none	none	none	none	0.0	0.0	1.e20	1.0
Remark				1, 2				3

VARIABLE **DESCRIPTION**

SEGID Segment set ID

LCID Load curve giving pore air pressure vs. time.

EQ.0: constant pressure assumed equal to CMULT

CMULT Factor on curve or constant pressure head if LCID = 0

CVMASS Initial mass of a control volume next to the segment set SETID

BLOCK Contact blockage effect,

EQ.0: When all segments in SEGID are subject to the pressure defined by LCID and CMULT;

EQ.1: When only elements in SEGID not involved in contact are subject to the pressure defined by LCID and CMULT.

TBIRTH Time at which boundary condition becomes active

TDEATH Time at which boundary condition becomes inactive

CVRPER Permeability factor of cover material, where cover refers to a shell layer coating the surface of the solid. Default value is 1.0 when it is not defined. See Remark 3 below.

$$0.0 \leq \text{CVRPER} \leq 1.0$$

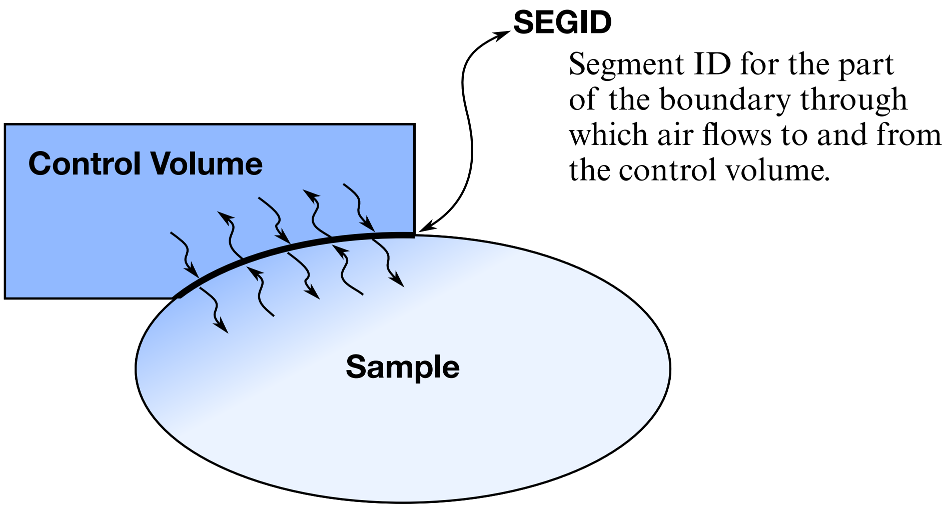

SEGID
Segment ID for the part of the boundary through which air flows to and from the control volume.
Control Volume
Sample

Figure 5-3. Air flows between the control volume and the sample. CVMASS specifies the control volume's initial mass, and CVMULT sets the initial pressure.

Remarks:

1. All structure surfaces subject to specified pressure have to be defined.

2. A non-zero CVMASS, together with a non-zero CMULT and an un-defined LCID, can be used to simulate air mass transfer between a control volume and a test specimen containing pore air. The control volume is assumed to have a fixed volume, and have initial pressure of CMULT and initial mass of CVMASS. Air mass transfer happens between control volume and its neighboring specimen. Such mass transfer results in pressure change in control volume and test specimen.

3. CVRPER allows users to model the porosity properties of the cover material. If SEGID is covered by a material of very low permeability (e.g., coated fabric), it is appropriate to set CVRPER = 0. In this case, P_c, the pressure calculated assuming no boundary condition, is applied to SEGID. If SEGID is not covered by any material, it is appropriate to set CVRPER = 1, the default value. In this case, the applied pressure becomes P_b, the boundary pressure determined by CMULT and LCID.

BOUNDARY_PORE_FLUID_*OPTION

Available options include:

PART

SET

Purpose: Define parts that contain pore fluid. Defaults are given on *CONTROL_PORE_FLUID.

Card 1	1	2	3	4	5	6	7	8
Variable	P(S)ID	WTABLE	PF_RHO	ATYPE	PF_BULK	ACURVE	WTCUR	SUCLIM
Type	I	F	F	I	F	I	I	F
Default	none	*	*	*	*	0	0	0.

* Defaults are taken from *CONTROL_PORE_FLUID

VARIABLE	DESCRIPTION
PID, PSID	Part ID (PID) or Part set ID, see *PART and *SET_PART. All elements within the part must lie below the water table.
WTABLE	Z-coordinate at which pore pressure = 0 (water table)
PF_RHO	Density of pore water in soil skeleton: EQ.0: Default density specified on *CONTROL_PORE_FLUID card is used.
ATYPE	Analysis type for Parts: EQ.0: Default to value specified on *CONTROL_PORE_FLUID EQ.1: Undrained analysis EQ.2: Drained analysis EQ.3: Time dependent consolidation (coupled) EQ.4: Consolidate to steady state (uncoupled) EQ.5: Drained in dynamic relaxation, undrained in transient

VARIABLE DESCRIPTION

PF_BULK Bulk modulus of pore fluid:

EQ.0: Default to value specified on *CONTROL_PORE_FLUID

ACURVE Curve of analysis type vs time (see notes below)

WTCUR Curve of water table (z-coordinate) vs time

SUCLIM Suction limit (defined in head, i.e. length units). Must not be negative. See remarks.

Remarks:

This card must be present for all parts having pore water.

The density on this card is used only to calculate pressure head. To ensure the correct gravity loading, the density of the soil material should be increased to include the mass associated with the pore water.

The y-axis values of the curve of analysis type vs time can only be 1, 2 or 3. During dynamic relaxation, the analysis type will be taken from the first value on the curve

The default for SUCLIM is zero, meaning that the pore fluid cannot generate suction. To allow unlimited suction, set this parameter to a large positive number.

*BOUNDARY_PRECRACK

Purpose: Define pre-cracks in XFEM shell formulations 52 or 54 for purposes of fracture analysis.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	CTYPE	NP					
Type	I	I	I					
Default		1						

Precrack Point Cards. Include NP cards, one for each point in the pre-crack.

Card 2	1	2	3	4	5	6	7	8
Variable	X	Y	Z					
Type	F	F	F					
Default								

VARIABLE	DESCRIPTION
PID	Part ID where the pre-crack is located
CTYPE	Type of pre-crack: EQ.1: straight line
NP	Number of points defining the pre-crack
X, Y, Z	Coordinates of the points defining the pre-crack

*BOUNDARY_PRESCRIBED_ACCELEROMETER_RIGID

Purpose: Prescribe the motion of a rigid body based on experimental data obtained from accelerometers affixed to the rigid body.

Note: This feature is available starting with LS-DYNA 971R3.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	SOLV						
Type	I	I						
Default	none	1						

Accelerometer Cards. Define one card for each accelerometer affixed to the rigid body. Input is terminated when a "*" card is found. A minimum of three accelerometers are required (see Remarks below).

Card 2	1	2	3	4	5	6	7	8
Variable	NID	CID	LCIDX	LCIDY	LCIDZ			
Type	I	I	I	I	I			
Default	none	none	none	none	none			

VARIABLE	DESCRIPTION
PID	Part ID for rigid body whose motion is prescribed.
SOLV	Solver type: EQ.1: Gaussian elimination (default), EQ.2: linear regression
NID	Node ID corresponding to the location of the accelerometer.
CID	Coordinate system ID describing the orientation of the accelerometer's local axes (see *DEFINE_COORDINATE_NODES). All nodes must reside on the same part. Set FLAG = 1.

VARIABLE	DESCRIPTION
LCIDX	Load curve ID containing the local x-acceleration time history from the accelerometer.
LCIDY	Load curve ID containing the local y-acceleration time history from the accelerometer.
LCIDZ	Load curve ID containing the local z-acceleration time history from the accelerometer.

Remarks:

1. Acceleration time histories from a minimum of three accelerometers each providing output from three channels are required. Load curves must have the same number of points and data must be uniformly spaced.

2. Local axes of the accelerometers must be orthogonal.

*BOUNDARY_PRESCRIBED_FINAL_GEOMETRY

The final displaced geometry for a subset of nodal points is defined. The nodes of this subset are displaced from their initial positions specified in the *NODE input to the final geometry along a straight line trajectory. A load curve defines a scale factor as a function of time that is bounded between zero and unity corresponding to the initial and final geometry, respectively. A unique load curve can be specified for each node, or a default load curve can apply to all nodes. The external work generated by the displacement field is included in the energy ratio calculation for the glstat file.

Card 1	1	2	3	4	5	6	7	8
Variable	BPFGID	LCIDF	DEATHD					
Type	I	I	F					
Default	0	0	infinity					

Node Cards. The next "*" keyword card terminates this input.

Card 2	1	2	3	4	5	6	7	8	9	10
Variable	NID	X		Y		Z		LCID	DEATH	
Type	I	F		F		F		I	F	
Default	none	0.		0.		0.		LCIDF	infinity	

VARIABLE **DESCRIPTION**

BPFGID ID for this set of imposed boundary conditions

LCIDF Default load curve ID. This curve varies between zero and unity.

DEATHD Default death time. At this time the prescribed motion is inactive and the nodal point is allowed to move freely.

NID Node ID for which the final position is defined. Nodes defined in this section must also appear under the *NODE input.

X x-coordinate of final geometry

VARIABLE	DESCRIPTION
Y	y-coordinate of final geometry
Z	z-coordinate of final geometry
LCID	Load curve ID. If zero the default curve ID, LCIDF, is used.
DEATH	Death time. If zero the default value, DEATHD, is used.

BOUNDARY_PRESCRIBED_MOTION_*OPTION1_{OPTION2}

Available options for *OPTION1* include:

NODE

SET

SET_BOX

SET_SEGMENT

RIGID

RIGID_LOCAL

SET_LINE

OPTION2 allows an optional ID to be given that applies either to the single node, node set or a rigid body.

ID

If a heading is defined with the ID, then the ID with the heading will be written at the beginning of the ASCII file, bndout.

Purpose: Define an imposed nodal motion (velocity, acceleration, or displacement) on a node or a set of nodes. Also velocities and displacements can be imposed on rigid bodies. If the local option is active the motion is prescribed with respect to the local coordinate system for the rigid body (See variable LCO for keyword *MAT_RIGID). Translational nodal velocity and acceleration specifications for rigid body nodes are allowed and are applied as described at the end of this section. For nodes on rigid bodies use the NODE option. Do not use the NODE option in r-adaptive problems since the node ID's may change during the adaptive step.

The SET_LINE option allows a node set to be generated including existing nodes and new nodes created from *h*-adaptive mesh refinement along the straight line connecting two specified nodes to be included in prescribed boundary conditions.

ID Card. Additional card for ID keyword option.

ID	1	2	3	4	5	6	7	8
Variable	ID	HEADING						
Type	I	A70						

Card 1	1	2	3	4	5	6	7	8
Variable	typeID	DOF	VAD	LCID	SF	VID	DEATH	BIRTH
Type	I	I	I	I	F	I	F	F
Default	none	none	0	none	1.	0	10^{28}	0.0

For the SET_BOX keyword option, define the following additional card.

Card 2	1	2	3	4	5	6	7	8
Variable	BOXID	TOFFSET	LCBCHK					
Type	I	I	I					
Default	none	0	0					

Additional card that is expected if DOF = 9, 10, 11 or VAD = 4 on the first card; otherwise skip this card.

Card 3	1	2	3	4	5	6	7	8
Variable	OFFSET1	OFFSET2	MRB	NODE1	NODE2			
Type	F	F	I	I	I			
Default	0.	0.	0	0	0			

For the SET_LINE keyword option, define the following additional card.

Card 2	1	2	3	4	5	6	7	8
Variable	NBEG	NEND						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
ID	Optional PRESCRIBED MOTION set ID to which this node, node set, segment set or rigid body belongs. This ID does not need to be unique.
HEADING	An optional descriptor for the given ID that will be written into the d3hsp file and the bndout file.
typeID	Node ID (NID in *NODE), nodal set ID (SID in *SET_NODE), segment set ID (SID in *SET_SEGMENT, see DOF = 12) or part ID (PID in *PART) for a rigid body.

DOF

Applicable degrees-of-freedom:

EQ.1: x-translational degree-of-freedom,

EQ.2: y-translational degree-of-freedom,

EQ.3: z-translational degree-of-freedom,

EQ.4: translational motion in direction given by the VID. Movement on plane normal to the vector is permitted.

EQ.-4: translational motion in direction given by the VID. Movement on plane normal to the vector is not permitted. This option does not apply to rigid bodies.

EQ.5: x-rotational degree-of-freedom,

EQ.6: y-rotational degree-of-freedom,

EQ.7: z-rotational degree-of-freedom,

EQ.8: rotational motion about a vector parallel to vector VID. Rotation about the normal axes is permitted.

EQ.-8: rotational motion about a vector parallel to vector VID. Rotation about the normal axes is not permitted. This

VARIABLE	DESCRIPTION
	option does not apply to rigid bodies.
	EQ.9: y/z degrees-of-freedom for node rotating about the x-axis at location (OFFSET1, OFFSET2) in the yz-plane, point (y, z). Radial motion is NOT permitted. Not applicable to rigid bodies.
	EQ.-9: y/z degrees-of-freedom for node rotating about the x-axis at location (OFFSET1, OFFSET2) in the yz-plane, point (y, z). Radial motion is permitted. Not applicable to rigid bodies.
	EQ.10: z/x degrees-of-freedom for node rotating about the y-axis at location (OFFSET1, OFFSET2) in the zx-plane, point (z, x). Radial motion is NOT permitted. Not applicable to rigid bodies.
	EQ.-10: z/x degrees-of-freedom for node rotating about the y-axis at location (OFFSET1, OFFSET2) in the zx-plane, point (z, x). Radial motion is permitted. Not applicable to rigid bodies.
	EQ.11: x/y degrees-of-freedom for node rotating about the z-axis at location (OFFSET1, OFFSET2) in the xy-plane, point (x, y). Radial motion is NOT permitted. Not applicable to rigid bodies.
	EQ.-11: x/y degrees-of-freedom for node rotating about the z-axis at location (OFFSET1, OFFSET2) in the xy-plane, point (x, y). Radial motion is permitted. Not applicable to rigid bodies.
	EQ.12: translational motion in direction given by the normals to the segments defined by the set typeID.
VAD	Velocity/Acceleration/Displacement flag: EQ.0: velocity (rigid bodies and nodes), EQ.1: acceleration (rigid bodies and nodes), EQ.2: displacement (rigid bodies and nodes). EQ.3: velocity versus displacement (rigid bodies only) EQ.4: relative displacement (rigid bodies only)

VARIABLE DESCRIPTION

LCID

Curve ID or function ID to describe motion value versus time, see *DEFINE_CURVE, *DEFINE_CURVE_FUNCTION, or *DEFINE_FUNCTION. If LCID refers to *DEFINE_FUNCTION, the function can have only time as an argument, e.g., $f(t) = 10.0 \times t$. See BIRTH below.

SF

Load curve scale factor. (default = 1.0)

VID

Vector ID for DOF values of 4 or 8, see *DEFINE_VECTOR. The direction of this vector is not updated with time.

DEATH

Time imposed motion/constraint is removed:

EQ.0.0: default set to 10^{28}

BIRTH

Time that the imposed motion/constraint is activated. The prescribed motion begins acting at time = BIRTH but from the zero abscissa value of the curve or function (*DEFINE_FUNCTION). In other words, the abscissae are shifted by an amount BIRTH, i.e., it has the same effect as setting OFFA = BIRTH in *DEFINE_CURVE. Warning: BIRTH is ignored if the LCID is defined as a function, i.e., *DEFINE_CURVE_FUNCTION.

BOXID

A box ID defining a box region in space in which the constraint is activated. Only the nodes falling inside the box will be applied the prescribed motion. If LCBCHK is not defined, the box volume is reevaluated every time step to determine the nodes for which the prescribed motion is active. This reevaluation of the volume is referred to as a "box-check".

LCBCHK

Optional load curve allowing more flexible and efficient use of SET_BOX option. Instead of performing box-check at every time step, discrete box-check times could be given as x-values of LCBCHK. LCBCHK's y-values specify corresponding death times. For example, a curve with points (20, 30) and (50, 70) will result in two box checks. The first will occur at 20, and the prescribed motion will be active from 20 to 30. The second will occur at 50, and the prescribed motion will be active from 50 to 70. A y-value of "0" means the prescribed motion will stay active until next box-check. For example, an additional 3rd point of (90, 0) will lead to another box-check at 90, and the prescribed motion will be active from 90 until the end of the simulation.

VARIABLE	DESCRIPTION
TOFFSET	Time offset flag for the SET_BOX option: EQ.1: the time value of the load curve, LCID, will be offset by the time when the node enters the box, EQ.0: no time offset is applied to LCID
OFFSET1	Offset for DOF types 9-11 (y, z, x direction)
OFFSET2	Offset for DOF types 9-11 (z, x, y direction)
MRB	Master rigid body for measuring the relative displacement.
NODE1	Optional orientation node, n1, for relative displacement
NODE2	Optional orientation node, n2, for relative displacement
NBEG	Node ID of a starting node.
NEND	Node ID of an ending node. All existing nodes and new nodes generated by h-adaptive mesh refinement along the straight line connecting NBEG and NEND will be included in the prescribed boundary motions.

Remarks:

When DOF = 5, 6, 7, or 8, nodal rotational degrees-of-freedom are prescribed in the case of deformable nodes (*OPTION1* = NODE or SET) whereas body rotations are prescribed in the case of a rigid body (*OPTION1* = RIGID). In the case of a rigid body, the axis of prescribed rotation always passes through the body's center of mass. For $|\mathrm{DOF}| = 8$, the axis of the prescribed rotation is parallel to vector VID. To prescribe a body rotation of a set of deformable nodes, with the axis of rotation parallel to global axes x, y, or z, use *OPTION1* = SET with $|\mathrm{DOF}| = 9$, 10, or 11, respectively. The load curve scale factor can be used for simple modifications or unit adjustments.

The relative displacement can be measured in either of two ways:

1. Along a straight line between the mass centers of the rigid bodies,

2. Along a vector beginning at node n_1 and terminating at node n_2.

With option 1, a positive displacement will move the rigid bodies further apart, and, likewise a negative motion will move the rigid bodies closer together. The mass centers of the rigid bodies must not be coincident when this option is used. With option 2 the relative displacement is measured along the vector, and the rigid bodies may be coincident. Note that the motion of the master rigid body is not directly affected by this option, i.e., no forces are generated on the master rigid body.

The activation time, BIRTH, is the time during the solution that the constraint begins to act. Until this time, the prescribed motion card is ignored. The function value of the load curves will be evaluated at the offset time given by the difference of the solution time and BIRTH, i.e., (solution time-BIRTH). Relative displacements that occur prior to reaching BIRTH are ignored. Only relative displacements that occur after BIRTH are prescribed.

When the constrained node is on a rigid body, the translational motion is imposed without altering the angular velocity of the rigid body by calculating the appropriate translational velocity for the center of mass of the rigid body using the equation:

$$\mathbf{v}_{\mathrm{cm}} = \mathbf{v}_{\mathrm{node}} - \boldsymbol{\omega} \times (\mathbf{x}_{\mathrm{cm}} - \mathbf{x}_{\mathrm{node}})$$

where $\mathbf{v}_{cm}$ is the velocity of the center of mass, $\mathbf{v}_{\mathrm{node}}$ is the specified nodal velocity, $\boldsymbol{\omega}$ is the angular velocity of the rigid body, $\mathbf{x}_{\mathrm{cm}}$ is the current coordinate of the mass center, and $\mathbf{x}_{\mathrm{node}}$ is the current coordinate of the nodal point. Extreme care must be used when prescribing motion of a rigid body node. Typically, for nodes on a given rigid body, the motion of no more than one node should be prescribed or unexpected results may be obtained.

When the RIGID option is used to prescribe rotation of a rigid body, the axis of rotation will always be shifted such that it passes through the center-of-mass of the rigid body. By using *PART_INERTIA or *CONSTRAINED_NODAL_RIGID_BODY_INERTIA, one can override the internally-calculated location of the center-of-mass.

When the RIGID_LOCAL option is invoked, the orientation of the local coordinate system rotates with time in accordance with rotation of the rigid body.

Angular displacements are applied in an incremental fashion hence it is not possible to correctly prescribe a successive set of rotations about multiple axes. In light of this the command *BOUNDARY_PRESCRIBED_ORIENTATION_RIGID should be used for the purpose of prescribing the general orientation of a rigid body.

Example:

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$   *BOUNDARY_PRESCRIBED_MOTION_SET
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  A set of nodes is given a prescribed translational velocity in the
$  x-direction according to a specified vel-time curve (which is scaled).
$
*BOUNDARY_PRESCRIBED_MOTION_SET
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
```

```
$     nsid       dof       vad      lcid        sf       vid     death
         4         1         0         8       2.0
$
$     nsid = 4    nodal set ID number, requires a *SET_NODE_option
$      dof = 1    motion is in x-translation
$      vad = 0    motion prescribed is velocity
$     lcid = 8    velocity follows load curve 8, requires a *DEFINE_CURVE
$       sf = 2.0  velocity specified by load curve is scaled by 2.0
$      vid        not used in this example
$    death        use default (essentially no death time for the motion)
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$

$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$   *BOUNDARY_PRESCRIBED_MOTION_RIGID
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  A rigid body is given a prescribed rotational displacement about the
$  z-axis according to a specified displacement-time curve.
$
*BOUNDARY_PRESCRIBED_MOTION_RIGID
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      pid       dof       vad      lcid        sf       vid     death
        84         7         2         9                          14.0
$
$      pid = 84   apply motion to part number 84
$      dof = 7    rotation is prescribed about the z-axis
$      vad = 2    the prescribed motion is displacement (angular)
$     lcid = 9    rotation follows load curve 9, requires a *DEFINE_CURVE
$                     (rotation should be in radians)
$       sf        use default (sf = 1.0)
$      vid        not used in this example
$    death = 14   prescribed motion is removed at 14 ms (assuming time is in ms)
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
```

SET_LINE option:

Referring to Figure 5-4 and a partial keyword example below, a flat plate is being pulled along one edge while the opposite edge is fully constrained. All four existing nodes and new nodes created from *h*-adaptive mesh refinement along the straight line connecting nodes 98 and 105 will be included in a node set ID 122, which is subjected to a velocity boundary condition defined by load curve ID 2. From the deformed shape, it is evident all nodes are pulled equally according to the boundary condition.

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
*BOUNDARY_PRESCRIBED_MOTION_SET_LINE
$#    nsid       dof       vad      lcid        sf       vid     death     birth
       122         3         0         2
$     NBEG      NEND
        98       105
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
*DEFINE_CURVE
2
```

```
          0.0             &velo
     &endtime             &velo
       1000.0             &velo
```

Revision Information:

SET_LINE option is available starting in Revision 109996 for both SMP and MPP.

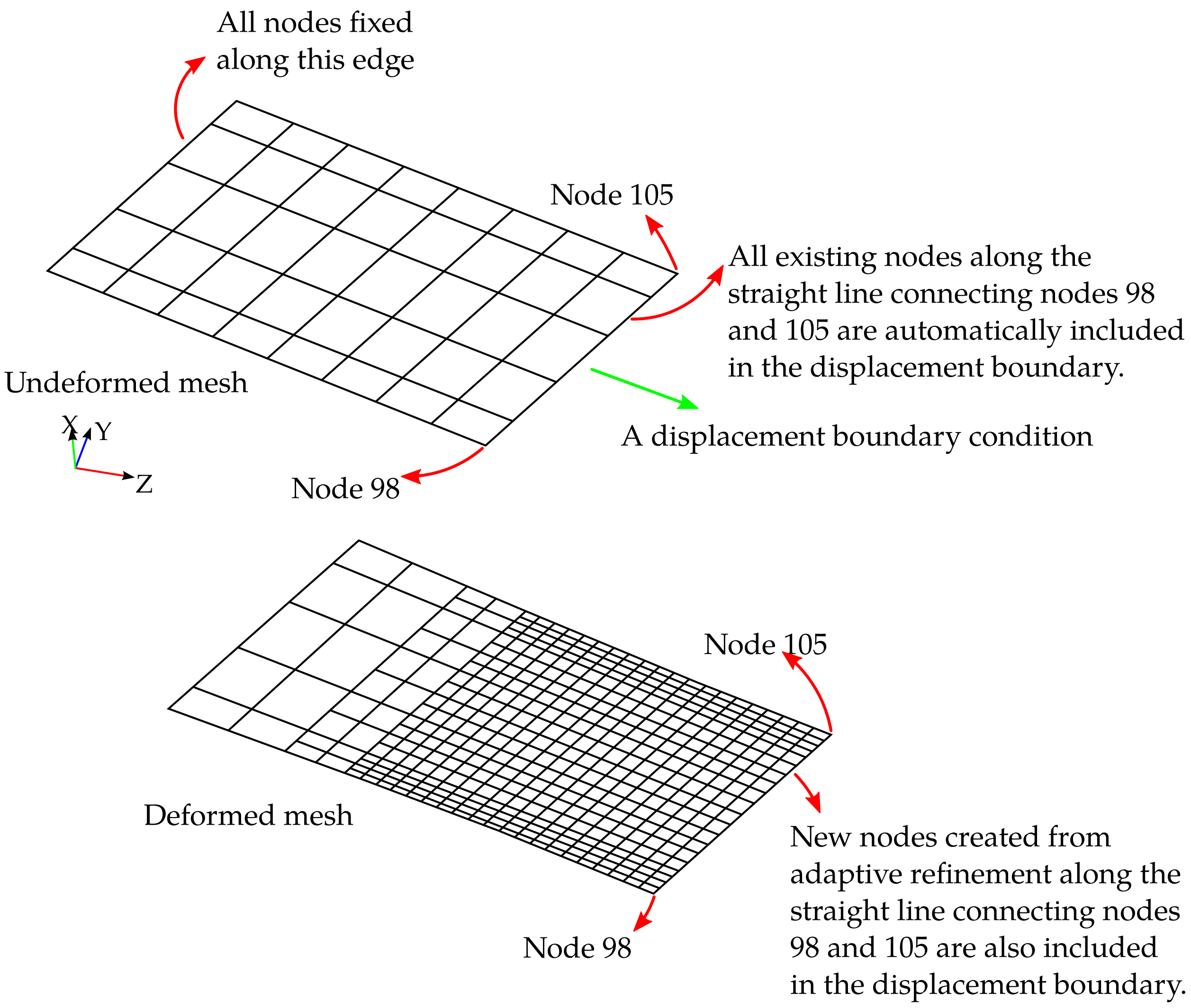

All nodes fixed
along this edge
Node 105
All existing nodes along the straight line connecting nodes 98 and 105 are automatically included in the displacement boundary.
Undeformed mesh
X
Y
Z
A displacement boundary condition
Node 98
Node 105
Deformed mesh
New nodes created from adaptive refinement along the straight line connecting nodes 98 and 105 are also included in the displacement boundary.
Node 98

Figure 5-4. The SET_LINE option usage.

*BOUNDARY_PRESCRIBED_ORIENTATION_RIGID_*OPTION1*_*{OPTION2}*

Available options *OPTION1* include:

DIRCOS

ANGLES

EULERP

VECTOR

OPTION2 allows an optional ID:

ID

The defined ID can be referred to by *SENSOR_CONTROL.

Purpose: Prescribe the orientation of rigid body as a function of time.

Card Formats:

ID Card. Optional card for ID keyword option.

ID	1	2	3	4	5	6	7	8
Variable	ID	HEADING						
Type	I	A70						

Card 1 is common to all orientation methods.
Cards 2 to 3 are unique for each orientation method.

Card 1	1	2	3	4	5	6	7	8
Variable	PIDB	PIDA	INTRP	BIRTH	DEATH	TOFFSET		
Type	I	I	I	F	F	I		
Default	none	0	1	0.	10^{20}	0		

VARIABLE	DESCRIPTION
ID	Optional ID for PRESCRIBED ORIENTATION that can be referred to by *SENSOR_CONTROL. When not defined, the sequential definition order will be used as ID when referred to by *SENSOR_CONTROL.
HEADING	An optional descriptor for the given ID.
PIDA	Part ID for rigid body A. If zero then orientation of PIDB is performed with respect to the global reference frame.
INTRP	Interpolation method used on time history curves: EQ.1: linear interpolation (default) EQ.2: cubic spline interpolation (experimental – under development)
BIRTH	Prior to this time the body moves freely under the action of other agents.
DEATH	The body is freed at this time and subsequently allowed to move under the action of other agents.
TOFFSET	Time offset flag: EQ.1: The time value of all load curves will be offset by the birth time. EQ.0: No time offset is applied.

Cosine Card 1. Additional card for DIRCOS option.

Card 2	1	2	3	4	5	6	7	8
Variable	LCIDC11	LCIDC12	LCIDC13	LCIDC21	LCIDC22	LCIDC23	LCIDC31	LCIDC32
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

Cosine Card 2. Additional card for DIRCOS option.

Card 3	1	2	3	4	5	6	7	8
Variable	LCIDC33							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
LCIDC*ij*	Load curve ID specifying direction cosine C_{ij} as a function of time. C_{ij} is defined as: $$C_{ij} = \mathbf{a}_i \cdot \mathbf{b}_j$$ where the $\{\boldsymbol{a}_i\}$ are mutually perpendicular unit vectors fixed in PIDA and the $\{\boldsymbol{b}_j\}$ are mutually perpendicular unit vectors fixed in PIDB. If PIDA = 0 then the $\{\boldsymbol{a}_i\}$ are unit vectors aligned with the global x, y, and z. See Remark 1.

Angles Card. Additional card for ANGLES option.

Card 2	1	2	3	4	5	6	7	8
Variable	LCIDQ1	LCIDQ2	LCIDQ3	ISEQ	ISHFT	BODY		
Type	I	I	I	I	I	I		
Default	none	none	none	none	1	0		

VARIABLE	DESCRIPTION
LCIDQ*i*	Load curve ID specifying the orientation angle q_i in radians as a function of time. See Remark 1.
ISEQ	Specifies the sequence in which the rotations are performed. In this first set of sequences three unique axes are involved. This sequence is associated with what are commonly called Cardan or Tait-Bryan angles. *All angles are in units of radians.* Whether these rotations are intrinsic or extrinsic is determined by the BODY field.

EQ.123: The first rotation is performed about the x axis through an angle of q_1, the second about the y axis through an angle of q_2, and the third about the z axis through an angle of q_3.

EQ.231: The first rotation is performed about the y axis through an angle of q_1, the second about the z axis through an angle of q_2, and the third about the x axis through an angle of q_3.

EQ.312: The first rotation is performed about the z axis through an angle of q_1, the second about the x axis through an angle of q_2, and the third about the y axis through an angle of q_3.

EQ.132: The first rotation is performed about the x axis through an angle of q_1, the second about the z axis through an angle of q_2, and the third about the y axis through an angle of q_3.

EQ.213: The first rotation is performed about the y axis through an angle of q_1, the second about the x axis through an angle of q_2, and the third about the z axis through an angle of q_3.

EQ.321: The first rotation is performed about the z axis through an angle of q_1, the second about the y axis through an angle of q_2, and the third about the z axis through an angle of q_3.

The second set of sequences involve only two unique axes where the first and third are repeated. This sequence is associated with what are commonly called Euler angles.

EQ.121: the first rotation is performed about the x axis through an angle of q_1, the second about the y axis through an angle of q_2, and the third about the x axis through an angle of q_3.

EQ.131: The first rotation is performed about the x axis through an angle of q_1, the second about the z axis through an angle of q_2, and the third about the x axis through an angle of q_3.

EQ.212: The first rotation is performed about the y axis through an angle of q_1, the second about the x axis through an angle of q_2, and the third about the y axis through an angle of q_3.

EQ.232: The first rotation is performed about the y axis through an angle of q_1, the second about the z axis through an angle of q_2, and the third about the y axis through an angle of q_3.

EQ.313: The first rotation is performed about the z axis through an angle of q_1, the second about the x axis through an angle of q_2, and the third about the z axis through an angle of q_3.

EQ.323: The first rotation is performed about the z axis through an angle of q_1, the second about the y axis through an angle of q_2, and the third about the z axis through an angle of q_3.

ISHFT Angle shift.

EQ.1: Angle curves are unaltered.

EQ.2: Shifts angle data in the LCIDQ*i* curves as necessary to eliminate discontinuities. If angles are confined to the range $[-\pi, \pi]$ and the data contains excursions exceeding π then set ISHFT = 2.

BODY Reference axes.

EQ.0: Rotations are performed about axes fixed in PIDA (extrinsic rotation, default).

EQ.1: Rotations are performed about axes fixed in PIDB (intrinsic rotation).

Euler Parameter Card. Additional card for EULERP option.

Card 2	1	2	3	4	5	6	7	8
Variable	LCIDE1	LCIDE2	LCIDE3	LCIDE4				
Type	I	I	I	I				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
LCIDE*i*	Load curve ID specifying Euler parameter e_i as a function of time. The Euler parameters are defined as follows. See Remark 1. $\varepsilon_i = \boldsymbol{\varepsilon} \cdot \boldsymbol{a}_i = \boldsymbol{\varepsilon} \cdot \boldsymbol{b}_i, \qquad (i = 1,2,3)$ $\varepsilon_4 = \cos\left(\frac{\theta}{2}\right)$ where $\boldsymbol{\varepsilon}$ is the Euler vector, $\{\boldsymbol{a}_i\}$ and $\{\boldsymbol{b}_i\}$ are dextral sets of unit vectors fixed in PIDA and PIDB, respectively, and θ (in radians) is associated with the rotation of PIDB in PIDA about Euler vector. If PIDA = 0 then the $\{\boldsymbol{a}_i\}$ are unit vectors aligned, respectively, with the global x, y, and z axes.

The Euler parameters are defined as $\boldsymbol{\varepsilon} = \sin\left(\frac{\theta}{2}\right)\boldsymbol{n}$ and $\varepsilon_4 = \cos\left(\frac{\theta}{2}\right)$, respectively. Here $\boldsymbol{n}$ is a unit vector defining the axis of rotation, and θ is the angle with which the rotation occurs, and consequently the four parameters are subjected to the condition $\boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon} + \varepsilon_4^2 = 1$. It is therefore recommended that the control points of the curves already fulfil this or else LS-DYNA will internally normalize these values. From the Euler parameters at time t, a unique rotation matrix $\boldsymbol{Q}_t$ is computed that is used to determine the total orientation $\boldsymbol{Q}$. The rotation is performed with respect to the *reference state* $\boldsymbol{Q}_0$ given by the Euler parameters at time 0. In general, $\boldsymbol{Q}_0 \neq \boldsymbol{I}$ and the rotation of the rigid body is given by $\boldsymbol{Q} = \boldsymbol{Q}_t\boldsymbol{Q}_0^T$. If the parameters are initially $\boldsymbol{\varepsilon} = \boldsymbol{0}$ and $\varepsilon_4 = 1$, then the reference state is $\boldsymbol{Q}_0 = \boldsymbol{I}$ and $\boldsymbol{Q} = \boldsymbol{Q}_t$ defines the orientation of the rigid body.

For a nonzero PIDA, the rotation matrix $\boldsymbol{Q}$ as defined above is expressed in a system that is fixed in rigid body A. If this system is denoted $\boldsymbol{R}_t$ at time t, and assuming $\boldsymbol{R}_0 = \boldsymbol{I}$, the orientation with respect to a global system is $\boldsymbol{RQ}$.

Vector Card. Additional card for VECTOR option.

Card 2	1	2	3	4	5	6	7	8
Variable	LCIDV1	LCIDV2	LCIDV3	LCIDS	VALSPIN			
Type	I	I	I	I	F			
Default	none	none	none	0	0.			

VARIABLE **DESCRIPTION**

LCIDV*i*

Load curve ID specifying the vector measure number v_i as a function of time. The vector measure numbers are defined as follows. See Remark 1.

$$v_i = \boldsymbol{v} \cdot \boldsymbol{n}_i, \qquad i = 1, 2, 3.$$

where $\boldsymbol{v}$ is a vector and $\{\boldsymbol{n}_i\}$ are unit vectors aligned, respectively, with the global axes x, y, and z axes. Note that the vector $\boldsymbol{v}$ is attached to the body in question, so changing the direction of this vector will induce a rotation of the body defined by $\dot{\boldsymbol{\varphi}} = \boldsymbol{v} \times \dot{\boldsymbol{v}}$.

LCIDS

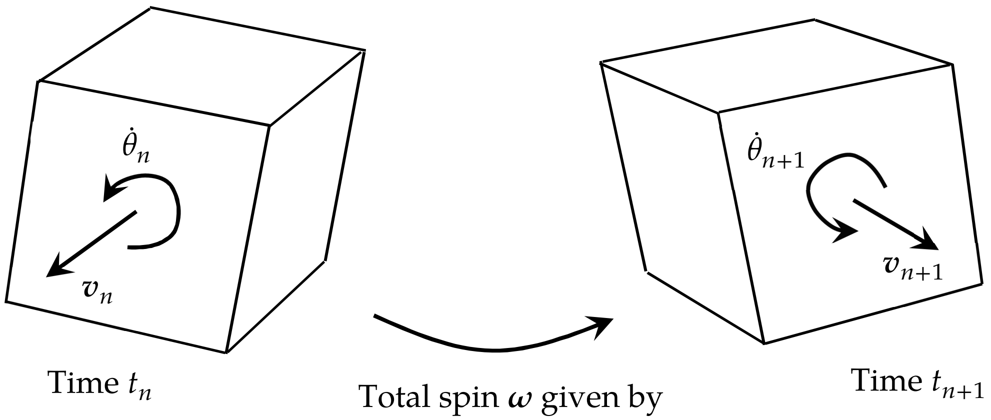
Load curve ID which specifies the overlayed spin speed $\dot{\theta}$ of PIDB about the axis parallel to the vector $\boldsymbol{v}$.

EQ.0: a constant spin speed as defined by VALSPIN is used,

GT.0: Load curve for spin speed (radians per unit time).

VALSPIN

Value for the constant spin speed of PIDB (radians per unit time $\dot{\theta}$). This option is bypassed if the load curve number defined above is non-zero.



Total spin $\boldsymbol{\omega}$ given by
$\boldsymbol{\omega} = \dot{\boldsymbol{\varphi}} + \dot{\boldsymbol{\theta}} = \boldsymbol{v} \times \dot{\boldsymbol{v}} + \dot{\theta}\boldsymbol{v}$

Remarks:

1. All load curves must contain the same number of points and the data must be uniformly spaced.

2. LC0 in *MAT_RIGID must be used to identify a coordinate system for each rigid body. The coordinate system must be defined with *DEFINE_COORDI-

NATE_NODES and FLAG = 1. Nodes used in defining the coordinate system must reside on the same body.

3. This feature is incompatible with *DEFINE_CURVE_FUNCTION.

BOUNDARY_PRESSURE_OUTFLOW_*OPTION

Available options include:

SEGMENT

SET

Purpose: Define pressure outflow boundary conditions. These boundary conditions are attached to solid elements using the Eulerian ambient formulation (refer to ELFORM in *SECTION_SOLID_ALE) and defined to be pressure outflow ambient elements (refer to AET in *SECTION_SOLID_ALE).

Card 1 for SET option.

Card 1	1	2	3	4	5	6	7	8
Variable	SSID							
Type	I							
Default	none							

Card 1 for SEGMENT option.

Card 1	1	2	3	4	5	6	7	8
Variable	N1	N2	N3	N4				
Type	I	I	I	I				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
SSID	Segment set ID
N1, N2, …	Node ID’s defining segment

BOUNDARY_PWP_*OPTION

Available options include:

NODE

SET

TABLE

TABLE_SET

Purpose: Define pressure boundary conditions for pore water, e.g. at soil surface. The TABLE option applies to a whole Part, while the other options apply to specified nodes.

Card 1	1	2	3	4	5	6	7	8
Variable	typeID	LC	CMULT	LCDR	TBIRTH	TDEATH		
Type	I	F	F	I	F	F		
Default	none	none	0.0	none	0.0	10^{20}		

Card 2	1	2	3	4	5	6	7	8
Variable	IPHRE	ITOTEX	IDRFLAG	TABLE				
Type	I	I	I	I				
Default	0	0	0	0				

VARIABLE	DESCRIPTION
typeID	Node ID (option = NODE) or Node set ID (option = SET) or Part ID (option = TABLE) or Part Set ID (option = TABLE_SET)
LC	Load curve or function giving pore water pressure head (length units) vs time. EQ.0: constant pressure head assumed equal to CMULT (leave blank for TABLE option)

VARIABLE **DESCRIPTION**

CMULT

Factor on curve or constant pressure head if LC = 0

LCDR

Load curve or function giving pore water pressure head during dynamic relaxation.

EQ.0: during dynamic relaxation, use first pressure head value on LC

(leave blank for TABLE option)

TBIRTH

Time at which boundary condition becomes active

TDEATH

Time at which boundary condition becomes inactive

IPHRE

EQ.0: default behavior

EQ.1: for phreatic behavior (water can be removed by the boundary condition but not added, e.g. at a sloping free surface). Not applicable to TABLE option. See remarks.

ITOTEX

Flag for type of pressure boundary condition: (see notes)

EQ.0: Total head

EQ.1: Excess head

EQ.2: Hydraulic head

EQ.4: *z*-coord where head = 0 (piezometric level)

IDRFLAG

Active flag:

EQ.0: Active only in transient analysis

EQ.1: Active only in dynamic relaxation

EQ.2: Active in all analysis phases

(leave blank for TABLE option)

TABLE

Table ID for TABLE option only. See notes below.

Remarks:

1. Pressure is given as pressure head, i.e. pressure/ρg.

2. NODE and SET options do not affect the pore pressure in Drained parts (the pore pressure for these is set on a part basis and overrides any nodal boundary conditions). The TABLE option should be used *only* with Drained parts.

3. *BOUNDARY_PWP_NODE or SET overrides pressure head from *BOUNDARY_PWP_TABLE at nodes where both are present.

4. 4. If LC is a *DEFINE_FUNCTION, the input arguments are (time, x, y, z, x0, y0, z0) where x, y and z are the current coordinates and x0, y0, z0 are the initial coordinates of the node.

TABLE and TABLE_SET options:

The table consists of a list of times in ascending order, followed immediately by curves of *z*-coordinate versus pore pressure head. Each curve represents the pore water pressure head distribution with *z*-coordinate at the corresponding time. There must be the same number of curves as time values, arranged immediately after the *DEFINE_-TABLE and in the correct order to correspond to the time values. Each curve should be arranged in ascending order of *z*-coordinate – they look upside-down on the page. The *z*-coordinate is the *x*-axis of the curve, the pore water pressure head (in length units) is the y-axis. Each curve should have the same *z*-coordinates (*x*-values) and use the same value of LCINT. Ensure that the range of *z*-coordinates in the curve exceeds by at least 5% the range of *z*-coordinates of the nodes belonging to the parts to which the boundary condition is applied.

IPHRE:

"Phreatic" means that water can be removed by the boundary condition but not added. The boundary condition enforces that the pressure head be less than or equal to the stated value. This condition occurs when the free surface of the soil is sloping so that any water emerging from the soil runs away down the slope.

ITOTEX = 0:

The value from curve or table is total head. This may be used with any pore pressure analysis type.

ITOTEX = 1:

The value from curve or table is excess head. Total head will be determined by adding the hydrostatic head. This option cannot be used with drained analysis, which sets excess head to zero.

ITOTEX = 2:

The value from curve or table is hydraulic head, to which excess head may be added due to volume change in the soil if the analysis type is not drained.

ITOTEX = 4:

The curve value is the z-coordinate of the water surface; pore pressure head at any node in this boundary condition is given by,

$$z_{\mathrm{surface}} - z_{\mathrm{node}}$$

This option allows a single boundary condition to be used for nodes at any depth, provided that the pressure distribution is hydrostatic below the given surface. This option is not available for the TABLE option.

*BOUNDARY_RADIATION_*OPTION1*_*{OPTION2}*_*{OPTION3}*

Available values for *OPTION1* include:

SET

SEGMENT

Available values for *OPTION2* include:

VF_READ

VF_CALCULATE

<BLANK>

Available values for *OPTION3* include:

RESTART

<BLANK>

OPTION1 specifies radiation boundary surface definition by a surface set (SET) or by a segment list (SEGMENT).

OPTION2 indicates the radiation boundary surface is part of an enclosure. When set to VF OPTION2 specifies the use of view factors. The suffix, READ, indicates that the view factors should be read from the file "viewfl". The suffix, CALCULATE, indicates that the view factors should be calculated. The Stefan Boltzmann constant must be defined for radiation in an enclosure on the *CONTROL_THERMAL_SOLVER keyword. The parameter DTVF entered on the CONTROL_THERMAL_SOLVER keyword defines the time interval between VF updates for moving geometries.

OPTION3 is the keyword suffix RESTART. This is only applicable in combination with the keyword VF_CALCULATE. In very long runs, it may be necessary to halt execution. This is accomplished by entering Ctrl-C followed by sw1. To restart the view factor calculation, add the suffix RESTART to all VF_CALCULATE keywords in the input file.

The status of an in-progress view factor calculation can be determined by using the sense switch. This is accomplished by first typing Control-C followed by:

sw1.	Stop run and save viewfl file for restart
sw2.	Viewfactor run statistics

A list of acceptable keywords are:

*BOUNDARY_RADIATION_SEGMENT

*BOUNDARY_RADIATION_SEGMENT_VF_READ

*BOUNDARY_RADIATION_SEGMENT_VF_CALCULATE

*BOUNDARY_RADIATION_SET

*BOUNDARY_RADIATION_SET_VF_READ

*BOUNDARY_RADIATION_SET_VF_CALCULATE

Remarks:

In models that include radiation boundary conditions, a thermodynamic temperature scale is required, i.e., zero degrees must correspond to absolute zero. The Kelvin and Rankine temperature scales meet this requirement whereas Celsius and Fahrenheit temperature scales do not.

*BOUNDARY_RADIATION_SEGMENT

Include the following 2 cards for each segment. Apply a radiation boundary condition on a SEGMENT to transfer heat between the segment and the environment. Setting TYPE = 1 on Card 1 below indicates that the segment transfers heat to the environment.

Card 1	1	2	3	4	5	6	7	8
Variable	N1	N2	N3	N4	TYPE			
Type	I	I	I	I	I			
Default	none	none	none	none	1			

Card 2	1	2	3	4	5	6	7	8
Variable	FLCID	FMULT	TLCID	TMULT	LOC			
Type	I	F	I	F	I			
Default	none	0.	none	0.	0			

VARIABLE	DESCRIPTION
N1, N2, N3, N4	Node ID’s defining segment
TYPE	Radiation type: EQ.1: Radiation to environment
FLCID	Radiation heat transfer coefficient, $f = \sigma\varepsilon F$, specification where σ = Stefan Boltzmann constant, ε = surface emissivity, F = surface view factor. This parameter can reference a load curve ID (see *DEFINE_CURVE) or a function ID (see *DEFINE_FUNCTION and Remark 1). When the reference is to a curve, FLCID has the following interpretation: GT.0: f is defined as a function of time, t, having points consisting of $(t,f(t))$ data pairs. EQ.0: f is a constant defined by the value FMULT.

VARIABLE	DESCRIPTION
	LT.0: f is defined as a function of temperature, T, by a curve consisting of $(T,f(T)\ )$ data pairs. Enter |FLCID| on the DEFINE_CURVE keyword.
FMULT	Radiation heat transfer coefficient, f, curve multiplier for use in the equation $\dot{q}'' = \sigma\varepsilon F(T_{\text{surface}}^4 - T_\infty^4) = f(T_{\text{surface}}^4 - T_\infty^4)$ If f is a function of temperature, f is evaluated at the surface temperature, T_{surface}. [σ = Stefan Boltzmann constant, ε = surface emissivity, F = surface view factor]
TLCID	Environment temperature, T_∞, specification. This parameter can reference a load curve ID (see *DEFINE_CURVE) or a function ID (see *DEFINE_FUNCTION and Remark 2). When the reference is to a curve, TLCID has the following interpretation: GT.0: T_∞ is defined as a function of time, t, by a curve consisting of $(t,T_\infty(t)\ )$ data pairs. EQ.0: T_∞ a constant defined by the value TMULT.
TMULT	Environment temperature, T_∞, curve multiplier.
LOC	For a thick thermal shell, the radiation will be applied to the surface identified by LOC. See the parameter THSHEL on the *CONTROL_SHELL keyword. EQ.-1: lower surface of thermal shell element EQ.0: middle surface of thermal shell element EQ.1: upper surface of thermal shell element

Remarks:

A radiation boundary condition is calculated using

$$\dot{q}'' = \sigma\varepsilon F(T_{\text{surface}}^4 - T_\infty^4) = f(T_{\text{surface}}^4 - T_\infty^4)$$

Where, f , is the radiation heat transfer coefficient. If f is a function of temperature, f, is evaluated at the surface temperature, T_{surface}.

1. If HLCID references a DEFINE_FUNCTION, the following function arguments are allowed $f(x,y,z,v_x,v_y,v_z,T,T_\infty,t)$ where:

   x,y,z = segment centroid coordinates

v_x, v_y, v_z = segment centroid velocity component

T = segment centroid temperature

T_∞ = environment temperature, T_∞

t = solution time

2. If TLCID references a DEFINE_FUNCTION, the following function arguments are allowed $f(x , y , z , v_x, v_y, v_z, t)$ where:

x, y, z = segment centroid coordinates

v_x, v_y, v_z = segment centroid velocity components

t = solution time

BOUNDARY_RADIATION_SEGMENT_VF_*OPTION

Available options include:

READ

CALCULATE

Include the following 2 cards for each segment. Apply a radiation boundary condition on a SEGMENT to transfer heat between the segment and an enclosure surrounding the segment using view factors. The enclosure is defined by additional segments using this keyword. Setting TYPE = 2 on Card 1 below specifies that the segment belongs to an enclosure.

The file "viewfl" must be present for the READ option, whereas it will be created with the CALCULATE option. If the file "viewfl" exists when using the CALCULATE option, LS-DYNA will terminate with an error message to prevent overwriting the file. The file "viewfl" contains the surface-to-surface area × view factor products (i.e., A_iF_{ij}). These products are stored by row and formatted as 5E16.0.

Card 1	1	2	3	4	5	6	7	8
Variable	N1	N2	N3	N4	TYPE	BLOCK	NINT	
Type	I	I	I	I	I	I	I	
Default	none	none	none	none	2	0	0	

Card 2	1	2	3	4	5	6	7	8
Variable	SELCID	SEMULT						
Type	I	F						
Default	none	0.						

VARIABLE	DESCRIPTION
N1, N2, N3, N4	Node ID's defining segment

VARIABLE	DESCRIPTION
TYPE	Radiation type: EQ.2: Radiation within an enclosure
BLOCK	Flag indicating if this surface blocks the view between any other 2 surfaces. EQ.0: no blocking (default) EQ.1: blocking
NINT	Number of integration points for viewfactor calculation EQ.0: LS-DYNA determines the number of integration points based on the segment size and separation distance 1 ≤ NINT ≤ 10: User specified number
SELCID	Load curve ID for surface emissivity, see *DEFINE_CURVE GT.0: function versus time EQ.0: use constant multiplier value, SEMULT LT.0: function versus temperature. The value of –SELCID must be an integer, and it is interpreted as a load curve ID. See the DEFINE_CURVE keyword.
SEMULT	Curve multiplier for surface emissivity, see *DEFINE_CURVE

*BOUNDARY_RADIATION_SET

Include the following 2 cards for each set. Apply a radiation boundary condition on a SEGMENT_SET to transfer heat between the segment set and the environment Setting TYPE = 1 on Card 1 below indicates that the segment transfers energy to the environment.

Card 1	1	2	3	4	5	6	7	8
Variable	SSID	TYPE						
Type	I	I						
Default	none	1						

Card 2	1	2	3	4	5	6	7	8
Variable	FLCID	FMULT	TLCID	TMULT	LOC			
Type	I	F	I	F	I			
Default	none	0.	none	0.	0			

VARIABLE	DESCRIPTION
SSID	SSID specifies the ID for a set of segments that comprise a portion of, or possibly, the entire enclosure. See *SET_SEGMENT.
TYPE	Radiation type: EQ.1: Radiation to environment
FLCID	Radiation heat transfer coefficient, $f = \sigma\varepsilon F$, specification where σ = Stefan Boltzmann constant, ε = surface emissivity, F = surface view factor. This parameter can reference a load curve ID (see *DEFINE_CURVE) or a function ID (see *DEFINE_FUNCTION and Remark 1). When the reference is to a curve, FLCID has the following interpretation: GT.0: f is defined as a function time, t, by a curve consisting of $(t,f(t))$ data pairs.

VARIABLE DESCRIPTION

EQ.0: f is a constant defined by the value FMULT.

LT.0: f is defined as a function of temperature, T, by a curve consisting of $(T, f(T))$ data pairs. Enter |-FLCID| on the DEFINE_CURVE keyword

FMULT

Curve multiplier for f for use in the equation

$$\dot{q}'' = \sigma \varepsilon F (T_{\text{surface}}^4 - T_\infty^4) = f (T_{\text{surface}}^4 - T_\infty^4)$$

If f is a function of temperature, f is evaluated at the surface temperature, T_{surface}. [σ = Stefan Boltzmann constant, ε = surface emissivity, F = surface view factor]

TLCID

Environment temperature, T_∞ , specification. This parameter can reference a load curve ID (see *DEFINE_CURVE) or a function ID (see *DEFINE_FUNCTION and Remark 2). When the reference is to a curve, TLCID has the following interpretation:

GT.0: T_∞ is defined as a function of time, t, by a curve consisting of $(t, T_\infty(t))$ data pairs.

EQ.0: T_∞ a constant defined by the value TMULT

TMULT

Curve multiplier for T_∞

LOC

For a thick thermal shell, the radiation will be applied to the surface identified by LOC. See the parameter THSHEL on the *CONTROL_SHELL keyword.

EQ.-1: lower surface of thermal shell element

EQ.0.: middle surface of thermal shell element

EQ.1: upper surface of thermal shell element

Remarks:

A radiation boundary condition is calculated using

$$\dot{q}'' = \sigma \varepsilon F (T_{\text{surface}}^4 - T_\infty^4) = f (T_{\text{surface}}^4 - T_\infty^4)$$

Where, f , is the radiation heat transfer coefficient. . If f is a function of temperature, f, is evaluated at the surface temperature, T_{surface}.

1. If HLCID references a DEFINE_FUNCTION, the following function arguments are allowed $f(x, y, z, v_x, v_y, v_z, T, T_\infty, t)$ where:

x, y, z = segment centroid coordinates

v_x, v_y, v_z = segment centroid velocity component

T = segment centroid temperature

T_∞ = environment temperature, T_∞

t = solution time

2. If TLCID references a DEFINE_FUNCTION, the following function arguments are allowed $f(x, y, z, v_x, v_y, v_z, t)$ where:

x, y, z = segment centroid coordinates

v_x, v_y, v_z = segment centroid velocity components

t = solution time

BOUNDARY_RADIATION_SET_VF_*OPTION

Available options include:

READ

CALCULATE

Include the following 2 cards for each set. Apply a radiation boundary condition on a SEGMENT_SET to transfer heat between the segment set and an enclosure surrounding the segments using view factors. Segments contained in the SEGMENT_SET may form the enclosure. Setting TYPE = 2 on Card 1 below specifies that the segment set belongs to an enclosure.

The file "viewfl" must be present for the READ option. The file "viewfl" will be created for the CALCULATE option. If the file "viewfl" exists when using the CACULATE option, LS-DYNA will terminate with an error message to prevent overwriting the file. The file "viewfl" contains the surface-to-surface area × view factor products (i.e. $A_i F_{ij}$). These products are stored by row and formatted as 5E16.0.

Card 1	1	2	3	4	5	6	7	8
Variable	SSID	TYPE	RAD_GRP	FILE_NO	BLOCK	NINT		
Type	I	I	I	I	I	I		
Default	none	2	0	0	0	0		

Card 2	1	2	3	4	5	6	7	8
Variable	SELCID	SEMULT						
Type	I	F						
Default	none	0.						

VARIABLE	DESCRIPTION
SSID	SSID specifies the ID for a set of segments that comprise a portion of, or possibly, the entire enclosure. See *SET_SEGMENT.

VARIABLE DESCRIPTION

TYPE

Radiation type:

EQ.2: Radiation within an enclosure

RAD_GRP

Radiation enclosure group ID. The segment sets from all radiation enclosure definitions with the same group ID are augmented to form a single enclosure definition. If RAD_GRP is not specified or set to zero, then the segments are placed in group zero. All segments defined by the SEGMENT option are placed in set zero.

FILE_NO

File number for view factor file. FILE_NO is added to "viewfl_" to form the name of the file containing the view factors. For example if FILE_NO is specified as 22, then the view factors are read from viewfl_22. For radiation enclosure group zero FILE_-NO is ignored and view factors are read from viewfl. The same file may be used for different radiation enclosure group definitions.

BLOCK

Flag indicating if this surface blocks the view between any other 2 surfaces.

EQ.0: no blocking (default)

EQ.1: blocking

NINT

Number of integration points for viewfactor calculation

EQ.0: LS-DYNA determines the number of integration points based on the segment size and separation distance

GE.11: Not allowed

SELCID

Load curve ID for surface emissivity, see *DEFINE_CURVE

GT.0: function versus time

EQ.0: use constant multiplier value, SEMULT

LT.0: function versus temperature. Enter –SELCID as |-SELCID| on the DEFINE_CURVE keyword.

SEMULT

Curve multiplier for surface emissivity, see *DEFINE_CURVE

Remarks:

Multiple enclosures can be modeled when using view factors. Consider the following example input. The order of segments in the view factor file follows the order the sets are assigned to the boundary radiation definition.

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$  *BOUNDARY_RADIATION_SET
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Make boundary enclosure radiation groups 8 and 9.
$
*BOUNDARY_RADIATION_SET_VF_READ
*      SSID      TYPE   RAD_GRP   FILE_NO
        15         2         9        10
       1.0       1.0
*BOUNDARY_RADIATION_SET_VF_READ
*      SSID      TYPE   RAD_GRP   FILE_NO
        12         2         9        10
       1.0       1.0
*BOUNDARY_RADIATION_SET_VF_READ
*      SSID      TYPE   RAD_GRP   FILE_NO
        13         2         8        21
       1.0       1.0
$
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

Enclosure radiation group 9 is composed of all the segments in segment set 15 followed by those in segment set 12. The view factors are stored in the file viewfl_10. Enclosure radiation group 8 is composed of the segments in segment set 13. The view factors are stored in the file viewfl_21.

*BOUNDARY_SLIDING_PLANE

Purpose: Define a sliding symmetry plane. This option applies to continuum domains modeled with solid elements.

Card 1	1	2	3	4	5	6	7	8
Variable	NSID	VX	VY	VZ	COPT			
Type	I	F	F	F	I			
Default	none	0	0	0	0			

VARIABLE	DESCRIPTION
NSID	Nodal set ID, see *SET_NODE
VX	x-component of vector defining normal or vector
VY	y-component of vector defining normal or vector
VZ	z-component of vector defining normal or vector
COPT	Option: EQ.0: node moves on normal plane, EQ.1: node moves only in vector direction.

Remarks:

Any node may be constrained to move on an arbitrarily oriented plane or line depending on the choice of COPT. Each boundary condition card defines a vector originating at (0, 0, 0) and terminating at the coordinates defined above. Since an arbitrary magnitude is assumed for this vector, the specified coordinates are non-unique and define only a direction. Use of *BOUNDARY_SPC is preferred over *BOUNDARY_SLIDING_PLANE as the boundary conditions imposed via the latter have been seen to break down somewhat in lengthy simulations owing to numerical roundoff.

*BOUNDARY_SPC_*OPTION1*_{*OPTION2*}_{*OPTION3*}

OPTION1 is required since it specifies whether the SPC applies to a single node or to a set. The two choices are:

NODE

SET

OPTION2 allows optional birth and death times to be assigned the single node or node set:

BIRTH_DEATH

This option requires one additional line of input. The BIRTH_DEATH option is inactive during the dynamic relaxation phase, which allows the SPC to be removed during the subsequent normal analysis phase. The BIRTH_DEATH option can be used only once for any given node and if used, no other *BOUNDARY_SPC commands can be used for that node.

OPTION3 allows an optional ID to be given that applies either to the single node or to the entire set:

ID

If a heading is defined with the ID, then the ID with the heading will be written at the beginning of the ASCII file, spcforc.

Purpose: Define nodal single point constraints. Do not use this option in r-adaptive problems since the nodal point ID's change during the adaptive step. If possible use CONSTRAINED_GLOBAL instead.

ID Card. Additional card for the ID keyword option.

Optional	1	2	3	4	5	6	7	8
Variable	ID	HEADING						
Type	I	A70						

Card 1	1	2	3	4	5	6	7	8
Variable	NID/NSID	CID	DOFX	DOFY	DOFZ	DOFRX	DOFRY	DOFRZ
Type	I	I	I	I	I	I	I	I
Default	none	0	0	0	0	0	0	0

Birth/Death Card. Additional card for the BIRTH_DEATH keyword option.

Card 2	1	2	3	4	5	6	7	8
Variable	BIRTH	DEATH						
Type	F	F						
Default	0.0	10^{20}						

VARIABLE	DESCRIPTION
ID	Optional SPC set ID to which this node or node set belongs. This ID does not need to be unique
HEADING	An optional SPC descriptor that will be written into the d3hsp file and the spcforc file.
NID/NSID	Node ID or nodal set ID, see *SET_NODE.
CID	Coordinate system ID, see *DEFINE_COORDINATE_SYSTEM.
DOFX	Insert 1 for translational constraint in local x-direction.
DOFY	Insert 1 for translational constraint in local y-direction.
DOFZ	Insert 1 for translational constraint in local z-direction.
DOFRX	Insert 1 for rotational constraint about local x-axis.
DOFRY	Insert 1 for rotational constraint about local y-axis.
DOFRZ	Insert 1 for rotational constraint about local z-axis.

VARIABLE	DESCRIPTION
BIRTH	Activation time for SPC constraint. The birth time is ignored during dynamic relaxation.
DEATH	Deactivation time for the SPC constraint. The death time is ignored during dynamic relaxation.

Remarks:

Constraints are applied if for each DOF*ij* field set to 1. A value of zero means no constraint. No attempt should be made to apply SPCs to nodes belonging to rigid bodies (see *MAT_RIGID for application of rigid body constraints).

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$   *BOUNDARY_SPC_NODE
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Make boundary constraints for nodes 6 and 542.
$
*BOUNDARY_SPC_NODE
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      nid       cid      dofx      dofy      dofz     dofrx     dofry     dofrz
         6         0         1         1         1         1         1         1
       542         0         0         1         0         1         0         1
$
$      Node 6 is fixed in all six degrees of freedom (no motion allowed).
$
$      Node 542 has a symmetry condition constraint in the x-z plane,
$        no motion allowed for y translation, and x & z rotation.
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

BOUNDARY_SPC_SYMMETRY_PLANE_*{OPTION}

This keyword is developed to create nodal symmetric constraints by defining a symmetric plane.

Available options include:

<BLANK>

SET

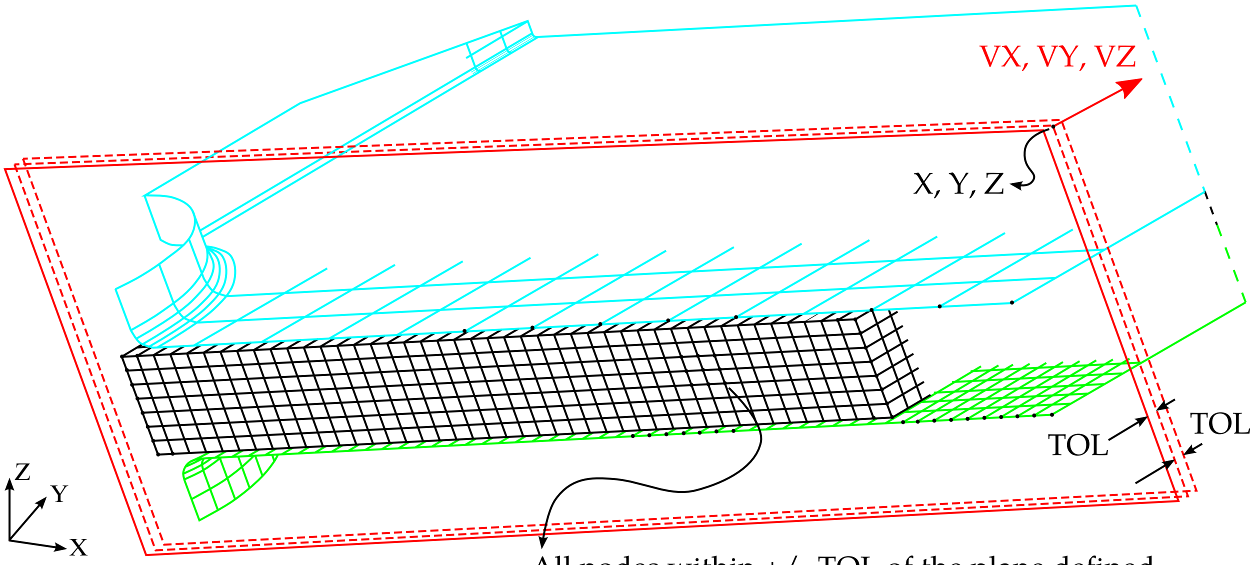
The option SET allows for symmetric boundary conditions to be applied on tailor-welded blanks (TWB).

Card Sets. For each symmetry plane input one pair of cards 1 and 2. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	IDSP	PID/PSID	X	Y	Z	VX	VY	VZ
Type	I	I	F	F	F	F	F	F
Default	none	none	0.0	0.0	0.0	0.0	0.0	0.0

Card 2	1	2	3	4	5	6	7	8
Variable	TOL							
Type	F							
Default	0.0							

VARIABLE	DESCRIPTION
IDSP	Identification number of the constraint. Must be unique.
PID/PSID	A part ID of the deformable part (sheet metal blank, for example) on which the constraints will be imposed. When the option SET is invoked, a part set ID can be input.



All nodes within +/- TOL of the plane defined by the point with coordinate (X, Y, Z) and the plane with normal vector (VX, VY, VZ) will be constrained symmetrically.

Figure 5-5. Define symmetry constraints using the variables in the keyword.

VARIABLE	DESCRIPTION
X, Y, Z	Position coordinates on the symmetry plane.
VX, VY, VZ	Vector components of the symmetry plane normal.
TOL	A distance tolerance value within which the nodes on the deformable part will be constrained.

Remarks:

1. Adaptive refined nodes generated along the symmetry plane during simulation are automatically included in the constraints.

2. Figure 5-5 shows an example of applying symmetry constraints using the variables in the keyword.

3. The following keyword creates symmetric constraints on nodes (from PID 11) within distance of 0.1mm from the defined symmetry plane (with normal vectors [1.0,1.0,1.0]) that goes through point coordinates (10.5, 40.0, 20.0):

```
*BOUNDARY_SPC_SYMMETRY_PLANE
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$     IDSP       PID         X         Y         Z        VX        VY        VZ
         1        11      10.5      40.0      20.0       1.0       1.0       1.0
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      TOL
```

4. The following keywords create two symmetric constraints on nodes from part set ID 99 (which includes part IDs 13 and 14) within distance of 0.1mm from two defined symmetry planes (with normal vectors [1.0, 0.0, 0.0] and [0.0, 1.0, 0.0], respectively) that all go through point coordinates (-76.0, 35.6, 0.0). ***Note*** the two point coordinates that define the two symmetry planes must be exactly the same.

```
*SET_PART_LIST
99
13,14
*BOUNDARY_SPC_SYMMETRY_PLANE_SET
$     IDSP       PID         X         Y         Z        VX        VY        VZ
         1        99     -76.0      35.6       0.0       1.0
$      TOL
      0.10
         2        99     -76.0      35.6       0.0       0.0       1.0
$      TOL
      0.10
```

Revision information:

This feature is available starting in Revision 85404. The option SET is available starting in Revision 113355.

*BOUNDARY_SPH_FLOW

Purpose: Define a flow of particles. This option applies to continuum domains modeled with SPH elements.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	STYP	DOF	VAD	LCID	SF	DEATH	BIRTH
Type	I	I	I	I	I	F	F	F
Default	none	none	none	0	none	1.	1.E+20	0.0

Card 2	1	2	3	4	5	6	7	8
Variable	NODE	VID						
Type	I	I						
Default	none	0						

VARIABLE	DESCRIPTION
NSID, PID	Nodal set ID (NSID), SEE *SET_NODE, or part ID (PID), see *PART.
STYP	Set type: EQ.1: part set ID, see *SET_PART, EQ.2: part ID, see *PART, EQ.3: node set ID, see *NODE_SET,

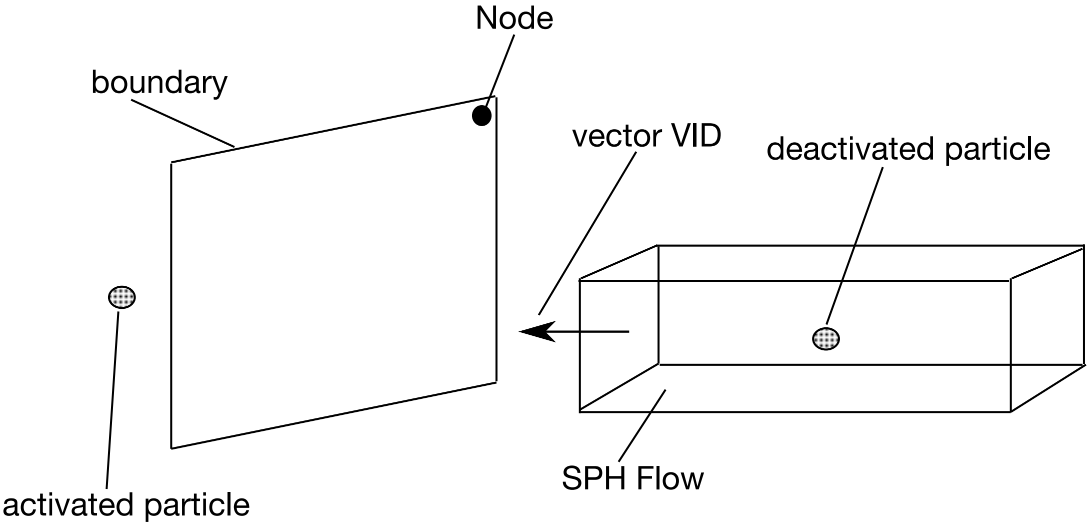


Figure 5-6. Vector **VID** determines the orientation of the SPH flow

VARIABLE **DESCRIPTION**

DOF

Applicable degrees-of-freedom:

EQ.1: x-translational degree-of-freedom,

EQ.2: y-translational degree-of-freedom,

EQ.3: z-translational degree-of-freedom,

EQ.4: translational motion in direction given by the VID. Movement on plane normal to the vector is permitted.

VAD

Velocity/Acceleration/Displacement flag applied to SPH elements before activation:

EQ.0: velocity,

EQ.1: acceleration,

EQ.2: displacement.

LCID

Load curve ID to describe motion value versus time, see *DEFINE_CURVE.

SF

Load curve scale factor. (default = 1.0)

DEATH

Time imposed motion/constraint is removed:

EQ.0.0: default set to 1020.

BIRTH

Time imposed motion/constraint is activated.

VARIABLE	DESCRIPTION
NODE	Node fixed in space which determines the boundary between activated particles and deactivated particles.
VID	Vector ID for DOF value of 4, see *DEFINE_VECTOR

Remarks:

Initially, the user defines the set of particles that are representing the flow of particles during the simulation. At time t = 0, all the particles are deactivated which means that no particle approximation is calculated. The boundary of activation is a plane determined by the NODE, and normal to the vector VID. The particles are activated when they reached the boundary. Since they are activated, particle approximation is started.

*BOUNDARY_SPH_NON_REFLECTING

Purpose: Define a non-reflecting boundary plane for SPH. This option applies to continuum domains modeled with SPH elements.

Card 1	1	2	3	4	5	6	7	8
Variable	VTX	VTY	VTZ	VHX	VHY	VHZ		
Type	F	F	F	F	F	F		
Default	0.	0.	0.	0.	0.	0.		

VARIABLE	DESCRIPTION
VTX	x-coordinate of tail of a normal vector originating on the wall (tail) and terminating in the body (head) (i.e., vector points from the non-reflecting boundary plane into the body).
VTY	y-coordinate of tail
VTZ	z-coordinate of tail
VHX	x-coordinate of head
VHY	y-coordinate of head
VHZ	z-coordinate of head

Remarks:

1. The non-reflecting boundary plane has to be normal to either the x, y or z direction.

*BOUNDARY_SPH_SYMMETRY_PLANE

Purpose: Define a symmetry plane for SPH. This option applies to continuum domains modeled with SPH elements.

Card 1	1	2	3	4	5	6	7	8
Variable	VTX	VTY	VTZ	VHX	VHY	VHZ		
Type	F	F	F	F	F	F		
Default	0.	0.	0.	0.	0.	0.		

VARIABLE	DESCRIPTION
VTX	x-coordinate of tail of a normal vector originating on the wall (tail) and terminating in the body (head) (i.e., vector points from the symmetry plane into the body).
VTY	y-coordinate of tail
VTZ	z-coordinate of tail
VHX	x-coordinate of head
VHY	y-coordinate of head
VHZ	z-coordinate of head

Remarks:

1. A plane of symmetry is assumed for all SPH elements defined in the model.

2. The plane of symmetry has to be normal to either the x, y or z direction.

3. For axi-symmetric SPH analysis, IDIM = -2, a plane of symmetry centered at the global origin and normal to x-direction is automatically created by LS-Dyna.

*BOUNDARY_SYMMETRY_FAILURE

Purpose: Define a symmetry plane with a failure criterion. This option applies to continuum domains modeled with solid elements.

Card 1	1	2	3	4	5	6	7	8
Variable	SSID	FS	VTX	VTY	VTZ	VHX	VHY	VHZ
Type	I	F	F	F	F	F	F	F
Default	none	0.	0.	0.	0.	0.	0.	0.

VARIABLE	DESCRIPTION
SSID	Segment set ID, see *SET_SEGMENT
FS	Tensile failure stress > 0.0. The average stress in the elements surrounding the boundary nodes in a direction perpendicular to the boundary is used.
VTX	x-coordinate of tail of a normal vector originating on the wall (tail) and terminating in the body (head) (i.e., vector points from the symmetry plane into the body).
VTY	y-coordinate of tail
VTZ	z-coordinate of tail
VHX	x-coordinate of head
VHY	y-coordinate of head
VHZ	z-coordinate of head

Remarks:

A plane of symmetry is assumed for the nodes on the boundary at the tail of the vector given above. Only the motion perpendicular to the symmetry plane is constrained. After failure the nodes are set free.

BOUNDARY_TEMPERATURE_*OPTION

Available options include:

NODE

SET

Purpose: Define temperature boundary conditions for a thermal or coupled thermal/ structural analysis.

Card 1	1	2	3	4	5	6	7	8
Variable	NID/SID	TLCID	TMULT	LOC				
Type	I	I	F	I				
Default	none	0	0.	0				

VARIABLE	DESCRIPTION
NID/SID	Node ID/Node Set ID, see *SET_NODE_*OPTION*
TLCID	Temperature, T, specification. This parameter can reference a load curve ID (see *DEFINE_CURVE) or a function ID (see *DEFINE_FUNCTION and Remark 2). When the reference is to a curve, TLCID has the following interpretation: GT.0: T is defined by a curve consisting of (t, T) data pairs. EQ.0: T is a constant defined by the value TMULT.
TMULT	Temperature, T, curve multiplier.
LOC	Application of surface for thermal shell elements, see parameter, THSHEL, in the *CONTROL_SHELL input: EQ.-1: lower surface of thermal shell element EQ.0: middle surface of thermal shell element EQ.1: upper surface of thermal shell element

Remarks:

1. This keyword can be used to apply temperature boundary conditions to SPH particles.

2. If TLCID references a DEFINE_FUNCTION, the following function arguments are allowed $f(x, y, z, v_x, v_y, v_z, t)$:

$$x, y, z = \text{node point coordinates}$$

$$v_x, v_y, v_z = \text{node point velocity components}$$

$$t = \text{solution time}$$

BOUNDARY_THERMAL_BULKFLOW_*OPTION1_OPTION2

Purpose: Used to define bulk fluid flow elements.

OPTION1 is required since it specifies whether the BULKFLOW applies to an element or set.

ELEMENT

SET

OPTION2 if used turns on the fluid upwind algorithm

UPWIND

Card 1	1	2	3	4	5	6	7	8
Variable	EID/SID	LCID	MDOT					
Type	I	I	F					
Default	none	none	none					

VARIABLE	DESCRIPTION
EID / SID	Beam element ID (EID) for ELEMENT option Beam set ID (SID) for SET option
LCID	Load Curve ID for mass flow rate versus time.
MDOT	Mass flow rate (e.g. kg/sec).

*BOUNDARY_THERMAL_BULKNODE

Purpose: Used to define thermal bulk nodes.

Card 1	1	2	3	4	5	6	7	8
Variable	NID	PID	NBNSEG	VOL	LCID	H	AEXP	BEXP
Type	I	I	I	F	I	F	F	F
Default	none	none	0	none	0	0.	0.	0.

Bulk Node Cards. Include NBNSEG cards, one for each bulk node segment.

Card 2	1	2	3	4	5	6	7	8
Variable	N1	N2	N3	N4				
Type	I	I	I	I				

VARIABLE	DESCRIPTION
NID	Bulk node number.
PID	Bulk node part ID.
VOL	Bulk node volume.
NBNSEG	Number of element surface segments that transfer heat with this bulk node.
N1, N2, N3, N4	Nodal point numbers
LCID	Load curve ID for H
H	Heat transfer coefficient
AEXP	*a* exponent
BEXP	*b* exponent

Remarks:

The heat flow between a bulk node (T_B) and a bulk node segment (T_S) is given by

$$q = h(T_B^a - T_B^a)^b$$

1. For convection, set $a = b = 1$.

2. For radiation, set $a = 4,\ b = 1$.

3. For flux, set $a = b = 0$. Mathematically, anything to the 0 power is 1. This produces the expression, $\left(T_B^0 - T_S^0\right)^0 = (1 - 1)^0 = 0^0 = 1$. However, some computer operating systems don't recognize 0^0. It is safer to set $a = b =$ very small number.

*BOUNDARY_THERMAL_WELD

Purpose: Define a moving heat source to model welding. Only applicable for a coupled thermal-structural simulations in which the weld source or work piece is moving.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	PTYP	NID	NFLAG	X0	Y0	Z0	N2ID
Type	I	I	I	I	F	F	F	I
Default	none	1	none	1	none	none	none	none

Card 2	1	2	3	4	5	6	7	8
Variable	a	b	cf	cr	LCID	Q	Ff	Fr
Type	F	F	F	F	I	F	F	F
Default	none	none	none	none	none	none	none	none

Beam Aiming Direction Card. Additional card for N2ID = -1.

Optional	1	2	3	4	5	6	7	8
Variable	TX	TY	TZ					
Type	F	F	F					
Default	none	none	none					

VARIABLE	DESCRIPTION
PID	Part ID or Part Set ID to which weld source is applied
PTYP	PID type: EQ.1: PID defines a single part ID EQ.2: PID defines a part set ID

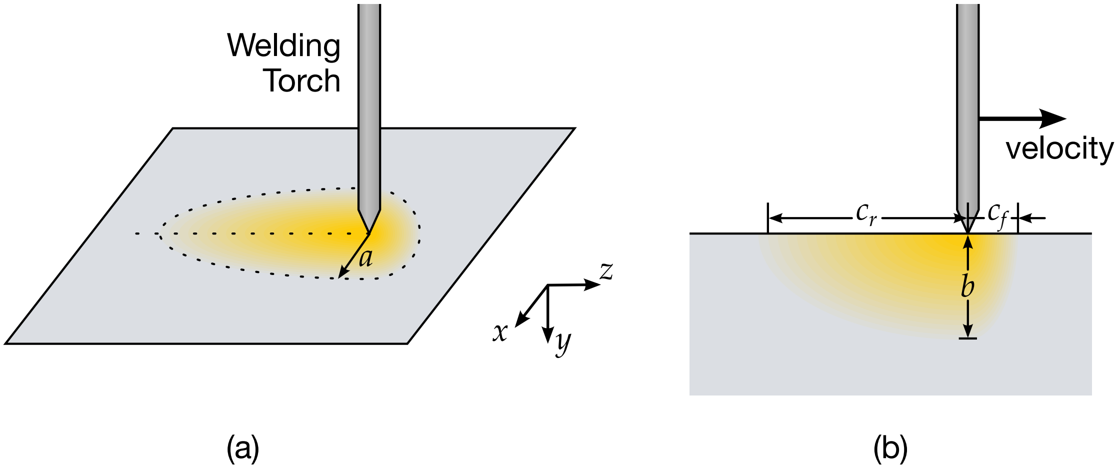


Figure 5-7. Schematic illustration of welding with moving torch. The left figure (a) shows the surface of the material from above, while the right figure (b) shows a slice along the dotted line in the *y*-*z* plane.

VARIABLE	DESCRIPTION
NID	Node ID giving location of weld source EQ.0: location defined by (X0, Y0, Z0) below
NFLAG	Flag controlling motion of weld source EQ.1: source moves with node NID EQ.2: source is fixed in space at original position of node NID
X0, Y0, Z0	Coordinates of weld source, which remains fixed in space (optional, ignored if NID nonzero above)
N2ID	Second node ID for weld beam aiming direction GT.0: beam is aimed from N2ID to NID, moves with these nodes EQ.-1: beam aiming direction is (tx, ty, tz) input on optional card 3
a	Weld pool radius (i.e., half width)
b	Weld pool depth (in beam aiming direction)
cf	Weld pool forward direction
cr	Weld pool rearward direction

VARIABLE	DESCRIPTION
LCID	Load curve ID for weld energy input rate vs. time EQ.0: use constant multiplier value Q.
Q	Curve multiplier for weld energy input rate [energy/time, e.g., Watt] LT.0: use absolute value and accurate integration of heat
Ff	Forward distribution function
Fr	Rear distribution function (Note:$F_f + F_r = 2.0$)
TX, TY, TZ	Weld beam direction vector in global coordinates (N2ID = -1 only)

Remarks:

This boundary condition allows simulation of a moving weld heat source, following the work of Goldak, Chakravarti, and Bibby [1984]. Heat is generated in an ellipsoidal region centered at the weld source, and decaying exponentially with distance according to:

$$q = \frac{6\sqrt{3}FQ}{\pi\sqrt{\pi}abc}\exp\left(\frac{-3x^2}{a^2}\right)\exp\left(\frac{-3y^2}{b^2}\right)\exp\left(\frac{-3z^2}{c^2}\right)$$

where:

$$q = \text{weld source power density}$$

$$(x, y, z) = \text{coordinates of point } p \text{ in weld material}$$

$$F = \begin{cases} \mathrm{F_f} \text{ if point } p \text{ is in front of beam} \\ \mathrm{F_r} \text{ if point } p \text{ is behind beam} \end{cases}$$

$$c = \begin{cases} \mathrm{c_f} \text{ if point } p \text{ is in front of beam} \\ \mathrm{c_r} \text{ if point } p \text{ is behind beam} \end{cases}$$

A local coordinate system is constructed which is centered at the heat source. The relative velocity vector of the heat source defines the "forward" direction, so material points that are approaching the heat source are in "front" of the beam. The beam aiming direction is used to compute the weld pool depth. The weld pool width is measured normal to the relative velocity - aiming direction plane. If Q is defined negative in the input, then the formula above is using the absolute value of Q, and a more accurate integration of the heat source is performed with some additional cost in CPU time.

To simulate a welding process during which the welding torch is fixed in space, NID and N2ID must be set to 0 and -1 respectively. The X0, Y0, and Z0 fields specify the global coordinates of the welding torch, and the TX, TY, and TZ fields specify the direction of the welding beam. The motion of the work piece is prescribed using the *BOUNDARY_PRESCRIBED_MOTION keyword.

To simulate a welding process for which the work piece fixed in space, NID and N2ID specify both the beam source location and direction. The X0, Y0, Z0, TX, TY, and TZ fields *are ignored*. The motion of welding source is prescribed with using the *BOUNDARY_PRESCRIBED_MOTION keyword applied to the two nodal points specified in the NID and N2ID fields.

*BOUNDARY_THERMAL_WELD_TRAJECTORY

Purpose: Define a moving heat source to model welding of solid or shell structures. Motion of the source is described by a nodal path and a prescribed velocity on this path. This keyword is applicable in coupled thermal-structural and thermal-only simulations and also supports thermal dumping. Different equivalent heat source descriptions are implemented.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	PTYP	NSID1	VEL1	SID2	VEL2	NCYC	RELVEL
Type	I	I	I	F	I	F	I	I
Default	none	1	none	none	none	none	1	0

Card 2	1	2	3	4	5	6	7	8
Variable	IFORM	LCID	Q	LCROT	LCMOV	LCLAT	DISC	
Type	I	I	F	I	I	I	F	
Default	none	none	none	none	none	none	none	

Card 3	1	2	3	4	5	6	7	8
Variable	P1	P2	P3	P4	P5	P6	P7	P8
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

Weld Source Aiming Direction Card. Additional card for NS2ID = 0.

Optional	1	2	3	4	5	6	7	8
Variable	TX	TY	TZ					
Type	F	F	F					
Default	none	none	none					

VARIABLE	DESCRIPTION

PID

Part ID or Part Set ID of solids or shells to which weld source is applied

PTYP

PID type:

EQ.1: PID defines a single part ID

EQ.2: PID defines a part set ID

NSID1

Node set ID containing the path (weld trajectory) information for the weld source movement. A sorted node set is requested. The order defines the weld path and the direction (see Remark 1).

VEL1

Velocity of the heat source on the weld trajectory

GT.0: constant velocity

LT.0: |VEL1| is a load curve ID defining weld speed vs. time

SID2

ID of second node set or segment set containing information for the weld source aiming direction (see Remark 2)

GT.0: SID2 refers to a sorted node set, the order of which defines the direction of the trajectory. The heat source is aimed from current position in SID2 to current position in the weld trajectory.

EQ.0: beam aiming direction is (tx, ty, tz) input on optional card 4.

LT.0: |SID2| is a segment set. The heat source is aiming in normal direction to segments in the set.

VARIABLE	DESCRIPTION
VEL2	Velocity of reference point in SID2, if SID2 > 0 GT.0: constant velocity LT.0: |VEL2| is a load curve ID defining weld speed vs. time
NCYC	Number of substeps for subcycling in evaluation of boundary condition. Allows thermal dumping (see Remark 3).
RELVEL	Defines if VEL1 and VEL2 are relative or absolute velocities in coupled simulations EQ.0: absolute velocities EQ.1: relative velocities with respect to underlying structure
IFORM	Geometry description for energy rate density distribution (see Remark 4): EQ.1: Goldak-type heat source EQ.2: double ellipsoidal heat source with constant density EQ.3: double conical heat source with constant density EQ.4: frustum-shaped heat source with constant density
LCID	Load curve ID for weld energy input rate vs. time EQ.0: use constant multiplier value Q.
Q	Curve multiplier for weld energy input rate [energy/time] LT.0: take absolute value and accurate integration of heat using integration cells with edge length DISC.
LCROT	Load curve defining the rotation (angle in degree) of weld source around the trajectory as function of time (see Remark 2).
LCMOV	Load curve for offset of weld source in direction of the weld beam as function of time (see Remark 2)
LCLAT	Load curve for lateral offset of weld source as function of time (see Remark 2)
DISC	Resolution for accurate integration, parameter defines edge length for integration cubes. Default is 5% of weld pool depth.

VARIABLE	DESCRIPTION
P*i*	Parameters defining for weld pool geometry, depending on parameter IFORM. See Remark 4 for details.
TX, TY, TZ	Weld beam direction vector in global coordinates (SID2 = 0 only)

Remarks:

1. This keyword can be applied for solid and thermal thick shells in thermal-only and coupled thermal-structural simulations. The nodes in the node set NSID1 have to be ordered, such that the node set defines the path geometry as well as the direction of the trajectory. The heat source starts at the position of the first node in the node set and automatically ends as soon as the last node is reached.

   By choosing nodes of the work piece for the path definition in NSID1, it can be ensured that the heat source always follows the movement of the piece. By setting parameter RELVEL to 1 the velocity of the heat source can even be defined relatively to the motion of the structure.

2. If a segment set referred to in SID2 which coincides with the work piece surface, the weld beam direction is always orthogonal to the work piece surface. To be applicable every two consecutive nodes of node set NSID1 have to be part of at least one segment. In case of more than one segment an averaged normal is computed.

   Based on the trajectory and the weld source aiming direction, a local coordinate system is constructed that is centered at the root of the heat source. By default, the relative velocity vector (on the trajectory) of the heat source defines the "forward" direction $\boldsymbol{r}$, so material points that are approaching the heat source are in "front" of the beam. The weld source aiming direction, denoted by $\boldsymbol{t}$, is used to compute the weld pool depth. The weld pool width (coordinate direction $\boldsymbol{s}$) is measured normal to the relative velocity - aiming direction plane.

   The keyword allows rotating and translating the coordinate system. First, the system is rotated around the vector $\boldsymbol{r}$ by a value given in the load curve LCROT resulting in a new local coordinate system $(\boldsymbol{r}, \boldsymbol{s}', \boldsymbol{t}')$. Second, the system is translated in directions $\boldsymbol{t}'$ and $\boldsymbol{s}'$ using LCMOV and LCLAT, respectively.

3. The subcycling method introduces an individual time step size for the weld source evaluation. Within one time step of the heat transfer solver, NCYC steps are used to determine the energy rate distribution of the boundary condition. In each substep the geometry of the weld pool is updated. Therefore, even with larger thermal time steps a relatively smooth temperature field around the weld source can be obtained and a jumping heat source across elements can be sup-

pressed.

4.This keyword allows application of different equivalent heat source geometries, depending on the parameter IFORM. The definition of the local coordinate system needed for the description of the weld pool shape is discussed in Remark 2.

For IFORM.EQ.1 heat is generated in an ellipsoidal region centered at the weld source, and decaying exponentially with distance according to the work of Goldak, Chakravarti, and Bibby [1984]. Energy rate distribution is governed by

$$q = \frac{2n\sqrt{n}FQ}{\pi\sqrt{\pi}abc}\exp\left(\frac{-nx^2}{a^2}\right)\exp\left(\frac{-ny^2}{b^2}\right)\exp\left(\frac{-nz^2}{c^2}\right)$$

where:

$$F = \begin{cases}\mathrm{F_f}\text{ if point }p\text{ is in front of beam}\\ \mathrm{F_r}\text{ if point }p\text{ is behind beam}\end{cases}$$

$$c = \begin{cases}\mathrm{c_f}\text{ if point }p\text{ is in front of beam}\\ \mathrm{c_r}\text{ if point }p\text{ is behind beam}\end{cases}$$

The local coordinates of point p are denoted by (x, y, z) and it is expected that the sum of the weighting factors $F_\mathrm{f}, F_\mathrm{r}$ equals 2. The half-width of the ellipsoid is given by a, the welding depth by b. The complete set of parameters $(a, b, c_\mathrm{f}, c_\mathrm{r}, F_\mathrm{f}, F_\mathrm{r}, n)$ is input by the parameters P1 to P7, see table below.

The energy rate density q defined by IFORM.EQ.2 is assumed to be constant in the double ellipsoidal region as defined for Goldak-type heat sources. Its value is given by

$$q = \frac{3FQ}{2\pi abc}$$

with the same assumptions for F and c as above. The set of parameters consequently reduces to $(a, b, c_\mathrm{f}, c_\mathrm{r}, F_\mathrm{f}, F_\mathrm{r})$, the input of which is given by P1 to P6.

In contrast to the above IFORM.EQ.3 defines an equivalent heat source with a constant energy rate density on a double conical region. The shape is defined by three radii $r_1, r_2, r_3$and two values b_1, b_2 defining the heights of the two parts of the shape. The respective power densities of the parts are given by

$$q_i = \frac{3F_iQ}{2\pi b_i(r_i^2 + r_{i+1}^2 + r_i^2 r_{i+1}^2)}.$$

Here it is assumed that $i = 1$ corresponds to the part closer to the weld source. The input for the complete parameter set $(r_1, r_2, r_3, b_1, b_2, F_1, F_2)$ is again defined by P1 to P7.

Finally, IFORM.EQ.4 defines a constant power density over a frustum. The density and the shape can easily be described using three geometrical parameters P1 to P3 corresponding to the radii r_1(at the heat source origin) and r_2 and the height b:

$$q = \frac{3}{\pi b(r_1^2 + r_2^2 + r_1^2 r_2^2)}$$

IFORM	1	2	3	4
P1	a	a	r_1	r_1
P2	b	b	r_2	r_2
P3	c_f	c_f	r_3	b_1
P4	c_r	c_r	b_1	
P5	F_f	F_f	b_2	
P6	F_r	F_r	F_f	
P7	n		F_r	
P8				

*BOUNDARY_USA_SURFACE

Purpose: Define a surface for coupling with the USA code [DeRuntz 1993]. The outward normal vectors should point into the fluid media. The coupling with USA is operational in explicit transient and in implicit natural frequency analyses.

Card	1	2	3	4	5	6	7	8
Variable	SSID	WETDRY	NBEAM					
Type	I	I	I					
Default	none	0	0					

VARIABLE	DESCRIPTION
SSID	Segment set ID, see *SET_SEGMENT
WETDRY	Wet surface flag: EQ.0: Dry, no coupling for USA DAA analysis, or Internal fluid coupling for USA CASE analysis EQ.1: Wet, coupled with USA for DAA analysis, or External fluid coupling for USA CASE analysis
NBEAM	The number of nodes touched by USA Surface-of-Revolution (SOR) elements. It is not necessary that the LS-DYNA model has beams where USA has beams (i.e., SOR elements), merely that the LS-DYNA model has nodes to receive the forces that USA will return.

Remarks:

The underwater shock analysis code is an optional module. To determine availability contact sales@lstc.com.

The wet surface of 3 and 4-noded USA general boundary elements is defined in LS-DYNA with a segment set of 4-noded surface segments, where the fourth node can duplicate the third node to form a triangle. The segment normal vectors should be directed into the USA fluid. If USA overlays are going to be used to reduce the size of the DAA matrices, the user should nonetheless define the wet surface here as if no overlay were being used. If Surface-of -Revolution elements (SORs) are being used in USA, then NBEAM should be non-zero on one and only one card in this section.

The wet surface defined here can cover structural elements or acoustic fluid volume elements, but it can not touch both types in one model.

When running a coupled problem with USA, the procedure requires an additional input file of USA keyword instructions. These are described in a separate USA manual. The name of this input file is identified on the command line with the usa = flag:

```
LSDYNA.USA i=inf  usa=uin
```

where **uin** is the USA keyword instruction file.

BOUNDARY_ELEMENT_METHOD_*OPTION

Available options include:

CONTROL

FLOW

NEIGHBOR

SYMMETRY

WAKE

Purpose: Define input parameters for boundary element method analysis of incompressible fluid dynamics or fluid-structure interaction problems.

The boundary element method (BEM) can be used to compute the steady state or transient fluid flow about a rigid or deformable body. The theory which underlies the method (see the LS-DYNA Theory Manual) is restricted to inviscid, incompressible, attached fluid flow. The method should not be used to analyze flows where shocks or cavitation are present.

In practice the method can be successfully applied to a wider class of fluid flow problems than the assumption of inviscid, incompressible, attached flow would imply. Many flows of practical engineering significance have large Reynolds numbers (above 1 million). For these flows the effects of fluid viscosity are small if the flow remains attached, and the assumption of zero viscosity may not be a significant limitation. Flow separation does not necessarily invalidate the analysis. If well-defined separation lines exist on the body, then wakes can be attached to these separation lines and reasonable results can be obtained. The Prandtl-Glauert rule can be used to correct for non-zero Mach numbers in a gas, so the effects of aerodynamic compressibility can be correctly modeled (as long as no shocks are present).

The BOUNDARY_ELEMENT_METHOD_FLOW card turns on the analysis, and is mandatory.

*BOUNDARY_ELEMENT_METHOD_CONTROL

Purpose: Control the execution time of the boundary element method calculation. The CONTROL option is used to control the execution time of the boundary element method calculation, and the use of this option is strongly recommended. The BEM calculations can easily dominate the total execution time of a LS-DYNA run unless the parameters on this card (especially DTBEM and/or IUPBEM) are used appropriately.

DTBEM is used to increase the time increment between calls to the BEM routines. This can usually be done with little loss in accuracy since the characteristic times of the structural dynamics and the fluid flow can differ by several orders of magnitude. The characteristic time of the structural dynamics in LS-DYNA is given by the size of the smallest structural element divided by the speed of sound of its material. For a typical problem this characteristic time might be equal to 1 microsecond. Since the fluid in the boundary element method is assumed to be incompressible (infinite speed of sound), the characteristic time of the fluid flow is given by the streamwise length of the smallest surface in the flow divided by the fluid velocity. For a typical problem this characteristic time might be equal to 10 milliseconds. For this example DTBEM might be set to 1 millisecond with little loss of accuracy. Thus, for this example, the boundary element method would be called only once for every 1000 LS-DYNA iterations, saving an enormous amount of computer time.

IUPBEM is used to increase the number of times the BEM routines are called before the matrix of influence coefficients is recomputed and factored (these are time-consuming procedures). If the motion of the body is entirely rigid body motion there is no need to ever recompute and factor the matrix of influence coefficients after initialization, and the execution time of the BEM can be significantly reduced by setting IUPBEM to a very large number. For situations where the structural deformations are modest an intermediate value (e.g., 10) for IUPBEM can be used.

Card 1	1	2	3	4	5	6	7	8
Variable	LWAKE	DTBEM	IUPBEM	FARBEM				
Type	I	F	I	F				
Default	50	0.	100	2.0				
Remark	1			2				

VARIABLE	DESCRIPTION
LWAKE	Number of elements in the wake of lifting surfaces. Wakes must be defined for all lifting surfaces.
DTBEM	Time increment between calls to the boundary element method. The fluid pressures computed during the previous call to the BEM will continue to be used for subsequent LS-DYNA iterations until a time increment of DTBEM has elapsed.
IUPBEM	The number of times the BEM routines are called before the matrix of influence coefficients is recomputed and refactored.
FARBEM	Nondimensional boundary between near-field and far-field calculation of influence coefficients.

Remarks:

1. Wakes convect with the free-stream velocity. The number of elements in the wake should be set to provide a total wake length equal to 5-10 times the characteristic streamwise length of the lifting surface to which the wake is attached. Note that each wake element has a streamwise length equal to the magnitude of the free stream velocity multiplied by the time increment between calls to the boundary element method routines. This time increment is controlled by DTBEM.

2. The most accurate results will be obtained with FARBEM set to 5 or more, while values as low as 2 will provide slightly reduced accuracy with a 50% reduction in the time required to compute the matrix of influence coefficients.

*BOUNDARY_ELEMENT_METHOD_FLOW

Purpose: Turn on the boundary element method calculation, specify the set of shells which define the surface of the bodies of interest, and specify the onset flow.

The *BOUNDARY_ELEMENT_METHOD_FLOW card turns on the BEM calculation. This card also identifies the shell elements which define the surfaces of the bodies of interest, and the properties of the onset fluid flow. The onset flow can be zero for bodies which move through a fluid which is initially at rest.

Card 1	1	2	3	4	5	6	7	8
Variable	SSID	VX	VY	VZ	RO	PSTATIC	MACH	
Type	I	F	F	F	F	F	F	
Default	none	none	none	none	none	0.	0.	
Remark	1					2	3	

VARIABLE	DESCRIPTION
SSID	Shell set ID for the set of shell elements which define the surface of the bodies of interest (see *SET_SHELL). The nodes of these shells should be ordered so that the shell normals point into the fluid.
VX, VY, VZ	x, y, and z components of the free-stream fluid velocity.
RO	Fluid density.
PSTATIC	Fluid static pressure.
MACH	Free-stream Mach number.

Remarks:

1. It is recommended that the shell segments in the SSID set use the NULL material (see *MAT_NULL). This will provide for the display of fluid pressures in the post-processor. For triangular shells the 4th node number should be the same as the 3rd node number. For fluid-structure interaction problems it is recommended that the boundary element shells use the same nodes and be

coincident with the structural shell elements (or the outer face of solid elements) which define the surface of the body. This approach guarantees that the boundary element segments will move with the surface of the body as it deforms.

2. A pressure of PSTATIC is applied uniformly to all segments in the segment set. If the body of interest is hollow, then PSTATIC should be set to the free-stream static pressure minus the pressure on the inside of the body.

3. The effects of subsonic compressibility on gas flows can be included using a non-zero value for MACH. The pressures which arise from the fluid flow are increased using the Prandtl-Glauert compressibility correction. MACH should be set to zero for water or other liquid flows.

*BOUNDARY_ELEMENT_METHOD_NEIGHBOR

Purpose: Define the neighboring elements for a given boundary element segment.

The pressure at the surface of a body is determined by the gradient of the doublet distribution on the surface (see the LS-DYNA Theory Manual). The "Neighbor Array" is used to specify how the gradient is computed for each boundary element segment. Ordinarily, the Neighbor Array is set up automatically by LS-DYNA, and no user input is required. The NEIGHBOR option is provided for those circumstances when the user desires to define this array manually.

Elements Cards. The next "*" card terminates the input.

Card 1	1	2	3	4	5	6	7	8
Variable	NELEM	NABOR1	NABOR2	NABOR3	NABOR4			
Type	I	I	I	I	I			
Default	none	none	none	none	none			

VARIABLE	DESCRIPTION
NELEM	Element number.
NABOR1	Neighbor for side 1 of NELEM.
NABOR2	Neighbor for side 2 of NELEM.
NABOR3	Neighbor for side 3 of NELEM.
NABOR4	Neighbor for side 4 of NELEM.

Remarks:

Each boundary element has 4 sides (Figure 6-1). Side 1 connects the 1st and 2nd nodes, side 2 connects the 2nd and 3rd nodes, etc. The 4th side is null for triangular elements.

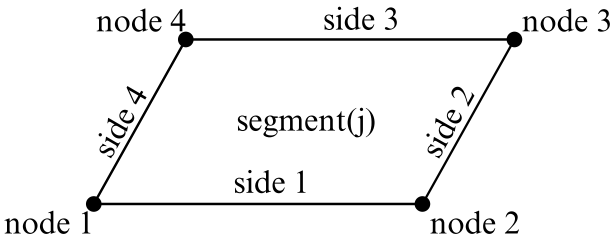


Figure 6-1. Each segment has 4 sides.

For most elements the specification of neighbors is straightforward. For the typical case a quadrilateral element is surrounded by 4 other elements, and the neighbor array is as shown in Figure 6-2.

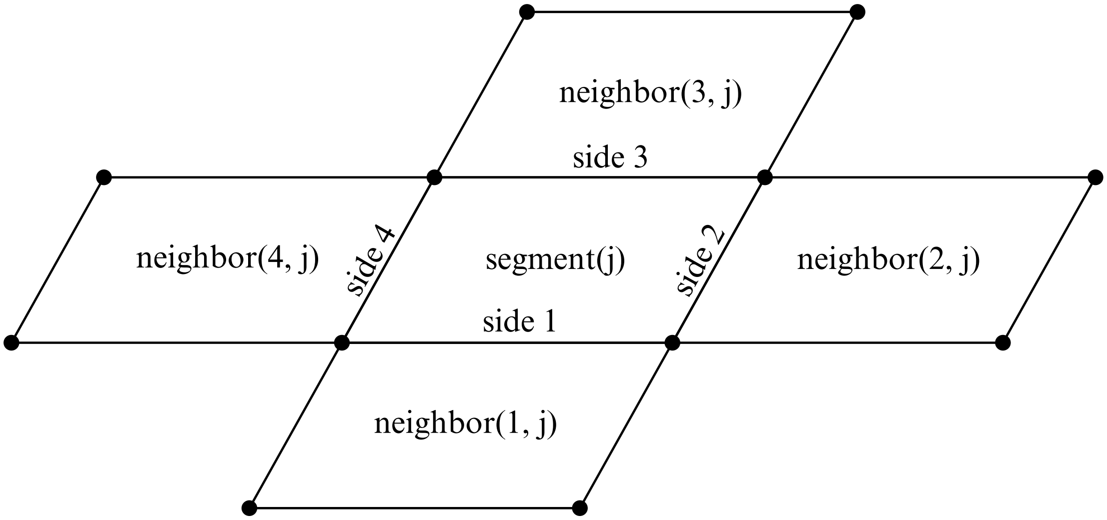


Figure 6-2. Typical neighbor specification.

There are several situations for which the user may desire to directly specify the neighbor array for certain elements. For example, boundary element wakes result in discontinuous doublet distributions, and neighbors which cross a wake should not be used. Figure 6-3 illustrates a situation where a wake is attached to side 2 of segment j. For this situation two options exist. If neighbor(2,j) is set to zero, then a linear computation of the gradient in the side 2 to side 4 direction will be made using the difference between the doublet strengths on segment j and segment neighbor(4,j). This is the default setup used by LS-DYNA when no user input is provided. By specifying neighbor(2,j) as a negative number a more accurate quadratic curve fit will be used to compute the gradient. The curve fit will use segment j, segment neighbor(4,j), and segment -neighbor(2,j); which is located on the opposite side of segment neighbor(4,j) as segment j.

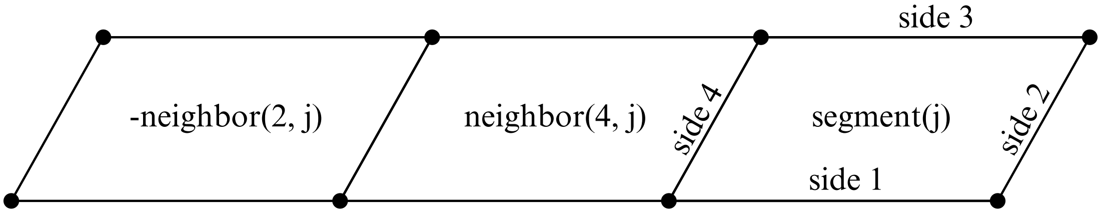


Figure 6-3. If neighbor(2,j) is a negative number it is assumed to lie on the opposite side of neighbor(4,j) as segment j.

Another possibility is that no neighbors at all are available in the side 2 to side 4 direction. In this case both neighbor(2,j) and neighbor(4,j) can be set to zero, and the gradient in that direction will be assumed to be zero. This option should be used with caution, as the resulting fluid pressures will not be accurate for three-dimensional flows. However, this option is occasionally useful where quasi-two dimensional results are desired. All of the above options apply to the side 1 to side 3 direction in the obvious ways.

For triangular boundary elements side 4 is null. Gradients in the side 2 to side 4 direction can be computed as described above by setting neighbor(4,j) to zero for a linear derivative computation (this is the default setup used by LS-DYNA when no user input is provided) or to a negative number to use the segment on the other side of neighbor(2,j) and a quadratic curve fit. There may also be another triangular segment which can be used as neighbor(4,j) (see Figure 6-4).

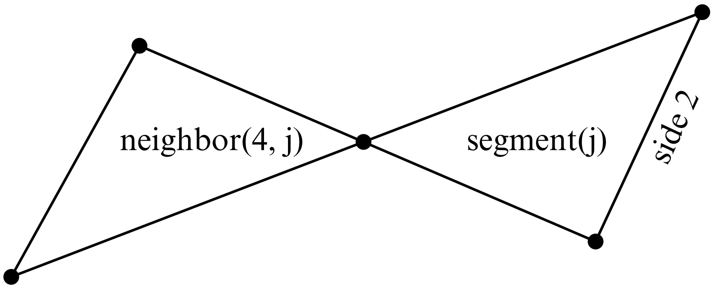


Figure 6-4. Sometimes another triangular boundary element segment can be used as neighbor (4,j).

The rules for computing the doublet gradient in the side 2 to side 4 direction can be summarized as follows (the side 1 to side 3 case is similar):

NABOR2	NABOR4	Doublet Gradient Computation
GT.0	GT.0	Quadratic fit using elements j, NABOR2, and NABOR4.
LT.0	GT.0	Quadratic fit using elements j, -NABOR2, and NABOR4. -NABOR2 is assumed to lie on the opposite side of NABOR4 as segment j (see Fig. 6-3).
GT.0	LT.0	Quadratic fit using elements j, NABOR2, and -NABOR4. -NABOR4 is assumed to lie on the opposite side of NABOR2 as segment j.
EQ.0	GT.0	Linear fit using elements j and NABOR4.
GT.0	EQ.0	Linear fit using elements j and NABOR2.
EQ.0	EQ.0	Zero gradient.

Table 3.1 Surface pressure computation for element j.

*BOUNDARY_ELEMENT_METHOD_SYMMETRY

Purpose: To define a plane of symmetry for the boundary element method. The SYMMETRY option can be used to reduce the time and memory required for symmetric configurations. For these configurations the reduction in the number of boundary elements by a factor of 2 will reduce the memory used by the boundary element method by a factor of 4, and will reduce the computer time required to factor the matrix of influence coefficients by a factor of 8. Only 1 plane of symmetry can be defined.

Card 1	1	2	3	4	5	6	7	8
Variable	BEMSYM							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
BEMSYM	Defines symmetry plane for boundary element method. EQ.0: no symmetry plane is defined EQ.1: x = 0 is a symmetry plane EQ.2: y = 0 is a symmetry plane EQ.3: z = 0 is a symmetry plane

*BOUNDARY_ELEMENT_METHOD_WAKE

Purpose: To attach wakes to the trailing edges of lifting surfaces. Wakes should be attached to boundary elements at the trailing edge of a lifting surface (such as a wing, propeller blade, rudder, or diving plane). Wakes should also be attached to known separation lines when detached flow is known to exist (such as the sharp leading edge of a delta wing at high angles of attack). Wakes are required for the correct computation of surface pressures for these situations. As described above, two segments on opposite sides of a wake should never be used as neighbors.

Element Cards. (The next "*" card terminates the input.)

Card 1	1	2	3	4	5	6	7	8
Variable	NELEM	NSIDE						
Type	I	I						
Default	none	none						
Remark	1							

VARIABLE	DESCRIPTION
NELEM	Element number to which a wake is attached.
NSIDE	The side of NELEM to which the wake is attached (see Fig. 3.2). This should be the "downstream" side of NELEM.

Remarks:

1. Normally two elements meet at a trailing edge (one on the "upper" surface and one on the "lower" surface). The wake can be attached to either element, but not to both.

*CASE

The *CASE command provides a way of running multiple LS-DYNA analyses (or cases) sequentially by submitting a single input file. When *CASE commands are used to define multiple cases, some portions of the input will be shared by some or all of the cases and other portions will be unique to each case. Because the cases are run sequentially, the results from one case, e.g., a dynain file, can be used in the analysis of a different, subsequent case. Each case creates a unique set of output file names by prepending "case***n***." to the default file name, e.g., case101.d3plot, case102.glstat.

When the *CASE keyword appears in an input deck, it becomes necessary to append the word "CASE" to the LS-DYNA execution line. For example, an SMP LS-DYNA execution line might look something like

```
path_to_ls-dyna i=input.k ncpu=-4 CASE
```

An MPP LS-DYNA execution line might look something like

```
mpirun –np 4  path_to_mpp971  i=input.k CASE
```

__CASE___{OPTION}*

Available options include:

<BLANK>

BEGIN_*N*

END_*N*

Purpose: Define a series of cases and perhaps subcases. The options *CASE_BEGIN_*n* and *CASE_END_*n* appear in pairs and *n* is a numeric ID of a subcase. Subcase IDs may be referenced by the *CASE command in defining a case. In other words, a case may consist of one or more subcases. All keywords appearing between *CASE_BEGIN_*n* and *CASE_END_*n* comprise subcase *n*. If no *CASE command is defined, then subcases defined by *CASE_BEGIN_*n* and *CASE_END_*n* then become cases. *CASE_BEGIN/*CASE_END can be nested, overlapped, and disjointed. Examples below demonstrate the use of these options.

An alternative way of defining subcases is by appending the string "CID = *n*" to the end of any keyword command. Any keyword so tagged will then be active only for those cases that reference subcase *n*. There can be more than one space between the keyword and "CID = *n*".

Any keyword in the input deck not associated with a subcase is active for all cases.

The following input syntax applies only to the *CASE command, not to *CASE_BEGIN/*CASE_END.

Card 1	1	2	3	4	5	6	7	8
Variable	CASEID	JOBID						
Type	I	C						
Default	none	none						

Command Line Argument Cards. Command line cards set additional command line arguments for the case CASEID (see Card 1, above). Include as many as needed, or as few as none. Command line cards end when the first character of the next card is numeric.

Card 2	1	2	3	4	5	6	7	8
Variable	COMMANDS							
Type	A							
Default	Not Required							

Subcase ID Cards. Define active subcase IDs for case CASEID (see Card 1, above). These cards continue until the next keyword ("*") card.

Card 3	1	2	3	4	5	6	7	8
Variable	SCID1	SCID2	…	…	…	…	…	…
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
CASEID	Identification number for case.

VARIABLE **DESCRIPTION**

JOBID Optional string (no spaces) to be used as the jobid for this case. If no JOBID is specified, the string CASEXX is used, where XX is the CASEID in field 1

COMMANDS Command line arguments.

SCIDn Subcase ID active for case CASEID.

Remarks:

1. If no *CASE keyword appears, subcases defined with *CASE_BEGIN/*CASE_END commands become cases and *CASE_BEGIN can optionally be followed by extra command line arguments.

2. If no *CASE keyword appears, it is an error to append "CID = n" to any keyword.

3. If multiple *CASE or *CASE_BEGIN keywords appear that have the same ID, their command line arguments and active commands are merged.

4. The *CASE or *CASE_BEGIN keywords cannot be used within an include (*INCLUDE) file.

Example 1:

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ Define case 101 which includes subcase 1.
$ Define case 102 which includes subcase 4.
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
*CASE
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7
$   CASEID
       101 JOBID_FOR_CASE101
MEMORY=20M
1
$
*CASE
$   CASEID
       102
MEMORY=20M  NCYCLE=1845
4
$
*TITLE  CID=1
THIS IS THE TITLE FOR CASE 101
$
*TITLE  CID=4
THIS IS THE TITLE FOR CASE 102
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

Example 2:

```
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$  Illustrate overlapping subcases.
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
*CASE_BEGIN_5
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>
*DATABASE_BINARY_D3THDT
1.e-5
*CASE_BEGIN_3
*DATABASE_NODOUT
1.e-5
*CASE_END_5
*DATABASE_ELOUT
1.e-5
*CASE_END_3
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

Example 2 above will generate d3thdt and nodout for CID = 5, and nodout and elout for CID = 3.

*COMMENT

All input that falls between a *COMMENT command and the subsequent line of input that has an asterisk in the first column thereby signaling the start of another keyword command, is not acted on by LS-DYNA. This provides a convenient way to interject multiple, successive lines of commentary anywhere inside an input deck.

*COMMENT also provides a convenient way to comment out an exisiting keyword command and all its associated input data as shown in an example below.

Lines of input that are deactivated by *COMMENT are echoed on the screen and to the messag and d3hsp files.

Card 1	1	2	3	4	5	6	7	8
Variable	COMMENT							
Type	A							
Default	none							

VARIABLE	DESCRIPTION
COMMENT	Any comment line.

Example:

In this excerpt from an input deck, 5 lines of comments including blank lines, are added to the input deck.

```
*KEYWORD
*COMMENT
Units of this model are mks.

Input prepared by John Doe.
Input checked by Jane Doe.

*CONTROL_TERMINATION
1.E-02
⋮
```

Example 2:

In this excerpt from an input deck, a contact is disabled by inserting *COMMENT command before the contact keyword command.

```
⋮
*COMMENT  *CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_ID
$#     cid                                                                 title
         1
$#    ssid      msid     sstyp     mstyp    sboxid    mboxid       spr       mpr
1,2,0,3
$#      fs        fd        dc        vc       vdc    penchk        bt        dt
0.2
$#     sfs       sfm       sst       mst      sfst      sfmt       fsf       vsf

$#    soft    sofscl    lcidab    maxpar     sbopt     depth     bsort    frcfrq
         2
*SET_SEGMENT
$#     sid       da1       da2       da3       da4    solver
         1     0.000     0.000     0.000     0.000MECH
$#      n1        n2        n3        n4        a1        a2        a3        a4
      2842       626      3232      3242     0.000     0.000     0.000     0.000
      2846      2842       627      2843     0.000     0.000     0.000     0.000
⋮
```

*COMPONENT

The keyword *COMPONENT provides a way of incorporating specialized components and features. The keyword control cards in this section are defined in alphabetical order:

COMPONENT_GEBOD_*OPTION

COMPONENT_GEBOD_JOINT_*OPTION

*COMPONENT_HYBRIDIII

COMPONENT_HYBRIDIII_JOINT_*OPTION

COMPONENT_GEBOD_*OPTION

Purpose: Generate a rigid body dummy based on dimensions and mass properties from the GEBOD database. The motion of the dummy is governed by equations integrated within LS-DYNA separately from the finite element model. Default joint characteristics (stiffness's, stop angles, etc.) are set internally and should give reasonable results, however, they may be altered using the *COMPONENT_GEBOD_-JOINT command. Contact between the segments of the dummy and the finite element model is defined using the *CONTACT_GEBOD command. The use of a positioning file is essential with this feature, see Appendix N for further details.

OPTION specifies the human subject type. The male and female types represent adults while the child is genderless.

MALE

FEMALE

CHILD

Card 1	1	2	3	4	5	6	7	8
Variable	DID	UNITS	SIZE					
Type	I	I	F					
Default	none	none	none					

Card 2	1	2	3	4	5	6	7	8
Variable	VX	VY	VZ	GX	GY	GZ		
Type	F	F	F	F	F	F		
Default	0.	0.	0.	0.	0.	0.		

VARIABLE	DESCRIPTION
DID	Dummy ID. A unique number must be specified.

VARIABLE **DESCRIPTION**

UNITS System of units used in the finite element model.

EQ.1: lbf × sec2/in - inch - sec

EQ.2: kg - meter - sec

EQ.3: kgf × sec2/mm - mm - sec

EQ.4: metric ton - mm - sec

EQ.5: kg - mm - msec

SIZE Size of the dummy. This represents a combined height and weight percentile ranging from 0 to 100 for the male and female types. For the child the number of months of age is input with an admissible range from 24 to 240.

VX, VY, VZ Initial velocity of the dummy in the global x, y and z directions.

GX, GY, GZ Global x, y, and z components of gravitational acceleration applied to the dummy.

Example:

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$   *COMPONENT_GEBOD_MALE
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  A 50th percentile male dummy with the ID number of 7 is generated in the
$  lbf*sec^2-inch-sec system of units.  The dummy is given an initial velocity of
$  616 in/sec in the negative x direction and gravity acts in the negative z
$  direction with a value 386 in/sec^2.
$
*COMPONENT_GEBOD_MALE
$
$...>....1...>....2...>....3...>....4...>....5...>....6...>....7...>....8
$      did     units      size
         7         1        50
$       vx        vy        vz        gx        gy        gz
      -616         0         0         0         0      -386
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
```

COMPONENT_GEBOD_JOINT_*OPTION

Purpose: Alter the joint characteristics of a GEBOD rigid body dummy. Setting a joint parameter value to zero retains the default value set internally. See Appendix N for further details.

The following options are available.

PELVIS

WAIST

LOWER_NECK

UPPER_NECK

LEFT_SHOULDER

RIGHT_SHOULDER

LEFT_ELBOW

RIGHT_ELBOW

LEFT_HIP

RIGHT_HIP

LEFT_KNEE

RIGHT_KNEE

LEFT_ANKLE

RIGHT_ANKLE

Card 1	1	2	3	4	5	6	7	8
Variable	DID	LC1	LC2	LC3	SCF1	SCF2	SCF3	
Type	F	I	I	I	F	F	F	

VARIABLE	DESCRIPTION
DID	Dummy ID, see *COMPONENT_GEBOD_*OPTION*.
LC*i*	Load curve ID specifying the loading torque versus rotation (in radians) for the i^{th} degree of freedom of the joint.
SCF*i*	Scale factor applied to the load curve of the i^{th} joint degree of freedom.

Card 2	1	2	3	4	5	6	7	8
Variable	C1	C2	C3	NEUT1	NEUT2	NEUT3		
Type	F	F	F	F	F	F		

VARIABLE	DESCRIPTION
C*i*	Linear viscous damping coefficient applied to the *i*th DOF of the joint. Units are torque × time/radian, where the units of torque and time depend on the choice of UNITS in card 1 of *COMPONENT_GEBOD_*OPTION.*
NEUT*i*	Neutral angle (degrees) of joint's *i*th DOF.

Card 3	1	2	3	4	5	6	7	8
Variable	LOSA1	HISA1	LOSA2	HISA2	LOSA3	HISA3		
Type	F	F	F	F	F	F		

VARIABLE	DESCRIPTION
LOSA*i*	Value of the low stop angle (degrees) for the *i*th DOF of this joint.
HISA*i*	Value of the high stop angle (degrees) for the *i*th DOF of this joint.

Card 4	1	2	3	4	5	6	7	8
Variable	UNK1	UNK2	UNK3					
Type	F	F	F					
Default	0.	0.	0.					

VARIABLE	DESCRIPTION
UNK*i*	Unloading stiffness (torque/radian) for the *i*th degree of freedom of the joint. This must be a positive number. Units of torque depend on the choice of UNITS in card 1 of *COMPONENT_GEBOD_*OPTION.*

Example 1:

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$   *COMPONENT_GEBOD_JOINT_LEFT_SHOULDER
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  The damping coefficients applied to all three degrees of freedom of the left
$  shoulder of dummy 7 are set to 2.5.  All other characteristics of this joint
$  remain set to the default value.
$
*COMPONENT_GEBOD_JOINT_LEFT_SHOULDER
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      did       lc1       lc2       lc3      scf1      scf2      scf3
         7         0         0         0         0         0         0
$       c1        c2        c3     neut1     neut2     neut3
       2.5       2.5       2.5         0         0         0
$    losa1     hisa1     losa2     hisa2     losa3     hisa3
         0         0         0         0         0         0
$     unk1      unk2      unk3
         0         0         0
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

Example 2:

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$   *COMPONENT_GEBOD_JOINT_WAIST
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Load curve 8 gives the torque versus rotation relationship for the 2nd DOF
$  (lateral flexion) of the waist of dummy 7.  Also, the high stop angle of the
$  1st DOF (forward flexion) is set to 45 degrees.  All other characteristics
$  of this joint remain set to the default value.
$
*COMPONENT_GEBOD_JOINT_WAIST
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      did       lc1       lc2       lc3      scf1      scf2      scf3
         7         0         8         0         0         0         0
$       c1        c2        c3     neut1     neut2     neut3
         0         0         0         0         0         0
$    losa1     hisa1     losa2     hisa2     losa3     hisa3
         0        45         0         0         0         0
$     unk1      unk2      unk3
         0         0         0
$
```

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

*COMPONENT_HYBRIDIII

Purpose: Define a HYBRID III dummy. The motion of the dummy is governed by equations integrated within LS-DYNA separately from the finite element model. The dummy interacts with the finite element structure through contact interfaces. Joint characteristics (stiffnesses, damping, friction, etc.) are set internally and should give reasonable results, however, they may be altered using the *COMPONENT_HYBRIDIII_JOINT command. Joint force and moments can be written to an ASCII file (see *DATABASE_H3OUT).

Card 1	1	2	3	4	5	6	7	8
Variable	DID	SIZE	UNITS	DEFRM	VX	VY	VZ	
Type	I	I	I	I	F	F	F	
Default	none	none	none	1	0.	0.	0.	

VARIABLE	DESCRIPTION

DID

Dummy ID. A unique number must be specified.

SIZE

Size of dummy.

EQ.1: 5th percentile adult

EQ.2: 50th percentile adult

EQ.3: 95th percentile adult

NOTE: If negative then the best of currently available joint properties are applied.

UNITS

System of units used in the finite element model.

EQ.1: lbf × sec2/in - inch - sec

EQ.2: kg - meter - sec

EQ.3: kgf × sec2/mm - mm - sec

EQ.4: metric ton - mm - sec

EQ.5: kg - mm - msec

VARIABLE	DESCRIPTION
DEFRM	Deformability type. EQ.1: all dummy segments entirely rigid EQ.2: deformable abdomen (low density foam, mat #57) EQ.3: deformable jacket (low density foam, mat #57) EQ.4: deformable headskin (viscoelastic, mat #6) EQ.5: deformable abdomen/jacket EQ.6: deformable jacket/headskin EQ.7: deformable abdomen/headskin EQ.8: deformable abdomen/jacket/headskin
VX, VY, VZ	Initial velocity of the dummy in the global x, y and z directions.

Card 2	1	2	3	4	5	6	7	8
Variable	HX	HY	HZ	RX	RY	RZ		
Type	F	F	F	F	F	F		
Default	0.	0.	0.	0.	0.	0.		

VARIABLE	DESCRIPTION
HX, HY, HZ	Initial global x, y, and z coordinate values of the H-point.
RX, RY, RZ	Initial rotation of dummy about the H-point with respect to the global x, y, and z axes (degrees).

Example 1:

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$  *COMPONENT_HYBRIDIII
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  A 50th percentile adult rigid HYBRID III dummy with an ID number of 7 is defined
$  in the lbf*sec^2-inch-sec system of units.  The dummy is assigned an initial
$  velocity of 616 in/sec in the negative x direction.  The H-point is initially
$  situated at (x,y,z)=(38,20,0) and the dummy is rotated 18 degrees about the
$  global x-axis.
$
```

```
*COMPONENT_HYBRIDIII
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      did    .  size     units     defrm        vx        vy        vz
         7         2         1         1     -616.        0.        0.
$       hx        hy        hz        rx        ry        rz
       38.       20.        0.       18.        0.        0.
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

COMPONENT_HYBRIDIII_JOINT_*OPTION

Purpose: Alter the joint characteristics of a HYBRID III dummy. Setting a joint parameter value to zero retains the default value set internally. Joint force and moments can be written to an ASCII file (see *DATABASE_H3OUT). Further details pertaining to the joints are found in the Hybrid III Dummies section of Appendix N.

The following options are available:

LUMBAR

LOWER_NECK

UPPER_NECK

LEFT_SHOULDER

RIGHT_SHOULDER

LEFT_ELBOW

RIGHT_ELBOW

LEFT_WRIST

RIGHT_WRIST

LEFT_HIP

RIGHT_HIP

LEFT_KNEE

RIGHT_KNEE

LEFT_ANKLE

RIGHT_ANKLE

STERNUM

LEFT_KNEE_SLIDER

RIGHT_KNEE_SLIDER

Card 1	1	2	3	4	5	6	7	8
Variable	DID	Q1	Q2	Q3	FRIC			
Type	F	F	F	F	F			

Card 2	1	2	3	4	5	6	7	8
Variable	C1	ALO1	BLO1	AHI1	BHI1	QLO1	QHI1	SCLK1
Type	F	F	F	F	F	F	F	F

Leave blank if joint has only one degree of freedom.

Card 3	1	2	3	4	5	6	7	8
Variable	C2	ALO2	BLO2	AHI2	BHI2	QLO2	QHI2	SCLK2
Type	F	F	F	F	F	F	F	F

Leave blank if the joint has only two degrees of freedom.

Card 4	1	2	3	4	5	6	7	8
Variable	C3	ALO3	BLO3	AHI3	BHI3	QLO3	QHI3	SCLK3
Type	F	F	F	F	F	F	F	F

VARIABLE	DESCRIPTION
DID	Dummy ID, see *COMPONENT_HYBRIDIII
Q*i*	Initial value of the joint's *i*th degree of freedom. Units of degrees are defined for rotational DOF. See Appendix N for a listing of the applicable DOF.
FRIC	Friction load on the joint.
C*i*	Linear viscous damping coefficient applied to the *i*th DOF of the joint.
ALO*i*	Linear coefficient for the low regime spring of the joint's *i*th DOF.
BLO*i*	Cubic coefficient for the low regime spring of the joint's *i*th DOF.
AHI*i*	Linear coefficient for the high regime spring of the joint's *i*th DOF.
BHI*i*	Cubic coefficient for the high regime spring of the joint's *i*th DOF.
QLO*i*	Value at which the low regime spring definition becomes active.
QHI*i*	Value at which the high regime spring definition becomes active.
SCLK*i*	Scale value applied to the stiffness of the joint's *i*th DOF (default = 1.0).

Example:

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$   *COMPONENT_HYBRIDIII_JOINT_LEFT_ANKLE
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  The damping coefficients applied to all three degrees of freedom of the left
$  ankle of dummy 7 are set to 2.5.  All other characteristics of this joint
$  remain set to the default value.  The dorsi-plantar flexion angle is set to
$  20 degrees.
```

```
$
*COMPONENT_HYBRIDIII_JOINT_LEFT_ANKLE
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      did        q1        q2        q3      fric
         7         0       20.         0         0         0
$       c1      alo1      blo1      ahi1      bhi1      qlo1      qhi1
       2.5         0         0         0         0         0         0
$       c2      alo2      blo2      ahi2      bhi2      qlo2      qhi2
       2.5         0         0         0         0         0         0
$      2.5      alo3      blo3      ahi3      bhi3      qlo3      qhi3
```

*CONSTRAINED

The keyword *CONSTRAINED provides a way of constraining degrees of freedom to move together in some way. The keyword cards in this section are defined in alphabetical order:

*CONSTRAINED_ADAPTIVITY

*CONSTRAINED_BEAM_IN_SOLID

*CONSTRAINED_BUTT_WELD

CONSTRAINED_COORDINATE_*{OPTION}

*CONSTRAINED_EULER_IN_EULER

CONSTRAINED_EXTRA_NODES_*OPTION

CONSTRAINED_GENERALIZED_WELD_*OPTION_{OPTION}

*CONSTRAINED_GLOBAL

CONSTRAINED_INTERPOLATION_*{OPTION}

**CONSTRAINED_INTERPOLATION_SPOTWELD*

CONSTRAINED_JOINT_*OPTION_{OPTION}_{OPTION}_{OPTION}

CONSTRAINED_JOINT_COOR_*OPTION_{OPTION}_{OPTION}_{OPTION}

CONSTRAINED_JOINT_STIFFNESS_*OPTION

*CONSTRAINED_JOINT_USER_FORCE

*CONSTRAINED_LAGRANGE_IN_SOLID

*CONSTRAINED_LINEAR_GLOBAL

*CONSTRAINED_LINEAR_LOCAL

*CONSTRAINED_LOCAL

*CONSTRAINED_MULTIPLE_GLOBAL

CONSTRAINED_NODAL_RIGID_BODY_*{OPTION}_{OPTION}

*CONSTRAINED_NODE_INTERPOLATION

CONSTRAINED_NODE_SET_*{OPTION}

*CONSTRAINED_NODE_TO_NURBS_PATCH

*CONSTRAINED_POINTS

*CONSTRAINED_RIGID_BODIES

*CONSTRAINED_RIGID_BODY_INSERT

*CONSTRAINED_RIGID_BODY_STOPPERS

CONSTRAINED_RIVET_*{OPTION}

*CONSTRAINED_SHELL_TO_SOLID

*CONSTRAINED_SPLINE

*CONSTRAINED_SPOTWELD_*{OPTION}*_*{OPTION}*

*CONSTRAINED_SPR2

*CONSTRAINED_TIEBREAK

*CONSTRAINED_TIED_NODES_FAILURE

*CONSTRAINED_ADAPTIVITY

Purpose: Constrains a node to the midpoint along an edge of an element. This keyword is automatically created by LS-DYNA during an h-adaptive simulation involving 3-D shells.

Card 1	1	2	3	4	5	6	7	8
Variable	SN	MN1	MN2					
Type	I	I	I					
Default	none	none	none					

VARIABLE	DESCRIPTION
SN	Slave node. This is the node constrained at the midpoint of an edge of an element.
MN1	The node at one end of an element edge.
MN2	The node at the other end of that same element edge.

CONSTRAINED_BEAM_IN_SOLID_*{OPTION}

Available options include:

<BLANK>

ID

TITLE

Purpose: This keyword constrains beam structures to move with Lagrangian solids/thick shells, which serve as the master component. This keyword constrains both acceleration and velocity. This feature is intended to sidestep certain limitations in the CTYPE = 2 implementation in *CONSTRAINED_LAGRANGE_IN_SOLID. Notable features of this keyword include:

1. **CDIR = 1 feature.** With the CDIR = 1 option coupling occurs only in the normal directions. This coupling allows for releasing the constraints along beam axial direction.

2. **Axial coupling force.** Debonding processes can be modeled with a user defined function or user provided subroutine giving the axial shear force based on the slip between rebar nodes and concrete solid elements. This feature is invoked by setting AXFOR flag to a negative integer which refers to the *DEFINE_FUNCTION ID or a number greater than 1000. In the latter case, first we need to modify the subroutine rebar_bondslip_get_getforce() in dyn21.F to add in one or more debonding laws; each tagged with a “lawid”. Then we could specify which debonding law to use with the AXFOR flag. AXFOR value greater than 1000 will call the user subroutine and pass AXFOR in as “lawid”. CDIR has to be set to 1 in this case to release the axial constraints.

3. **NCOUP feature.** Coupling not only at nodes, but also at multiple coupling points in between the two beam element nodes. Please note, the previous implementation done in *CONSTRAINED_LAGRANGE_IN_SOLID CTYPE 2 causes errors in energy balance.

4. **Tetrahedral and pentahedra solid elements are supported.** They are treated as degenerated hexahedra in CTYPE2 implementation.

5. **Velocity/Fixed boundary condition.** The CTYPE 2 implementation failed to constrain beam nodes that were buried inside elements whose nodes had velocity/fixed boundary conditions prescribed.

6. **Optimized Sorting.** Sorting subroutine is optimized for larger problems to achieve better performance and less memory usage.

If a title is not defined, LS-DYNA will automatically create an internal title for this coupling definition.

Title Card. Additional card for TITLE and ID keyword options.

Title	1	2	3	4	5	6	7	8
Variable	COUPID	TITLE						
Type	I	A70						

Card 1	1	2	3	4	5	6	7	8
Variable	SLAVE	MASTER	SSTYP	MSTYP			NCOUP	CDIR
Type	I	I	I	I			I	I
Default	none	none	0	0			0	0

Card 2	1	2	3	4	5	6	7	8
Variable	START	END		AXFOR				
Type	F	F		I				
Default	0	10^{10}		0				

VARIABLE	DESCRIPTION
COUPID	Coupling (card) ID number (I10). If not defined, LS-DYNA will assign an internal coupling ID based on the order of appearance in the input deck.
TITLE	A description of this coupling definition.
SLAVE	Slave set ID defining a part, part set ID of the Lagrangian beam structure (see *PART, *SET_PART).

VARIABLE	DESCRIPTION
MASTER	Master set ID defining a part or part set ID of the Lagrangian solid elements or thick shell elements (see *PART or *SET_PART).
SSTYP	Slave set type of "SLAVE": EQ.0: part set ID (PSID). EQ.1: part ID (PID).
MSTYP	Master set type of "MASTER": EQ.0: part set ID (PSID). EQ.1: part ID (PID).
NCOUP	Number of coupling points generated in one beam element. If set to 0, coupling only happens at beam nodes. Otherwise, coupling is done at both the beam nodes and those automatically generated coupling points.
CDIR	Coupling direction. EQ.0: default, constraint applied along all directions. EQ.1: Constraint only applied along normal directions; along the beam axial direction there is no constraint.
START	Start time for coupling.
END	End time for coupling.
AXFOR	ID of a user defined function describes coupling force versus slip along beam axial direction. GE.0: OFF EQ.-n: n is the function ID in *DEFINE_FUNCTION EQ.n > 1000: n is the debonding law id "lawid" in user defined subroutine rebar_bondslip_get_force().

Example:

1. The example below shows how to define a function and use it to prescribe the debonding process. User can define his own function based on different debonding theories.

```
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|...
.8
```

```
*CONSTRAINED_BEAM_IN_SOLID
$#    slave     master       sstyp      mstyp      ctype       empty       nquad
idir
         2           1           1          1           0                      2
1
$#   start        end                axfor
     0.000     0.000                  -10
*DEFINE_FUNCTION
        10
float force(float slip,float leng)
{
float force,pi,d,area,shear,pf;
pi = 3.1415926;
d = 0.175;
area = pi*d*leng;
pf = 1.0;
if (slip < 0.25) {
shear = slip*pf;
} else {
shear = 0.25*pf;
}
force = shear*area;
return force;
}
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|...
.8
```

2. The example below shows how to define a user subroutine and use it to prescribe the debonding process.

```
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|...
.8
*CONSTRAINED_BEAM_IN_SOLID
$#    slave     master       sstyp      mstyp      ctype       empty       nquad
idir
         2           1           1          1           0                      2
1
$#   start        end                axfor
     0.000     0.000                 1001
*CONSTRAINED_BEAM_IN_SOLID
$#    slave     master       sstyp      mstyp      ctype       empty       nquad
idir
         3           1           1          1           0                      2
1
$#   start        end                axfor
     0.000     0.000                 1002
*USER_LOADING
$     parm1      parm2       parm3      parm4       parm5      parm6       parm7
parm8
       1.0       6.0
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|...
.8
```

And the user debonding law subroutine:

```
      subroutine rebar_bondslip_get_force(slip,dl,force,hsv,
     .  userparm,lawid)
      real hsv
      dimension hsv(12),cm(8),userparm(*)
c
c in this subroutine user defines debonding properties and
c call his own debonding subroutine to get force
      cm(1)=userparm(1)
      cm(2)=userparm(2)
      cm(3)=2.4*(cm(2)/5.0)**0.75
      cm(8)=0.
c
      pi = 3.1415926
      d = 0.175
      area = pi*0.25*d*d*dl
      pf = 1.0
c
      if (lawid.eq.1001) then
        if (slip.lt.0.25) then
          shear = slip*pf
        else
          shear = 0.25*pf
        endif
        force = sign(1.0,slip)*shear*area
      elseif (lawid.eq.1002) then
        if (slip.lt.0.125) then
          shear = slip*pf
        else
          shear = 0.125*pf
        endif
      endif
      return
      end
```

*CONSTRAINED_BUTT_WELD

Purpose: Define a line of coincident nodes that represent a structural butt weld between two parts defined by shell elements. Failure is based on nodal plastic strain for ductile failure and stress resultants for brittle failure. This input is much simpler than the alternative approach for defining butt welds, see *CONSTRAINED_GENERALIZED_WELD_BUTT. The local coordinate system, the effective length, and thickness for each pair of butt welded nodes are determined automatically in the definition below. In the GENERALIZED option these quantities must be defined in the input.

Card 1	1	2	3	4	5	6	7	8
Variable	SNSID	MNSID	EPPF	SIGF	BETA			
Type	I	I	F	F	F			
Default	none	none	0.	1.e+16	1.0			
Remarks		1, 2	3, 4	3	3			

VARIABLE	DESCRIPTION
SNSID	Slave node set ID, see *SET_NODE_*OPTION*.
MNSID	Master node set ID, see *SET_NODE_*OPTION*.
EPPF	Plastic strain at failure
SIGF	σ_f, stress at failure for brittle failure.
BETA	β, failure parameter for brittle failure.

Remarks:

1. Nodes in the master and slave sets must be given in the order they appear as one moves along the edge of the surface. An equal number of coincident nodes must be defined in each set. In a line weld the first and last node in a string of nodes can be repeated in the two sets. If the first and last pair of nodal points are identical, a circular or closed loop butt weld is assumed. See Figure 10-1, where the line butt weld and closed loop weld are illustrated.

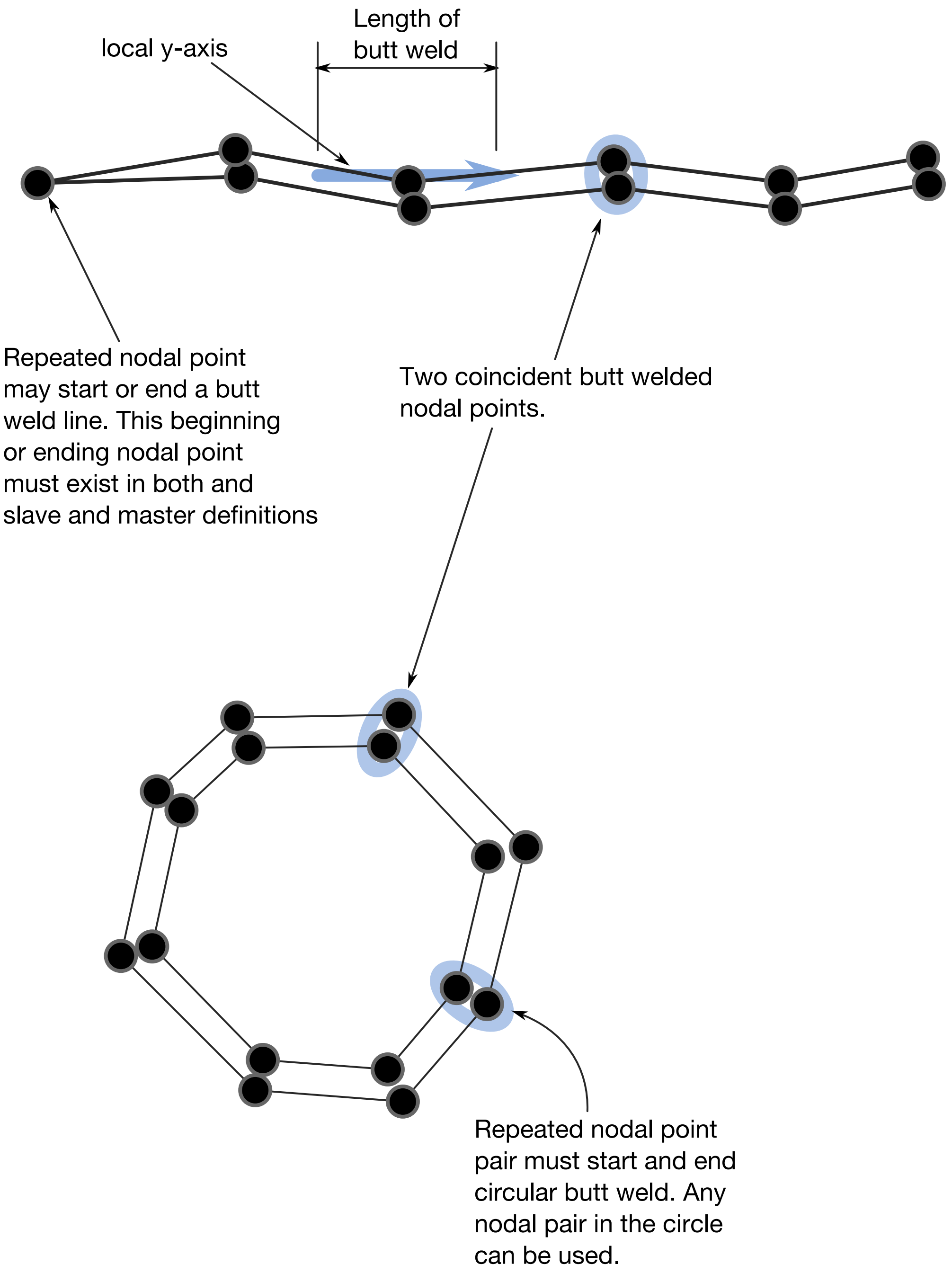

local y-axis
Length of butt weld
Repeated nodal point may start or end a butt weld line. This beginning or ending nodal point must exist in both and slave and master definitions
Two coincident butt welded nodal points.
Repeated nodal point pair must start and end circular butt weld. Any nodal pair in the circle can be used.

Figure 10-1. Definition of butt welds are shown above. The butt weld can be represented by a line of nodal points or by a closed loop

2. Butt welds may not cross. For complicated welds, this option can be combined with the input in *CONSTRAINED_GENERALIZED_WELD_BUTT to handle the case where crossing occurs. Nodes in a butt weld must not be members of rigid bodies.

3. If the average volume-weighted effective plastic strain in the shell elements adjacent to a node exceeds the specified plastic strain at failure, the node is released. Brittle failure of the butt welds occurs when:

$$\beta\sqrt{\sigma_n^2 + 3(\tau_n^2 + \tau_t^2)} \geq \sigma_f$$

where,

σ_n = normal stress (local x)

τ_n = shear stress in direction of weld (local y)

τ_t = shear stress normal to weld (local z)

σ_f = failure stress

β = failure parameter

The component σ_n is nonzero for tensile values only. The nodes defining the slave and master sides of the butt weld must coincide. The local z-axis at a master node is normal to the *master side* plane of the butt weld at the node, and the local y-axis is taken as the vector in the direction of a line connecting the mid-points of the line segments lying on either side of the master node. The normal vector is found by summing the unit normal vectors of all shell elements on the *master side* sharing the butt welded node. The direction of the normal vector at the node is chosen so that the x-local vector points towards the elements on the slave side in order to identify tensile versus compressive stresses. The thickness of the butt weld and length of the butt weld are needed to compute the stress values. The thickness is based on the average thickness of the shell elements that share the butt welded nodal pair, and the chosen length of the butt weld is shown in Figure 10-1.

4. Butt welds may be used to simulate the effect of failure along a predetermined line, such as a seam or structural joint. When the failure criterion is reached at a nodal pair, the nodes begin to separate. As this effect propagates, the weld will appear to “unzip,” thus simulating failure of the connection.

CONSTRAINED_COORDINATE_*{OPTION}

To define constraints based on position coordinates the following options are available:

<BLANK>

LOCAL

Purpose: The keyword is developed to allow the definition of constraints in position coordinates in springback simulation. With the frequent application of adaptive mesh in stamping simulation, nodes needed for springback constraints are often unavailable until the last process simulation before springback is complete. On the other hand, if the nodes are available, their positions may not be exactly on the desired locations required for springback constraints. With this new keyword, the springback simulation is no longer dependent on the previous process simulation results and the exact springback constraint locations can be specified.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	PID	IDIR	X1	Y1	Z1	CID	
Type	I	I	I	F	F	F	I	
Default	none	none	none	0.0	0.0	0.0	0	

VARIABLE	DESCRIPTION
ID	Identification number of a constraint.
PID	Part ID of the part to be constrained.
IDIR	Applicable degrees-of-freedom being constrained: EQ.1: x translational degree-of-freedom, EQ.2: y translational degree-of-freedom, EQ.3: z translational degree-of-freedom.
X1, Y1, Z1	X, Y, Z coordinates of the location being constrained.
CID	Local coordinate system ID.

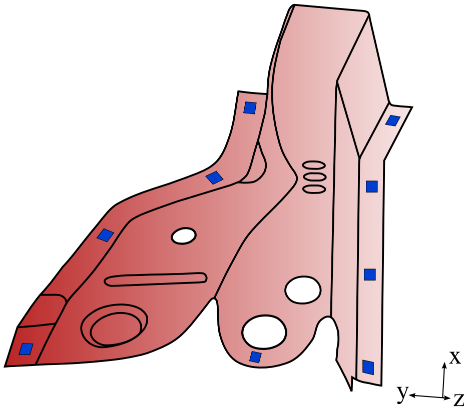


Figure 10-2. Constrained locations of a trim panel (NUMISHEET 2005 cross member).

General remarks:

The identification number of a constraint must be unique; in particular, the IDs must be unique even for two constraints involving the same X, Y, Z coordinates but different degrees of freedom. When the LOCAL option is invoked, a local coordinate system ID, as defined with *DEFINE_COORDINATE_*{OPTION}* keyword, should be provided in the CID field.

Defining constraints using coordinates can now be done in *Springback* process of *LS-PrePost4.0 eZSetup* for metal forming application, using the *Pick location* button (http://ftp.lstc.com/anonymous/outgoing/lsprepost/4.0/metalforming/).

Application example:

An example of using the keyword is listed below. A part with PID 18 is constrained in 6 locations in a local coordinate system ID 9, defined by the keyword *DEFINE_COORDINATE_SYSTEM. Constrained DOFs are indicated by IDIR.

```
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*CONSTRAINED_COORDINATE
$       ID      IDPT      IDIR         x         y         z       CID
         1        18         2  -555.128      86.6   1072.29         9
         2        18         3  -555.128      86.6   1072.29         9
         3        18         3  -580.334    -62.15   1068.32         9
         4        18         1   568.881   81.2945   1033.72         9
         5        18         2   568.881   81.2945   1033.72         9
         6        18         3   568.881   81.2945   1033.74         9
```

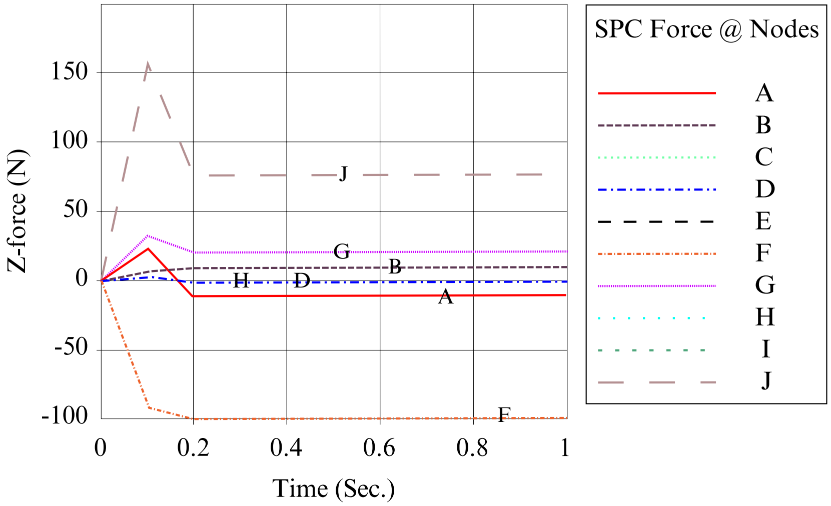


Figure 10-3. SPC Z-forces at 10 nodes.

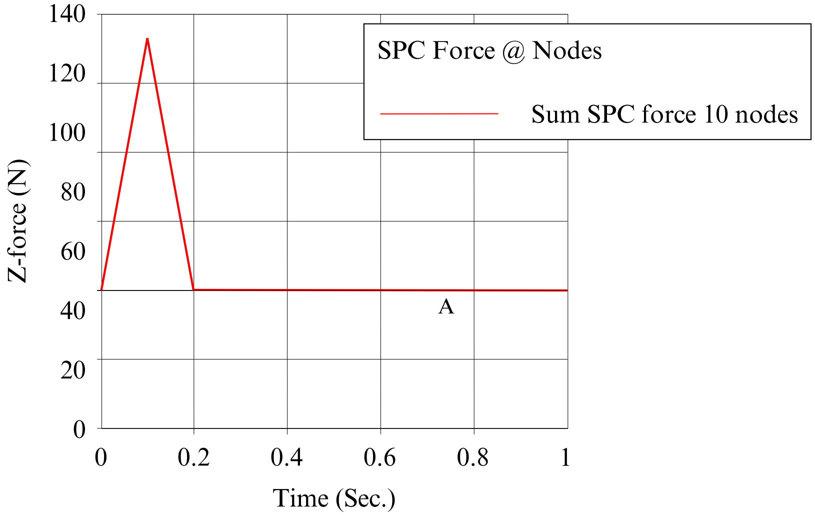


Figure 10-4. SPC Z-force summation of the 10 nodes

```
*DEFINE_COORDINATE_SYSTEM
$       CID        X0        Y0        Z0        XL        YL        ZL
          9       0.0       0.0       0.0       0.0      10.0       0.0
$        XP        YP        ZP
       10.0      10.0       0.0
```

It is possible to output SPC forces on the coordinates constrained. For each position coordinate set, an extra node will be generated and SPC forces are calculated and output to SPCFORC file. The frequency of the output is specified with the keyword *DATABASE_SPCFORC. Shown in the Figure 10-2 are the Z-constrained locations on the trimmed panel (half with symmetric conditions at the smaller end) of the NUMISHEET 2005 cross member. SPC forces in Z direction of these 10 locations were recovered after a multi-steps static implicit springback with this over-constrained boundary condition, Figure 10-3. The summation of these Z-forces is shown in Figure 10-4 and it approaches to zero as the residual stresses are balanced out by the springback shape, absent of gravity.

Revision information:

This feature is now available in LS-DYNA R5 Revision 52619 or later releases. The SPC output feature is available in LS-DYNA Revision 62560 and later releases, both in SMP and MPP.

*CONSTRAINED_EULER_IN_EULER

Purpose: This command defines the coupling interaction between EULERIAN materials in two overlapping, geometrically similar, multi-material Eulerian mesh sets. The command allows a frictionless "contact" between two or more different Eulerian materials.

Card 1	1	2	3	4	5	6	7	8
Variable	PSIDSLV	PSIDMST	PFAC					
Type	I	I	F					
Default	0	0	0.1					

VARIABLE	DESCRIPTION
PSIDSLV	Part set ID of the 1st ALE or Eulerian set of mesh(es) (slave).
PSIDMST	Part set ID of the 2nd ALE or Eulerian set of mesh(es) (master).
PFAC	A penalty factor for the coupling interaction between the two PSIDs.

Remarks:

1. The 2 meshes must be of Eulerian formulation (the meshes are fixed in space, not moving). Consider 2 overlapping Eulerian meshes. Each Eulerian mesh contains 2 physical materials, say a vacuum and a metal. This card provides a frictionless "contact" or interaction between the 2 metals, each resides in a different Eulerian mesh system. Due to its restrictive nature, this option is currently only an experimental feature.

2. Contact pressure is built up in two overlapping Eulerian elements if their combined material fill fraction exceeds 1.0 (penalty formulation).

3. This feature needs to be combined with *MAT_VACUUM (element formulation 11).

Example:

Consider an ALE/Eulerian multi-material model (ELFORM = 11) consisting of:

PID 1 = *MAT_NULL (material 1)

PID 2 = *MAT_VACUUM ⇒ PID 1 is merged at its boundary to PID 2.

PID 3 = *MAT_NULL (material 3)

PID 4 = *MAT_VACUUM ⇒ PID 3 is merged at its boundary to PID 4.

The mesh set containing PID 1 & 2 intersects or overlaps with the mesh set containing PID 3 & 4. PID 1 is given an initial velocity in the positive x direction. This will cause material 1 to contact material 3 (note that materials 2 & 4 are void). The interaction between materials 1 & 3 is possible by defining this coupling command. In this case material 1 can flow within the mesh region of PID 1 & 2 only, and material 3 can flow within the mesh region of PID 3 & 4 only.

```
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|...8
*ALE_MULTI-MATERIAL_GROUP
$      SID   SIDYTPE
         1         1
         2         1
         3         1
         4         1
*CONSTRAINED_EULER_IN_EULER
$    PSID1     PSID2     PENAL
        11        12       0.1
*SET_PART_LIST
        11
         1         2
*SET_PART_LIST
        12
         3         4
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|...8
```

CONSTRAINED_EXTRA_NODES_*OPTION

Available options include:

NODE

SET

Purpose: Define extra nodes for rigid body.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	NID/NSID	IFLAG					
Type	I	I	I					
Default	none	none	0					

VARIABLE	DESCRIPTION
PID	Part ID of rigid body to which the nodes will be added, see *PART.
NID / NSID	Node (keyword option: NODE) or node set ID (keyword option: SET), see *SET_NODE, of added nodes.
IFLAG	This flag is meaningful if and only if the inertia properties of the Part ID are defined in PART_INERTIA. If set to unity, the center-of-gravity, the translational mass, and the inertia matrix of the PID will be updated to reflect the merged nodal masses of the node or node set. If IFLAG is defaulted to zero, the merged nodes will not affect the properties defined in PART_INERTIA since it is assumed the properties already account for merged nodes.

Remarks:

Extra nodes for rigid bodies may be placed anywhere, even outside the body, and they are assumed to be part of the rigid body. They have many uses including:

1. The definition of draw beads in metal forming applications by listing nodes along the draw bead.

2. Placing nodes where joints will be attached between rigid bodies.

3. Defining a node where point loads are to be applied or where springs may be attached.

4. Defining a lumped mass at a particular location.

The coordinates of the extra nodes are updated according to the rigid body motion.

Examples:

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$  *CONSTRAINED_EXTRA_NODES_NODE
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Rigidly attach nodes 285 and 4576 to part 14.  (Part 14 MUST be a rigid body.)
$
*CONSTRAINED_EXTRA_NODES_NODE
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      pid       nid
        14       285
        14      4576
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
```

```
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$  *CONSTRAINED_EXTRA_NODES_SET
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Rigidly attach all nodes in set 4 to part 17.  (Part 17 MUST be a rigid body.)
$
$  In this example, four nodes from a deformable body are attached
$  to rigid body 17 as a means of joining the two parts.
$
*CONSTRAINED_EXTRA_NODES_SET
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      pid      nsid
        17         4
$
$
*SET_NODE_LIST
$      sid
         4
$     nid1      nid2      nid3      nid4      nid5      nid6      nid7      nid8
       665       778       896       827
$
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
```

*CONSTRAINED_GENERALIZED_WELD_*OPTION*_*{OPTION}*

Available options include:

SPOT

FILLET

BUTT

CROSS_FILLET

COMBINED

To define an ID for the weld use the option:

ID

Purpose: Define spot, fillet, butt, and other types of welds. Coincident nodes are permitted if the local coordinate ID is defined. For the spot weld a local coordinate ID is not required if the nodes are offset. Failures can include both the plastic and brittle failures. These can be used either independently or together. Failure occurs when either criteria is met. The welds may undergo large rotations since the equations of rigid body mechanics are used to update their motion. Weld constraints between solid element nodes are not supported.

ID Card. Additional card for ID keyword option.

ID Card	1	2	3	4	5	6	7	8
Variable	WID							
Type	I							
Default	0							

Card 1	1	2	3	4	5	6	7	8
Variable	NSID	CID	FILTER	WINDOW	NPR	NPRT		
Type	I	I	I	E	I	I		
Default	none	none						

VARIABLE	DESCRIPTION
WID	Optional weld ID.
NSID	Nodal set ID, see *SET_NODE_*OPTION*.
CID	Coordinate system ID for output of spot weld data to SWFORC in local system, see *DEFINE_COORDINATE_*OPTION*. CID is not required for spot welds if the nodes are not coincident.
FILTER	Number of force vectors saved for filtering. This option can eliminate spurious failures due to numerical force spikes; however, memory requirements are significant since 6 force components are stored with each vector. LE.1: no filtering GE.2: simple average of force components divided by FILTER or the maximum number of force vectors that are stored for the time window option below.
WINDOW	Time window for filtering. This option requires the specification of the maximum number of steps which can occur within the filtering time window. If the time step decreases too far, then the filtering time window will be ignored and the simple average is used. EQ.0: time window is not used
NPR	Number of individual nodal pairs in the cross fillet or combined general weld.

VARIABLE	DESCRIPTION
NPRT	Print option in file rbdout. EQ.0: default from the control card, *CONTROL_OUTPUT, is used, see variable name IPRTF. EQ.1: data is printed EQ.2: data is not printed

Spot Weld Card. Additional Card required SPOT keyword option.

Card 2	1	2	3	4	5	6	7	8
Variable	TFAIL	EPSF	SN	SS	N	M		
Type	F	F	F	F	F	F		

VARIABLE	DESCRIPTION
TFAIL	Failure time for constraint set, t_f . (default = 1.E+20)
EPSF	Effective plastic strain at failure, $\epsilon^p_{\mathrm{fail}}$ defines ductile failure.
SN	S_n, normal force at failure, only for the brittle failure of spot welds.
SS	S_s, shear force at failure, only for the brittle failure of spot welds.
N	n, exponent for normal force, only for the brittle failure of spot welds.
M	m, exponent for shear force, only for the brittle failure of spot welds.

Remarks:

Spot weld failure due to plastic straining occurs when the effective nodal plastic strain exceeds the input value, $\varepsilon^p_{\mathrm{fail}}$. This option can model the tearing out of a spot weld from the sheet metal since the plasticity is in the material that surrounds the spot weld, not the spot weld itself. A least squares algorithm is used to generate the nodal values of plastic strains at the nodes from the element integration point values. The plastic strain is integrated through the element and the average value is projected to the nodes via a

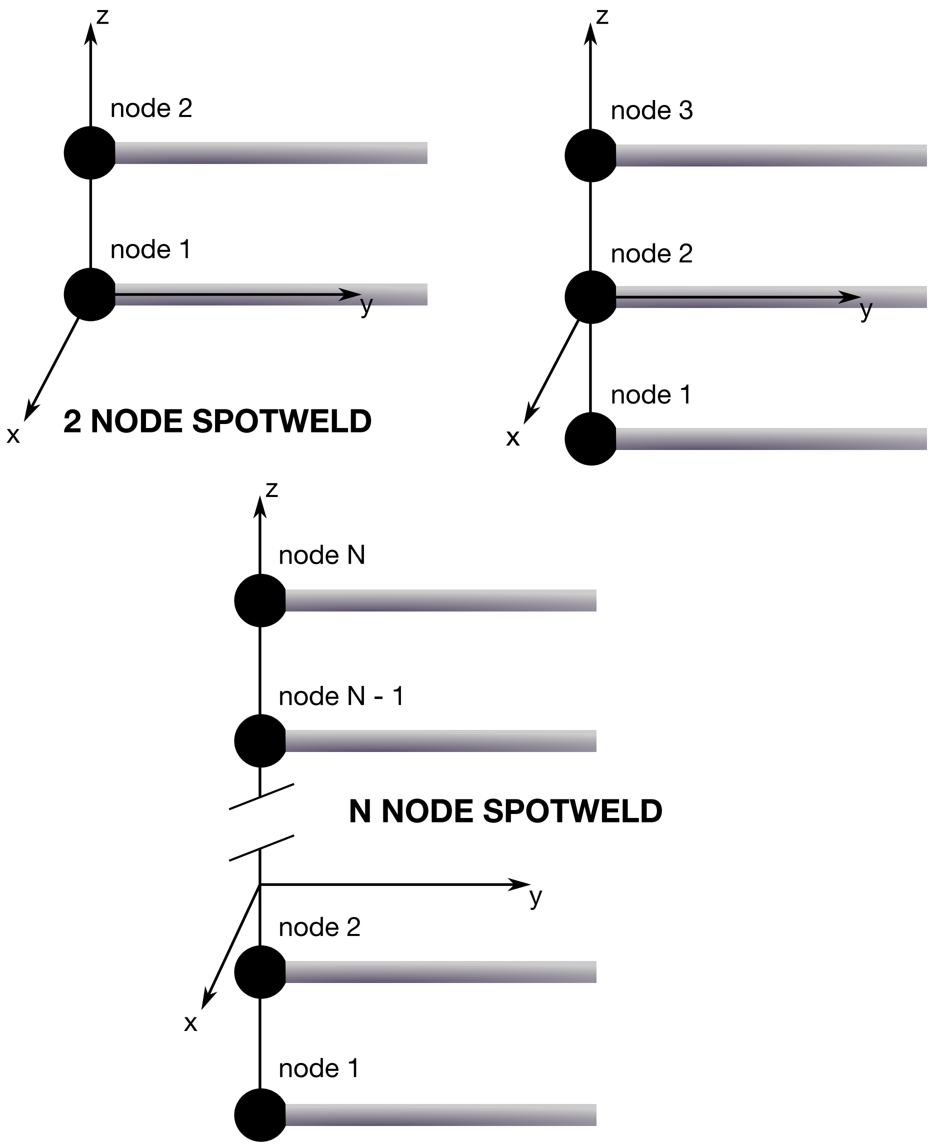


Figure 10-5. Nodal ordering and orientation of the local coordinate system is important for determining spotweld failure.

least square fit. This option should only be used for the material models related to metallic plasticity and can result in slightly increased run times.

Brittle failure of the spot welds occurs when:

$$\left[\frac{\max(f_n,0)}{S_n}\right]^n + \left[\frac{|f_s|}{S_s}\right]^m \ge 1$$

where f_n and f_s are the normal and shear interface force. Component f_n contributes for tensile values only. When the failure time, t_f, is reached the nodal rigid body becomes inactive and the constrained nodes may move freely. In Figure 10-5 the ordering of the nodes is shown for the 2 node and 3 node spot welds. This order is with respect to the local coordinate system where the local z-axis determines the tensile direction. The nodes in the spot weld may coincide. The failure of the 3 node spot weld may occur gradually with first one node failing and later the second node may fail. For n noded spot welds the failure is progressive starting with the outer nodes (1 and n) and then moving inward to nodes 2 and n - 1. Progressive failure is necessary to preclude failures that would create new rigid bodies.

Fillet Weld Card. Additional Card required for the FILLET keyword option.

Card 2	1	2	3	4	5	6	7	8
Variable	TFAIL	EPSF	SIGF	BETA	L	W	A	ALPHA
Type	F	F	F	F	F	F	F	F

VARIABLE	DESCRIPTION
TFAIL	Failure time for constraint set, t_f (default = 1.E+20).
EPSF	Effective plastic strain at failure, ϵ_{fail}^p defines ductile failure.
SIGF	σ_f, stress at failure for brittle failure.
BETA	β, failure parameter for brittle failure.
L	L, length of fillet weld (see Figure 10-6).
W	w, separation of parallel fillet welds (see Figure 10-6).
A	a, fillet weld throat dimension (see Figure 10-6).
ALPHA	α, weld angle (see Figure 10-6) in degrees.

Remarks:

Ductile fillet weld failure, due to plastic straining, is treated identically to spot weld failure. Brittle failure occurs when the following weld stress condition is met on the narrowest fillet weld cross section (across the throat):

$$\beta\sqrt{\sigma_n^2 + 3(\tau_n^2 + \tau_t^2)} \geq \sigma_f$$

Where

$$\sigma_n = \text{normal stress}$$

$$\tau_n = \text{shear stress in local z-x plane}$$

$$\tau_t = \text{shear stress in local-y direction}$$

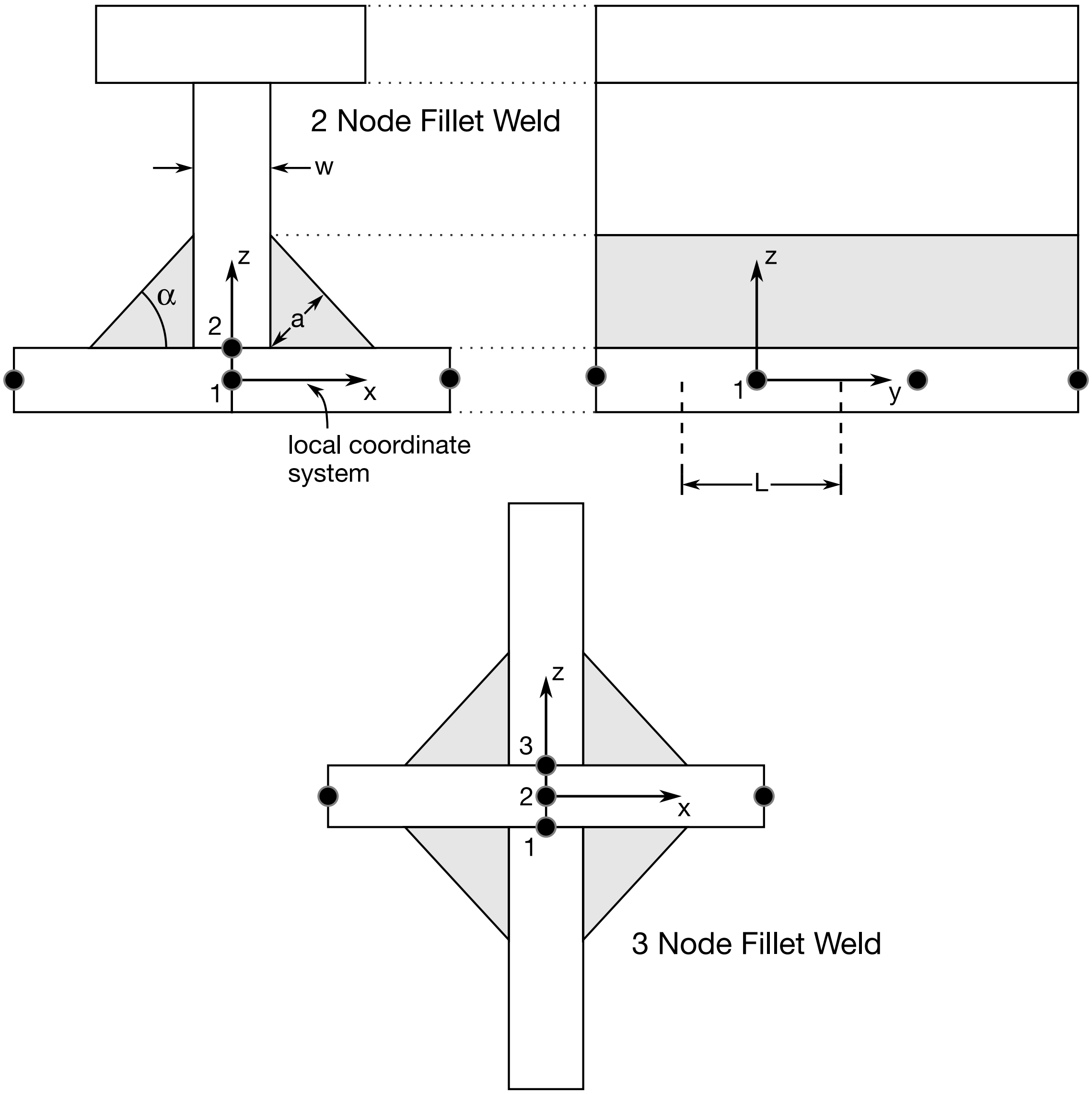


Figure 10-6. Nodal ordering and orientation of the local coordinate system is shown for fillet weld failure. The angle is defined in degrees.

$$\sigma_f = \text{failure stress}$$

$$\beta = \text{failure parameter}$$

The component σ_n is nonzero for tensile values only. When the failure time, t_f , is reached the nodal rigid body becomes inactive and the constrained nodes may move freely. In Figure 10-6 the ordering of the nodes is shown for the 2 node and 3 node fillet welds. This order is with respect to the local coordinate system where the local z axis determines the tensile direction. The initial orientation of the local coordinate system is defined by CID. If CID = 0 then the global coordinate system is used. The local coordinate system is updated according to the rotation of the rigid body representing the weld. The failure of the 3 node fillet weld may occur gradually with first one node failing and later the second node may fail.

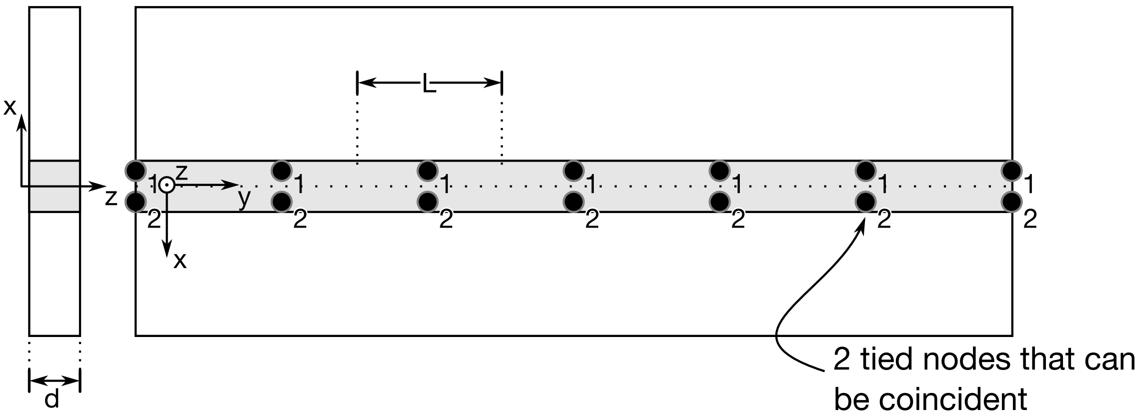


Figure 10-7. Orientation of the local coordinate system and nodal ordering is shown for butt weld failure.

Butt Weld Card. Additional Card required for the BUTT keyword option.

Card 2	1	2	3	4	5	6		8
Variable	TFAIL	EPSF	SIGY	BETA	L	D		
Type	F	F	F	F	F	F		

VARIABLE	DESCRIPTION
TFAIL	Failure time for constraint set, t_f. (default = 1.E+20)
EPSF	Effective plastic strain at failure, ϵ_{fail}^p defines ductile failure.
SIGY	σ_f, stress at failure for brittle failure.
BETA	β, failure parameter for brittle failure.
L	L, length of butt weld (see Figure 10-7).
D	d, thickness of butt weld (see Figure 10-7).

Remarks:

Ductile butt weld failure, due to plastic straining, is treated identically to spot weld failure. Brittle failure of the butt welds occurs when:

$$\beta\sqrt{\sigma_n^2 + 3(\tau_n^2 + \tau_t^2)} \geq \sigma_f$$

where

σ_n = normal stress

τ_n = shear stress in direction of weld (local y)

τ_t = shear stress normal to weld (local z)

σ_f = failure stress

β = failure parameter

The component σ_n is nonzero for tensile values only. When the failure time, t_f , is reached the nodal rigid body becomes inactive and the constrained nodes may move freely. The nodes in the butt weld may coincide.

Example:

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$  *CONSTRAINED_GENERALIZED_WELD_BUTT
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Weld two plates that butt up against each other at three nodal pair
$  locations.  The nodal pairs are 32-33, 34-35 and 36-37.
$
$  This requires 3 separate *CONSTRAINED_GENERALIZED_WELD_BUTT definitions,
$  one for each nodal pair.  Each weld is to have a length (L) = 10,
$  thickness (D) = 2, and a transverse length (Lt) = 1.
$
$  Failure is defined two ways:
$    Ductile failure if effective plastic strain exceeds 0.3
$    Brittle failure if the stress failure criteria exceeds 0.25
$       - scale the brittle failure criteria by beta = 0.9.
$    Note: beta > 1 weakens weld          beta < 1 strengthens weld
$
*CONSTRAINED_GENERALIZED_WELD_BUTT
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$     nsid       cid
        21
$    tfail      epsf      sigy      beta         L         D        Lt
               0.3     0.250       0.9      10.0       2.0       1.0
$
$
*CONSTRAINED_GENERALIZED_WELD_BUTT
$     nsid       cid
        23
$    tfail      epsf      sigy      beta         L         D        Lt
               0.3     0.250       0.9      10.0       2.0       1.0
$
$
*CONSTRAINED_GENERALIZED_WELD_BUTT
$     nsid       cid
        25
$    tfail      epsf      sigy      beta         L         D        Lt
               0.3     0.250       0.9      10.0       2.0       1.0
$
$
*SET_NODE_LIST
$      sid
        21
```

```
$     nid1      nid2
        32        33
*SET_NODE_LIST
        23
        34        35
*SET_NODE_LIST
        25
        36        37
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

Cross Fillet Weld Card. Additional Card for the CROSS_FILLET keyword option.

Card 2	1	2	3	4	5	6	7	8
Variable	TFAIL	EPSF	SIGY	BETA	L	W	A	ALPHA
Type	F	F	F	F	F	F	F	F

Nodal Pair Cards. Read NPR additional cards for the CROSS_FILLET keyword option.

Card 3	1	2	3	4	5	6	7	8
Variable	NODEA	NODEB	NCID					
Type	I	I	I					

VARIABLE	DESCRIPTION
TFAIL	Failure time for constraint set, t_f . (default = 1.E+20)
EPSF	Effective plastic strain at failure, ϵ^p_{fail} defines ductile failure.
SIGY	σ_f, stress at failure for brittle failure.
BETA	β, failure parameter for brittle failure.
L	L, length of fillet weld (see Figure 10-6).
W	w, separation of parallel fillet welds (see Figure 10-6).
A	a, throat dimension of fillet weld (see Figure 10-6).
ALPHA	α, weld angle (see Figure 10-6) in degrees.

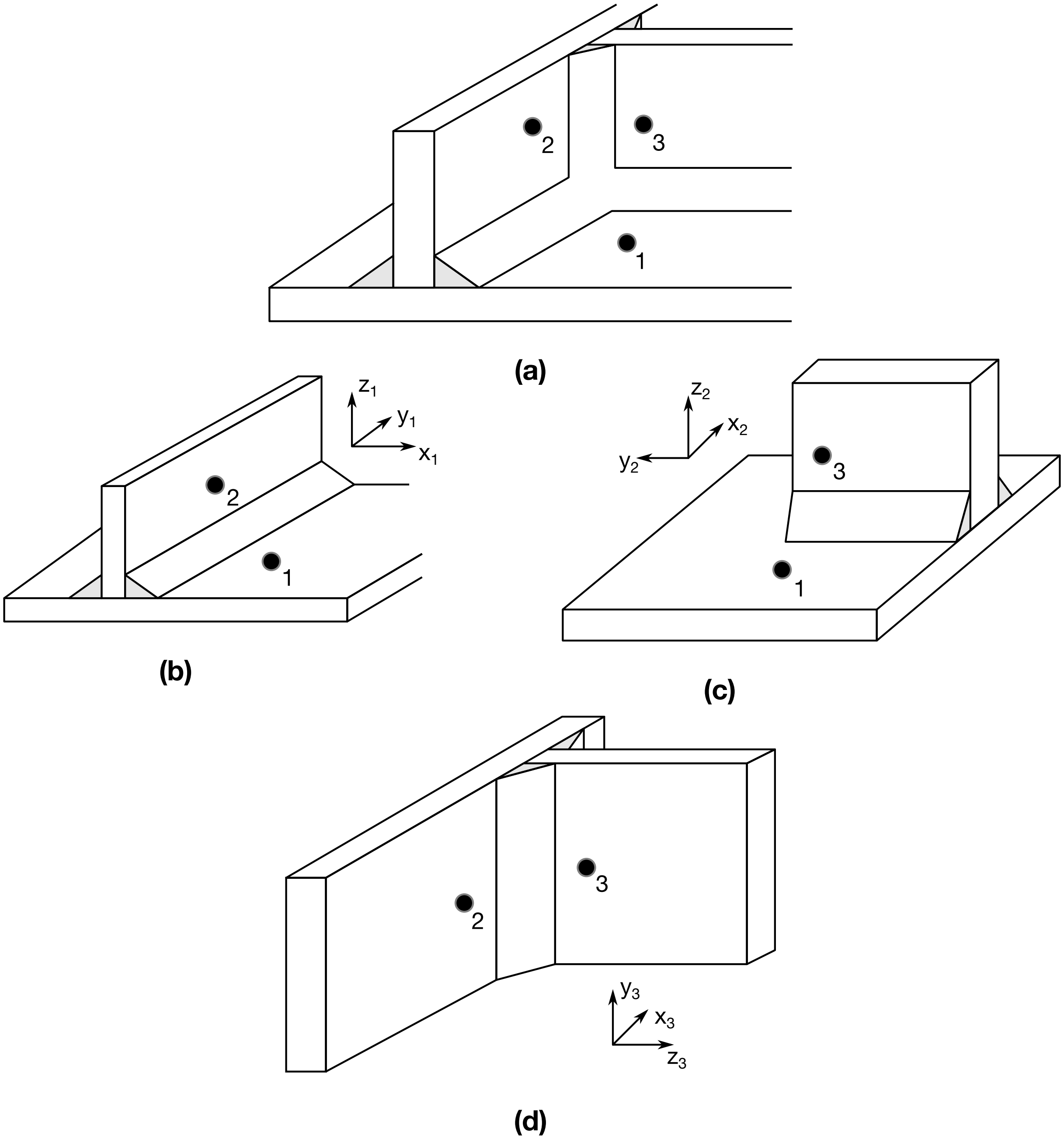


Figure 10-8. A simple cross fillet weld illustrates the required input. Here NPR = 3 with nodal pairs (A = 2, B = 1), (A = 3, B = 1), and (A = 3, B = 2). The local coordinate axes are shown. These axes are fixed in the rigid body and are referenced to the local rigid body coordinate system which tracks the rigid body rotation.

VARIABLE	DESCRIPTION
NODEA	Node ID, A, in weld pair (CROSS or COMBINED option only). See Figure 10-8.

VARIABLE	DESCRIPTION
NODEB	Node ID, B, in weld pair (CROSS or COMBINED option only).
NCID	Local coordinate system ID (CROSS or COMBINED option only).

Combined Weld Cards:

Additional cards for the COMBINED keyword option. Read in NPR pairs of Cards 2 and 3 for a total of 2 × NPR cards.

Card 2	1	2	3	4	5	6	7	8
Variable	TFAIL	EPSF	SIGY	BETA	L	W	A	ALPHA
Type	F	F	F	F	F	F	F	F

Card 3	1	2	3	4	5	6	7	8
Variable	NODEA	NODEB	NCID	WTYP				
Type	I	I	I	I				

VARIABLE	DESCRIPTION
TFAIL	Failure time for constraint set, t_f. (default = 1.E+20)
EPSF	Effective plastic strain at failure, $\epsilon^p_{\mathrm{fail}}$ defines ductile failure.
SIGY	σ_f, stress at failure for brittle failure.
BETA	β, failure parameter for brittle failure.
L	L, length of fillet/butt weld (see Figure 10-6 and 10-7).
W	w, width of flange (see Figure 10-6).
A	a, width of fillet weld (see Figure 10-6).
ALPHA	α, weld angle (see Figure 10-6) in degrees.

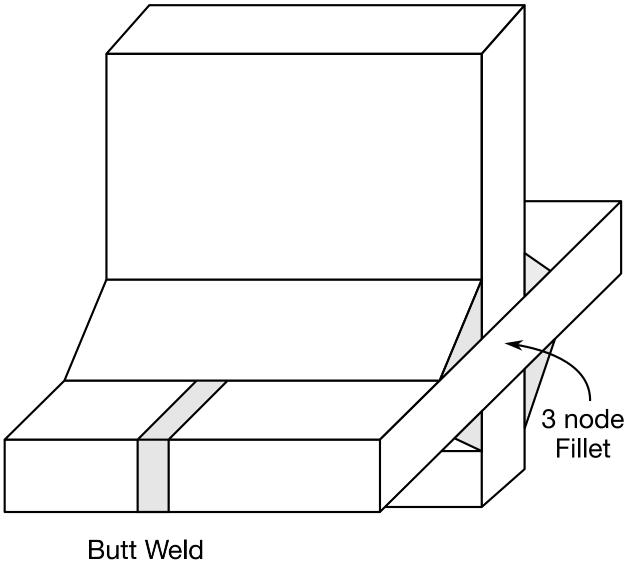


Figure 10-9. A combined weld is a mixture of fillet and butt welds.

VARIABLE	DESCRIPTION
NODEA	Node ID, A, in weld pair (CROSS or COMBINED option only).
NODEB	Node ID, B, in weld pair (CROSS or COMBINED option only).
NCID	Local coordinate system ID (CROSS or COMBINED option only).
WTYPE	Weld pair type (GENERAL option only). See Figure 10-9. EQ.0: fillet weld EQ.1: butt weld

*CONSTRAINED_GLOBAL

Purpose: Define a global boundary constraint plane.

Card 1	1	2	3	4	5	6	7	8
Variable	TC	RC	DIR	X	Y	Z		
Type	I	I	I	F	F	F		
Default	0	0	0	0	0	0		

VARIABLE	DESCRIPTION
TC	Translational Constraint: EQ.1: constrained x translation, EQ.2: constrained y translation, EQ.3: constrained z translation, EQ.4: constrained x and y translations, EQ.5: constrained y and z translations, EQ.6: constrained x and z translations, EQ.7: constrained x, y, and z translations,
RC	Rotational Constraint: EQ.1: constrained x-rotation, EQ.2: constrained y-rotation, EQ.3: constrained z-rotation, EQ.4: constrained x and y rotations, EQ.5: constrained y and z rotations, EQ.6: constrained z and x rotations, EQ.7: constrained x, y, and z rotations.

VARIABLE	DESCRIPTION
DIR	Direction of normal for constraint plane. EQ.1: global x, EQ.2: global y, EQ.3: global z.
X	Global x-coordinate of a point on the constraint plane.
Y	Global y-coordinate of a point on the constraint plane.
Z	Global z-coordinate of a point on the constraint plane.

Remarks:

Nodes within a mesh-size-dependent tolerance are constrained on a global plane. This option is recommended for use with r-method adaptive remeshing where nodal constraints are lost during the remeshing phase. See *CONSTRAINED_LOCAL for specifying constraints to nodes lying on a local plane.

CONSTRAINED_INTERPOLATION_*{OPTION}

Available options include:

<BLANK>

LOCAL

Purpose: Define an interpolation constraint. With this constraint type, the motion of a single dependent node is interpolated from the motion of a set of independent nodes.

This option is useful for the redistribution of a load applied to the dependent node by the surrounding independent nodes. This load may be a translational force or a rotational moment. This keyword is typically used to model shell-brick and beam-brick interfaces.

The mass and rotary inertia of the dependent nodal point is also redistributed. This constraint is applied in the global coordinate system unless the option LOCAL is active. *One *CONSTRAINED_INTERPOLATION card is required for each constraint definition.* The input list of independent nodes is terminated when the next "*" card is found. In explicit calculations the independent nodes cannot be dependent nodes in other constraints such as nodal rigid bodies; however, implicit calculations are not bound by this limitation.

Card 1	1	2	3	4	5	6	7	8
Variable	ICID	DNID	DDOF	CIDD	ITYP			
Type	I	I	I	I	I			
Default	0	0	123456	optional	0			

Independent Node Card Sets:

If LOCAL option is not set, for each independent node include the following card; if the LOCAL keyword option is set, include only the following *pair* of cards. This input is terminated at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	INID	IDOF	TWGHTX	TWGHTY	TWGHTZ	RWGHTX	RWGHTY	RWGHTZ
Type	I	I	F	F	F	F	F	F
Default	0	123456	1.0	TWGHTX	TWGHTX	TWGHTX	TWGHTX	TWGHTX

Local Coordinate Card. Additional card for the LOCAL keyword option to be paired with card 2.

Card 3	1	2	3	4	5	6	7	8
Variable	CIDI							
Type	I							
Default	0							

VARIABLE **DESCRIPTION**

ICID Interpolation constraint ID.

DNID Dependent node ID. This node should not be a member of a rigid body, or elsewhere constrained in the input.

DDOF Dependent degrees-of-freedom. The list of dependent degrees-of-freedom consists of a number with up to six digits, with each digit representing a degree of freedom. For example, the value 1356 indicates that degrees of freedom 1, 3, 5, and 6 are controlled by the constraint. The default is 123456. Digit: degree of freedom ID's:

EQ.1: x

EQ.2: y

VARIABLE DESCRIPTION

EQ.3: z

EQ.4: rotation about x axis

EQ.5: rotation about y axis

EQ.6: rotation about z axis

CIDD

Local coordinate system ID if LOCAL option is active. If blank the global coordinate system is assumed.

ITYP

Specifies the meaning of INID.

EQ.0: INID is a node ID

EQ.1: INID is a node set ID

INID

Independent node ID or node set ID.

IDOF

Independent degrees-of-freedom using the same form as for the dependent degrees-of-freedom, DDOF, above.

TWGHTX

Weighting factor for node INID with active degrees-of-freedom IDOF. This weight scales the x-translational component. It is normally sufficient to define only TWGHTX even if its degree-of-freedom is inactive since the other factors are set equal to this input value as the default. There is no requirement on the values that are chosen as the weighting factors, i.e., that they sum to unity. The default value for the weighting factor is unity.

TWGHTY

Weighting factor for node INID with active degrees-of-freedom IDOF. This weight scales the y-translational component.

TWGHTZ

Weighting factor for node INID with active degrees-of-freedom IDOF. This weight scales the z-translational component.

RWGHTX

Weighting factor for node INID with active degrees-of-freedom IDOF. This weight scales the x-rotational component.

RWGHTY

Weighting factor for node INID with active degrees-of-freedom IDOF. This weight scales the y-rotational component.

RWGHTZ

Weighting factor for node INID with active degrees-of-freedom IDOF. This weight scales the z-rotational component.

CIDI

Local coordinate system ID if LOCAL option is active. If blank the global coordinate system is assumed.

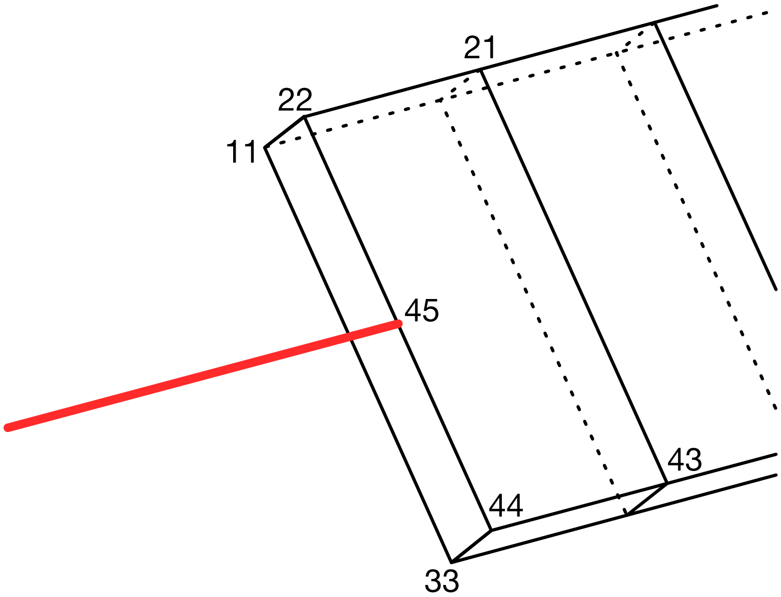


Figure 10-10. Illustration of Example 1.

Example 1:

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$  *CONSTRAINED_INTERPOLATION  (Beam to solid coupling)
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Tie a beam element to a solid element.
$
$  The node of the beam to be tied does not share a common node with the solids.
$  If the beam node is shared, for example, then set ddof=456.
$
*CONSTRAINED_INTERPOLATION
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$     icid      dnid      ddof
         1        45    123456
$     inid      idof    twghtx    twghty    twghtz    rwghtx    rwghty    rwghtz
        22       123
        44       123
        43       123
$
*.........
$
```

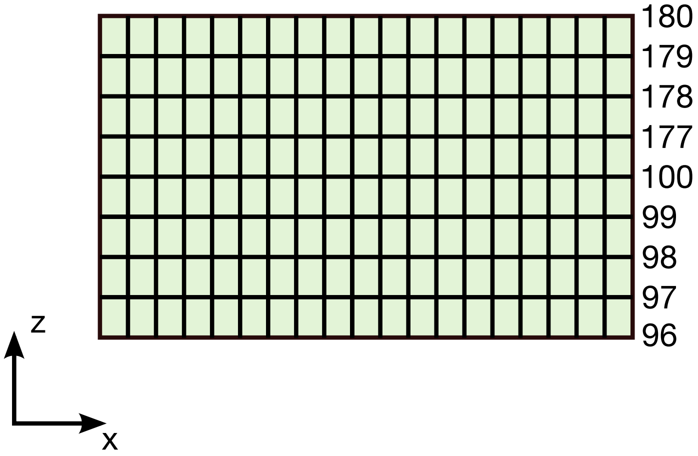


Figure 10-11. Illustration of Example 2.

Example 2:

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$  *CONSTRAINED_INTERPOLATION  (Load redistribution)
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Moment about normal axis of node 100 is converted to an equivalent load by
$  applying x-force resultants to the nodes lying along the right boundary
$
*DEFINE_CURVE
1,0,0.,0.,0.,0.,0
0.,0.
.1,10000.
*LOAD_NODE_POINT
100,6,1,1.0
$
*CONSTRAINED_INTERPOLATION
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$     icid      dnid      ddof
         1       100         5
$     inid      idof    twghtx    twghty    twghtz    rwghtx    rwghty    rwghtz
        96         1
        97         1
        98         1
        99         1
       177         1
       178         1
       179         1
       180         1
$
*.........
$
```

*CONSTRAINED_INTERPOLATION_SPOTWELD

(prior notation *CONSTRAINED_SPR3 still works)

Purpose: Define a spotweld with failure. This model includes a plasticity-damage model that reduces the force and moment resultants to zero as the spotweld fails. The location of the spotweld is defined by a single node at the center of two connected sheets. The domain of influence is specified by a radius, which should be approximately equal to the spotweld's radius. The algorithm does a normal projection from the two sheets to the spotweld node and locates all nodes within the user-defined diameter of influence. The numerical implementation of this model is similar to the SPR2 model (*CONSTRAINED_SPR2).

Card 1	1	2	3	4	5	6	7	8
Variable	PID1	PID2	NSID	THICK	R	STIFF	ALPHA1	MODEL
Type	I	I	I	F	F	F	F	F
Default	none	none	none	none	none	none	none	1.0

Card 2	1	2	3	4	5	6	7	8
Variable	RN	RS	BETA	LCF	LCUPF	LCUPR	DENS	INTP
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

Additional Card for MODEL = 2, 12, or 22.

Card 3	1	2	3	4	5	6	7	8
Variable	UPFN	UPFS	ALPHA2	BETA2	UPRN	UPRS	ALPHA3	BETA3
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

Additional Card for MODEL = 2, 12, or 22.

Card 4	1	2	3	4	5	6	7	8
Variable	MRN	MRS						
Type	F	F						
Default	none	none						

VARIABLE	DESCRIPTION
PID1	Part ID of first sheet.
PID2	Part ID of second sheet.
NSID	Node set ID of spotweld location nodes.
THICK	Total thickness of both sheets.
R	Spotweld radius.
STIFF	Elastic stiffness. Function ID if MODEL > 10.
ALPHA1	Scaling factor α_1. Function ID if MODEL > 10.
MODEL	Material behavior and damage model, see remarks. EQ. 1: SPR3 (default), EQ. 2: SPR4, EQ.11: same as 1 with selected material parameters as functions, EQ.12: same as 2 with selected material parameters as functions, EQ.21: same as 11 with slight modification, see remarks, EQ.22: same as 12 with slight modification, see remarks.
RN	Tensile strength factor. Function ID if MODEL > 10.
RS	Shear strength factor. Function ID if MODEL > 10.
BETA	Exponent for plastic potential β_1. Function ID if MODEL > 10.

VARIABLE	DESCRIPTION
LCF	Load curve ID describing force versus plastic displacement: $F^0(\bar{u}^{pl})$.
LCUPF	Load curve ID describing plastic initiation displacement versus mode mixity: $\bar{u}_0^{pl}(\kappa)$. Only for MODEL = 1, 11, or 21.
LCUPR	Load curve ID describing plastic rupture displacement versus mode mixity: $\bar{u}_f^{pl}(\kappa)$. Only for MODEL = 1, 11, or 21.
DENS	Spotweld density (necessary for time step calculation).
INTP	Flag for interpolation. EQ.0: linear (default), EQ.1: uniform, EQ.2: inverse distance weighting.
UPFN	Plastic initiation displacement in normal direction $\bar{u}_{0,\mathrm{ref}}^{pl,n}$.
UPFS	Plastic initiation displacement in shear direction $\bar{u}_{0,\mathrm{ref}}^{pl,s}$.
ALPHA2	Plastic initiation displacement scaling factor α_2 .
BETA2	Exponent for plastic initiation displacement β_2.
UPRN	Plastic rupture displacement in normal direction $\bar{u}_{f,\mathrm{ref}}^{pl,n}$.
UPRS	Plastic rupture displacement in shear direction $\bar{u}_{f,\mathrm{ref}}^{pl,s}$.
ALPHA3	Plastic rupture displacement scaling factor α_3 .
BETA3	Exponent for plastic rupture displacement β_3.
MRN	Proportionality factor for dependency RN.
MRS	Proportionality factor for dependency RS.

Remarks:

When this feature is used, it is recommended to use the drilling rotation constraint method for the connected components in explicit analysis, i.e. parameter DRCPSID of

*CONTROL_SHELL should refer to all shell parts involved in INTERPOLATION_SPOTWELD connections.

MODEL = 1, 11, or 21 ("SPR3")

This numerical model is similar to the self-piercing rivet model SPR2 (see *CONSTRAINED_SPR2) but with some differences to make it more suitable for spotwelds. The first difference is symmetric behavior of the spotweld connection, i.e. there is no distinction between a master sheet and a slave sheet. This is done by averaging the normals of both parts and by always distributing the balance moments equally to both sides.

The second difference is that there are not only two but three quantities to describe the kinematics, namely the normal relative displacement δ_n, the tangential relative displacement δ_t, and the relative rotation ω_b - all with respect to the plane-of-maximum opening. I.e. a relative displacement vector is defined as

$$\mathbf{u} = (\delta_n, \delta_t, \omega_b)$$

The third difference is the underlying material model. With the described kinematic quantities, an elastic effective force vector is computed first:

$$\tilde{\mathbf{f}} = (f_n, f_t, m_b) = \mathrm{STIFF} \times \mathbf{u} = \mathrm{STIFF} \times (\boldsymbol{\delta_n}, \boldsymbol{\delta_t}, \boldsymbol{\omega_b})$$

From that, two resultant forces for normal direction and tangential direction (shear) are computed via

$$F_n = \langle f_n \rangle + \alpha_1 m_b, \qquad F_s = f_t$$

Then, a yield function is defined for plastic behavior

$$\phi(\tilde{\mathbf{f}}, \bar{\boldsymbol{u}}^{\mathrm{pl}}) = P(\tilde{\mathbf{f}}) - F^0(\bar{\boldsymbol{u}}^{\mathrm{pl}}) \le 0$$

with relative plastic displacement $\bar{u}^{pl}$, potential P

$$P(\tilde{\mathbf{f}}) = \left[\left(\frac{F_n}{R_n} \right)^{\boldsymbol{\beta}} + \left(\frac{F_s}{R_s} \right)^{\boldsymbol{\beta}} \right]^{\mathbf{1/\beta}}$$

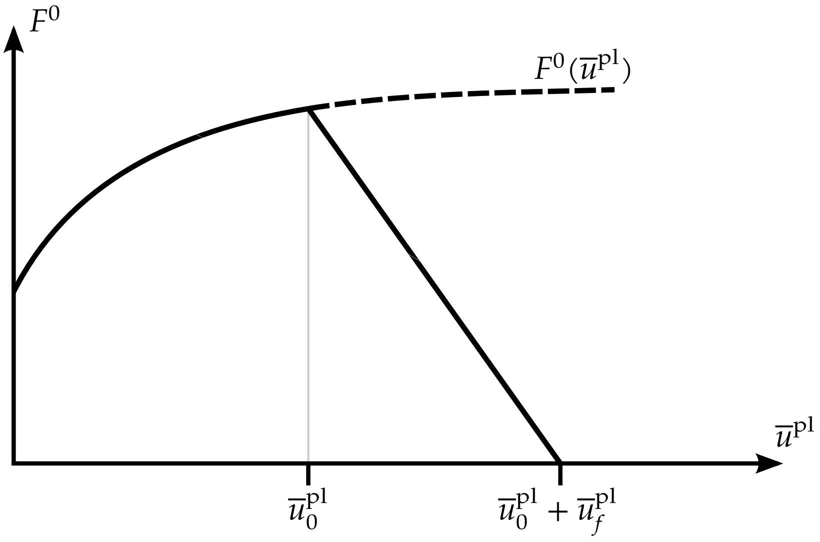


Figure 10-12. Force-displacement curve: plasticity and linear damage

and isotropic hardening described by load curve LCF (see Figure 10-12):

$$F^0 = F^0(\bar{\boldsymbol{u}}^{\mathrm{pl}})$$

In addition, a linear softening evolution is incorporated, where damage is defined as:

$$d = \frac{\bar{u}^{\mathrm{pl}} - \bar{u}_0^{\mathrm{pl}}(\kappa)}{\bar{u}_f^{\mathrm{pl}}(\kappa)}, \qquad 0 < d < 1$$

with mode mixity

$$\kappa = \frac{2}{\pi}\arctan\left(\frac{F_n}{F_s}\right), \qquad 0 < \kappa < 1$$

Finally, the nominal force is computed as:

$$\mathbf{f} = (1 - d)\tilde{\mathbf{f}}$$

<u>MODEL = 2, 12, or 22 (“SPR4”)</u>

In this approach, the relative displacement vector is defined as in model 1

$$\mathbf{u} = (\delta_n, \delta_t)$$

The elastic effective force vector is computed using the elastic stiffness STIFF

$$\tilde{\mathbf{f}} = (f_n, f_t) = \mathrm{STIFF} \times \mathbf{u} = \mathrm{STIFF} \times (\delta_n, \delta_t)$$

A yield function is defined for plastic behavior

$$\phi(\tilde{\mathbf{f}}, \bar{\boldsymbol{u}}^{\boldsymbol{pl}}) = P(\tilde{\mathbf{f}}) - F^0(\bar{\boldsymbol{u}}^{\boldsymbol{pl}}) \le 0$$

with relative plastic displacement $\bar{u}^{pl}$, potential P

$$P(\tilde{\mathbf{f}}) = \left[\left(\frac{f_n}{\tilde{R}_n}\right)^{\beta_1} + \left(\frac{f_t}{\tilde{R}_s}\right)^{\beta}\right]^{1/\beta_1}$$

wherein $\tilde{R}_n$ and $\tilde{R}_s$ represents the load capacity in normal and tangential direction respectively. They are calculated by the values of RN and RS and the influence of relative rotation angle $\omega_b$scaled by ALPHA1

$$\tilde{R}_s = R_s$$

$$\tilde{R}_n = R_n(1 - \alpha_1\ \omega_b)$$

In addition, a linear softening evolution is incorporated, where damage is defined as:

$$d = \frac{\bar{u}^{pl} - \bar{u}_0^{pl}}{\bar{u}_f^{pl}}, \qquad 0 < d < 1$$

The calculation of $\bar{u}_0^{pl}$ and $\bar{u}_f^{pl}$ is done by solving the following equations

$$\left\{\left[\frac{\bar{u}_0^{pl,n}}{\bar{u}_{0,\mathrm{ref}}^{pl,n}(1 - \alpha_2\omega_b)}\right]^{\beta_2} + \left(\frac{\bar{u}_0^{pl,s}}{\bar{u}_{0,ref}^{pl,s}}\right)^{\beta_2}\right\}^{\frac{1}{\beta_2}} - 1 = 0$$

$$\bar{u}_0^{pl,n} = \sin(\varphi)\,\bar{u}_0^{pl}$$

$$\bar{u}_0^{pl,s} = cos(\varphi)\bar{u}_0^{pl}$$

$$\left\{\left[\frac{\bar{u}_f^{pl,n}}{\bar{u}_{f,ref}^{pl,n}(1 - \alpha_3\omega_b)}\right]^{\beta_3} + \left(\frac{\bar{u}_f^{pl,s}}{\bar{u}_{f,ref}^{pl,s}}\right)^{\beta_3}\right\}^{\frac{1}{\beta_3}} - 1 = 0$$

$$\bar{u}_f^{pl,n} = \sin(\varphi)\,\bar{u}_f^{pl}$$

$$\bar{u}_f^{pl,s} = cos(\varphi)\bar{u}_f^{pl}$$

considering the load angle φ

$$\varphi = \arctan\left(\frac{f_n}{f_s}\right)$$

To describe a rate dependent behavior a plastic deformation rate $\dot{\bar{u}}^{pl}$ is defined by

$$\dot{\bar{u}}^{pl} = \frac{\Delta\bar{u}^{pl}}{\Delta t}$$

wherein $\Delta\bar{u}^{pl}$ is the plastic increment in the current time step and Δt is the time step size. If MRN and MRS are defined, the calculation of $\tilde{R}_n$ and $\tilde{R}_s$ is changed to

$$\tilde{R}_n\left(\dot{\bar{u}}^{pl}\right) = \left(R_n + m_{R_n}\dot{\bar{u}}^{pl}\right)(1 - \alpha_1\ \omega_b)$$

$$\tilde{R}_s(\dot{\bar{u}}^{pl}) = R_s + m_{R_s}\dot{\bar{u}}^{pl}$$

A detailed description of the SPR4 approach (MODEL = 2) is given in Bier and Sommer [2013], where this model is called “SPR3_IWM”.

MODEL > 10

If MODEL is chosen to be greater than 10, then 5 variables have to be defined as function IDs: STIFF, ALPHA1, RN, RS, and BETA. These functions incorporate the following input values: thicknesses of both weld partners (t1, t2) and maximum engineering yield stresses, also called necking points (sm1, sm2). For ALPHA1 = 100 such a function could look like,

```
*DEFINE_FUNCTION
       100
 func(t1,t2,sm1,sm2)=sm1/sm2
```

(This function is only a demonstration, it does not make any physical sense). For MODEL = 11 or 12, the master part is the first weld partner represented by t1 and sm1. For MODEL = 21 or 22, the thinner part is the first weld partner. Since material parameters have to be identified from both weld partners during initialization, this feature is only available for a subset of material models at the moment, namely no. 24, 120, 123, and 124.

*CONSTRAINED_JOINT_*OPTION*_{*OPTION*}_{*OPTION*}_{*OPTION*}

Available forms include (one is mandatory):

*CONSTRAINED_JOINT_SPHERICAL

*CONSTRAINED_JOINT_REVOLUTE

*CONSTRAINED_JOINT_CYLINDRICAL

*CONSTRAINED_JOINT_PLANAR

*CONSTRAINED_JOINT_UNIVERSAL

*CONSTRAINED_JOINT_TRANSLATIONAL

*CONSTRAINED_JOINT_LOCKING

*CONSTRAINED_JOINT_TRANSLATIONAL_MOTOR

*CONSTRAINED_JOINT_ROTATIONAL_MOTOR

*CONSTRAINED_JOINT_GEARS

*CONSTRAINED_JOINT_RACK_AND_PINION

*CONSTRAINED_JOINT_CONSTANT_VELOCITY

*CONSTRAINED_JOINT_PULLEY

*CONSTRAINED_JOINT_SCREW

If the force output data is to be transformed into a local coordinate use the option:

LOCAL

to define a joint ID and heading the following option is available:

ID

and to define failure for penalty-based joints (LMF = 0 in *CONTROL_RIGID) use:

FAILURE

The ordering of the bracketed options is arbitrary.

Purpose: Define a joint between two rigid bodies.

Card Format:

Card 1:	required for all joint types
Card 2:	required for joint types: MOTOR, GEARS, RACK_AND_PINION, PULLEY, and SCREW
Optional Card:	required only if LOCAL is specified in the keyword

In the first seven joint types above excepting the Universal joint, the nodal points within the nodal pairs (1, 2), (3, 4), and (5, 6) (see Figures 10-14 through 10-19) should coincide in the initial configuration, and the nodal pairs should be as far apart as possible to obtain the best behavior. For the Universal Joint the nodes within the nodal pair (3, 4) do not coincide, but the lines drawn between nodes (1, 3) and (2, 4) must be perpendicular.

For the Gear joint the nodes within the nodal pair (1, 2) must not coincide.

When the penalty method is used (see *CONTROL_RIGID), at each time step, the relative penalty stiffness is multiplied by a function dependent on the step size to give the maximum stiffness that will not destroy the stability of the solution. Instabilities can result in the explicit time integration scheme if the penalty stiffness is too large. If instabilities occur, the recommended way to eliminate these problems is to decrease the time step or reduce the scale factor on the penalties.

For cylindrical joints, by setting node 3 to zero, it is possible to use a cylindrical joint to join a node that is not on a rigid body (node 1) to a rigid body (nodes 2 and 4).

ID Card. Additional card for ID keyword option.

Optional	1	2	3	4	5	6	7	8
Variable	JID	HEADING						
Type	I	A70						

The heading is picked up by some of the peripheral LS-DYNA codes to aid in post-processing.

VARIABLE	DESCRIPTION
JID	Joint ID. This must be a unique number.
HEADING	Joint descriptor. It is suggested that unique descriptions be used.

Card 1	1	2	3	4	5	6	7	8
Variable	N1	N2	N3	N4	N5	N6	RPS	DAMP
Type	I	I	I	I	I	I	F	F
Default	0	0	0	0	0	0	1.0	1.0

VARIABLE	DESCRIPTION
N1	Node 1, in rigid body A. Define for all joint types.
N2	Node 2, in rigid body B. Define for all joint types.
N3	Node 3, in rigid body A. Define for all joint types except SPHERICAL.
N4	Node 4, in rigid body B. Define for all joint types except SPHERICAL.
N5	Node 5, in rigid body A. Define for joint types TRANSLATIONAL, LOCKING, ROTATIONAL_MOTOR, CONSTANT_VELOCITY, GEARS, RACK_AND_PINION, PULLEY, and SCREW
N6	Node 6, in rigid body B. Define for joint types TRANSLATIONAL, LOCKING, ROTATIONAL_MOTOR, CONSTANT_VELOCITY, GEARS, RACK_AND_PINION, PULLEY, and SCREW
RPS	Relative penalty stiffness (default = 1.0): GT.0.0: constant value, LT.0.0: time dependent value given by load curve ID = -RPS (only for SPHERICAL, REVOLUTE, and CYLINDRICAL).
DAMP	Damping scale factor on default damping value. (Revolute and Spherical Joints): EQ.0.0: default is set to 1.0, GT.0.0.AND.LE.0.01: no damping is used.

Rotational Properties Card. Additional card for joint types MOTOR, GEARS, RACK_AND_PINION, PULLEY, and SCREW.

Card 2	1	2	3	4	5	6	7	8
Variable	PARM	LCID	TYPE	R1	H_ANGLE			
Type	F	I	I	F	F			
Default	none				0.0			

VARIABLE	DESCRIPTION

PARM

Parameter, which a function of joint type:

Gears:	define R_2/R_1
Rack and Pinion:	define h
Pulley:	define R_2/R_1
Screw:	define $\dot{x}/\omega$
Motors:	leave blank

LCID

Define load curve ID for MOTOR joints.

TYPE

Define integer flag for MOTOR joints as follows:

EQ.0: translational/rotational velocity

EQ.1: translational/rotational acceleration

EQ.2: translational/rotational displacement

R1

Radius, R_1, for the gear and pulley joint type. If left undefined, nodal points 5 and 6 are assumed to be on the outer radius. The value of R1 and R2 affect the reaction forces written to output. The forces are calculated from the moments by dividing them by the radii.

H_ANGLE

Helix angle in degrees. This is only necessary for the gear joint if the gears do not mesh tangentially, e.g., worm gears, see remarks below for a definition.

Local Card. Additional card required for LOCAL keyword option.

Card 3	1	2	3	4	5	6	7	8
Variable	RAID	LST						
Type	I	I						
Default	0	0						

VARIABLE **DESCRIPTION**

RAID Rigid body or accelerometer ID. The force resultants are output in the local system of the rigid body or accelerometer.

LST Flag for local system type:

EQ.0: rigid body

EQ.1: accelerometer

Failure Card 1. Additional card for FAILURE keyword option.

Card 4	1	2	3	4	5	6	7	8
Variable	CID	TFAIL	COUPL					
Type	I	F	F					
Default	0	0	0.					

Failure Card 2. Additional card for FAILURE keyword option.

Card 5	1	2	3	4	5	6	7	8
Variable	NXX	NYY	NZZ	MXX	MYY	MZZ		
Type	F	F	F	F	F	F		
Default	0	0	0	0	0	0		

VARIABLE **DESCRIPTION**

CID

Coordinate ID for resultants in the failure criteria. If zero, the global coordinate system is used.

TFAIL

Time for joint failure. If zero, joint never fails.

COUPL

Coupling between the force and moment failure criteria. If COUPL is less than or equal to zero, the failure criteria is identical to the spotwelds. When COUPL is greater than zero, the force and moment results are considered independently. See the remark below.

NXX

Axial force resultant N_{xx_F}at failure. If zero, failure due to this component is not considered.

NYY

Force resultant N_{yy_F} at failure. If zero, failure due to this component is not considered.

NZZ

Force resultant N_{zz_F} at failure. If zero, failure due to this component is not considered.

MXX

Torsional moment resultant M_{zz_F} at failure. If zero, failure due to this component is not considered.

MYY

Moment resultant M_{yy_F} at failure. If zero, failure due to this component is not considered.

MZZ

Moment resultant M_{zz_F} at failure. If zero, failure due to this component is not considered.

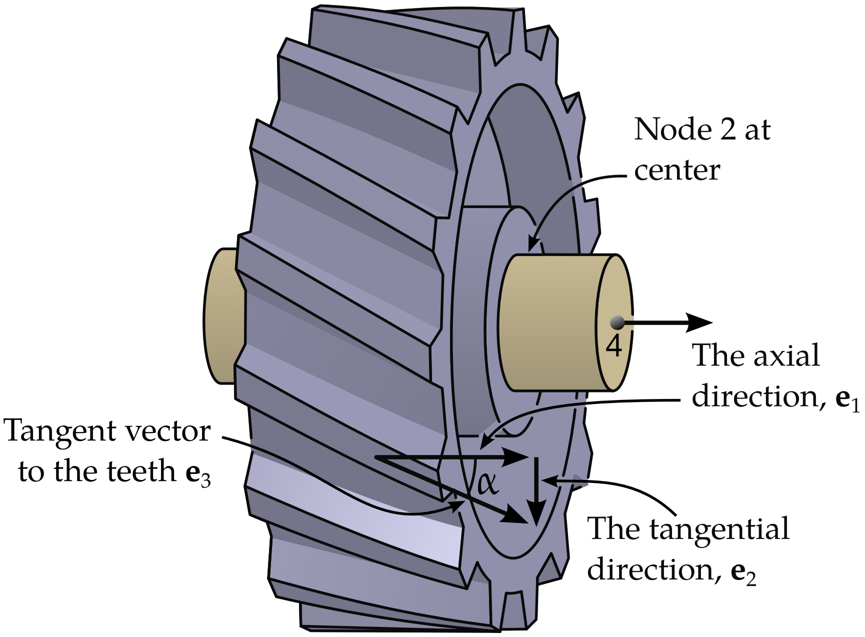

Node 2 at center
4
The axial direction, $\mathbf{e}_1$
Tangent vector to the teeth $\mathbf{e}_3$
α
The tangential direction, $\mathbf{e}_2$

Figure 10-13. Helix angle α definition, gear #2 viewed from the extension of node n_2 to node n_6.

Remarks:

The moments for the revolute, cylindrical, planar, translational, and locking joints are calculated at the midpoint of nodes N1 and N3. The moments for the spherical, universal, constant velocity, gear, pulley, and rack and pinion joints are calculated at node N1. When COUPL is less than or equal to zero, the failure criteria is

$$\left(\frac{N_{xx}}{N_{xx_F}}\right)^2+\left(\frac{N_{yy}}{N_{yy_F}}\right)^2+\left(\frac{N_{zz}}{N_{zz_F}}\right)^2+\left(\frac{M_{xx}}{M_{xx_F}}\right)^2+\left(\frac{M_{yy}}{M_{yy_F}}\right)^2+\left(\frac{M_{zz}}{M_{zz_F}}\right)^2-1=0.$$

Otherwise, it consists of both

$$\left(\frac{N_{xx}}{N_{xx_F}}\right)^2+\left(\frac{N_{yy}}{N_{yy_F}}\right)^2+\left(\frac{N_{zz}}{N_{zz_F}}\right)^2-1=0,$$

and

$$\left(\frac{M_{xx}}{M_{xx_F}}\right)^2+\left(\frac{M_{yy}}{M_{yy_F}}\right)^2+\left(\frac{M_{zz}}{M_{zz_F}}\right)^2-1=0.$$

For a gear joint, the relative direction and magnitude of rotation between the two gears is determined by the *helix angle*. Let $\mathbf{e}_1$ be the unit normal directed from node 2 to 4, which corresponds to the second gear's rotation axis. See Figure 10-23. Let $\mathbf{e}_2$ be defined as the positively oriented tangent vector to motion of the teeth when spun about the $\mathbf{e}_1$ axis (the gear's axis). See Figure 10-13. The helix angle α characterizes the deviation of the teeth axis from the gear axis. In particular, α is defined as the angle between the direction of teeth, called $\mathbf{e}_3$, and the axis of the gear $\mathbf{e}_1$,

$$\mathbf{e}_3 = \cos\alpha\mathbf{e}_1 + \sin\alpha\mathbf{e}_2.$$

The gears are assumed to be setup so that the teeth initially fit having matching $\mathbf{e}_3$ directions. A nonzero helix angle is typically used to model worm gears.

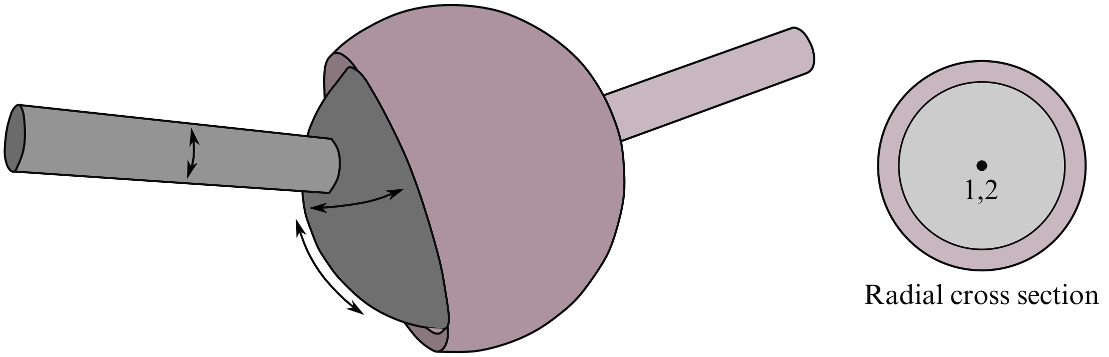


Figure 10-14. *Spherical joint*. The relative motion of the rigid bodies is constrained so that nodes which are initially coincident remain coincident. In the above figure the socket's node is not interior to the socket—LS-DYNA does not require that a rigid body's nodes be interior to the body.

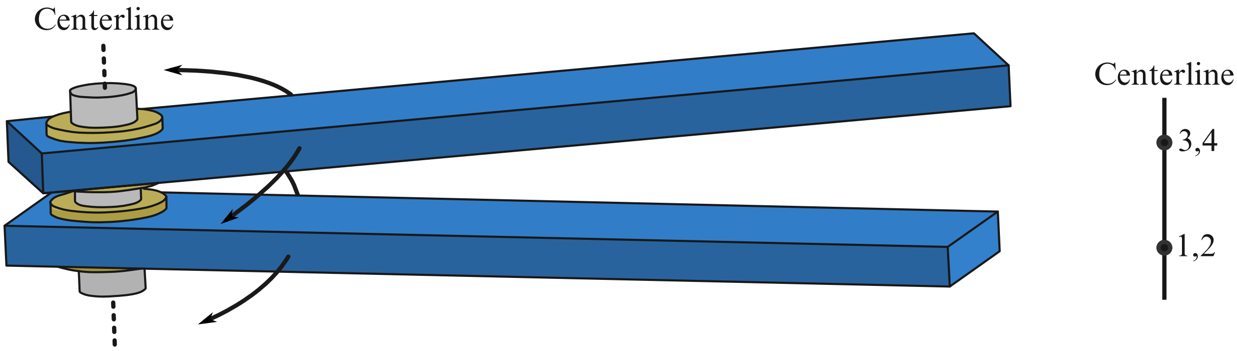


Figure 10-15. *Revolute Joint*. Nodes 1 and 2 are coincident; nodes 3 and 4 are coincident. Nodes 1 and 3 belong to rigid body A; nodes 2 and 4 belong to rigid body B. The relative motion of the two rigid bodies is restricted to rotations about the axis formed by the two pairs of coincident nodes. This axis is labeled the "centerline".

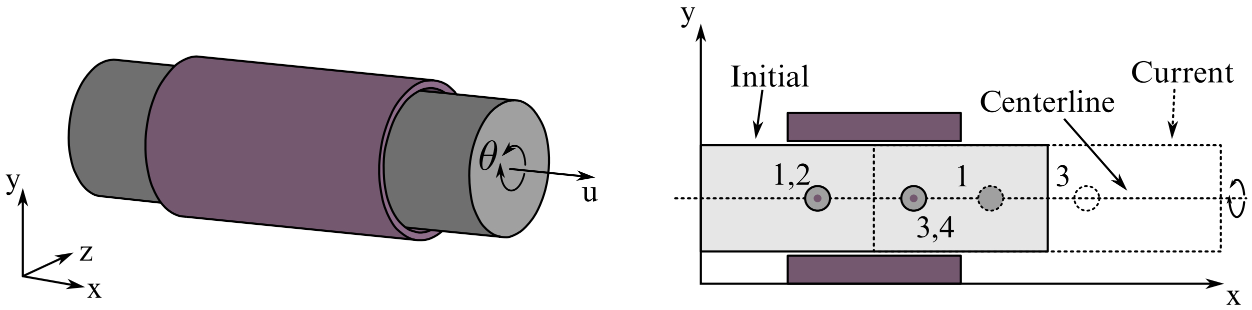


Figure 10-16. *Cylindrical Joint*. This joint is derived from the rotational joint by relaxing the constraints along the centerline. This joint admits relative rotation and translation along the centerline.

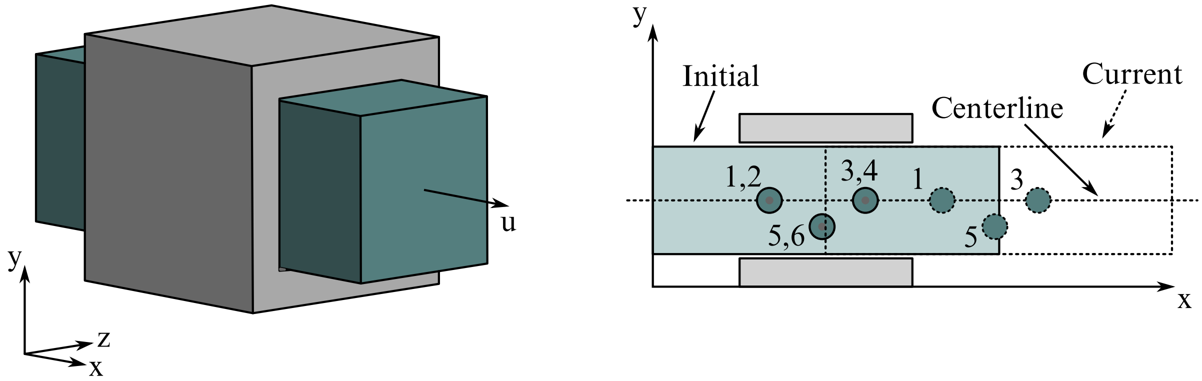


Figure 10-17. *Translational joint*. This is a cylindrical joint with a third pair of off-centerline nodes which restrict rotation. Aside from translation along the centerline, the two rigid bodies are stuck together.

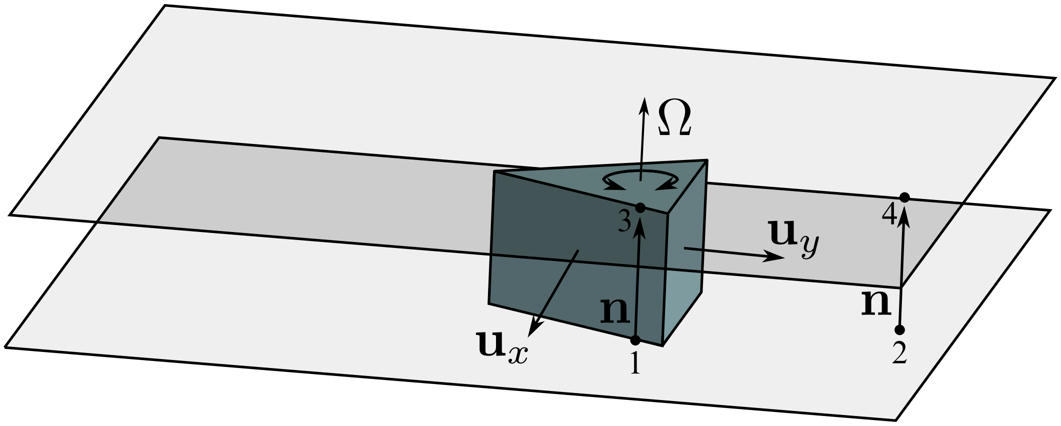


Figure 10-18. *Planar joint*. This joint is derived from the rotational joint by relaxing the constraints normal to the centerline. Relative displacements along the direction of the centerline are excluded.

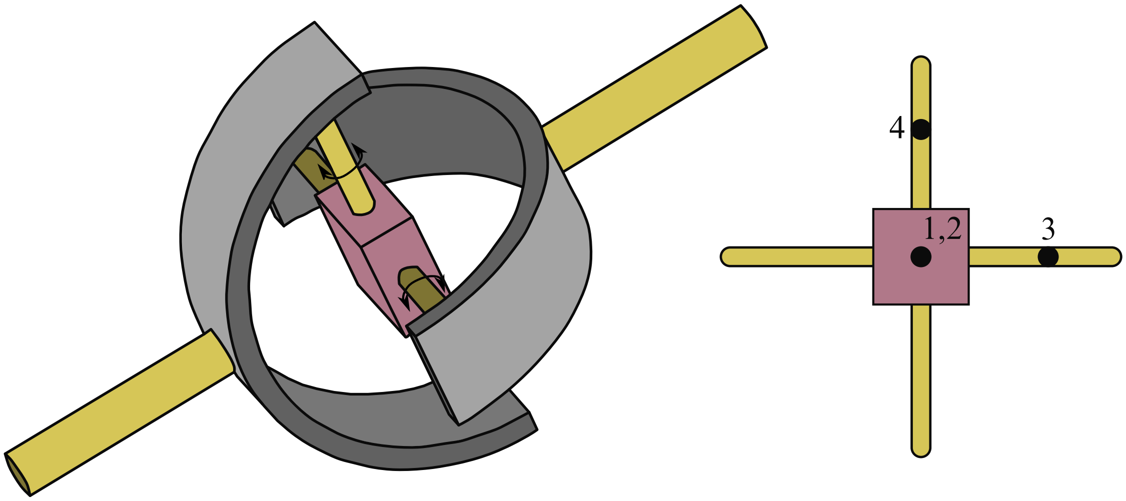


Figure 10-19. *Universal Joint*. Nodes 1 and 2 are initially coincident. The segments formed by nodal pairs (1, 3) and (2, 4) must be orthogonal; they serve as axes about which the two bodies may undergo relative rotation. The universal joint excludes all other relative motion and the axes remain orthogonal at all time.

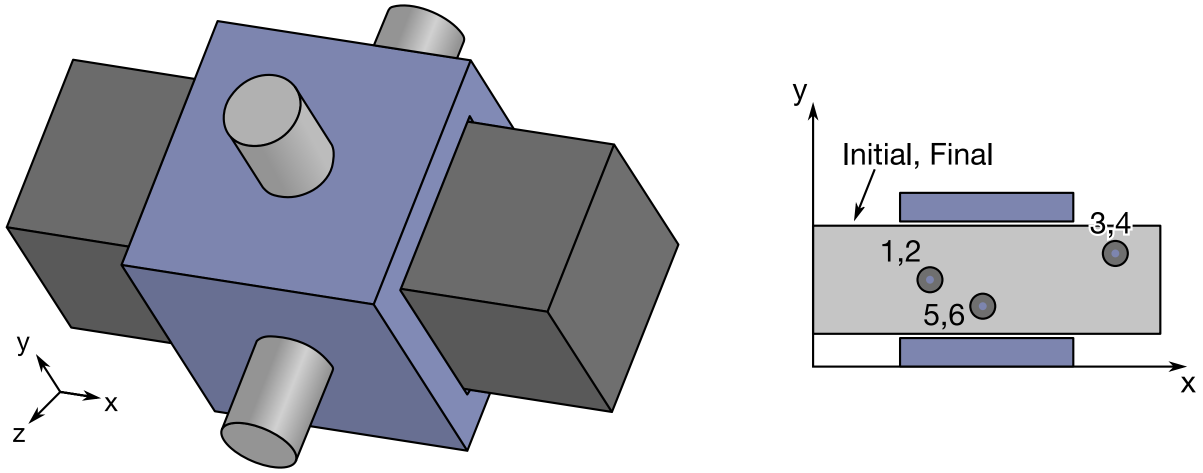


Figure 10-20. *Locking Joint*. A locking joint couples two rigid bodies in all six degrees-of-freedom. The forces and moments required to form this coupling are written to the jntforc file (*DATABASE_JNTFORC). As stated in the Remarks, forces and moments in jntforc are calculated halfway between N1 and N3. Nodal pairs (1, 2), (3, 4) and (5, 6) must be coincident. The three spatial points corresponding to three nodal pairs must be neither collocated nor collinear.

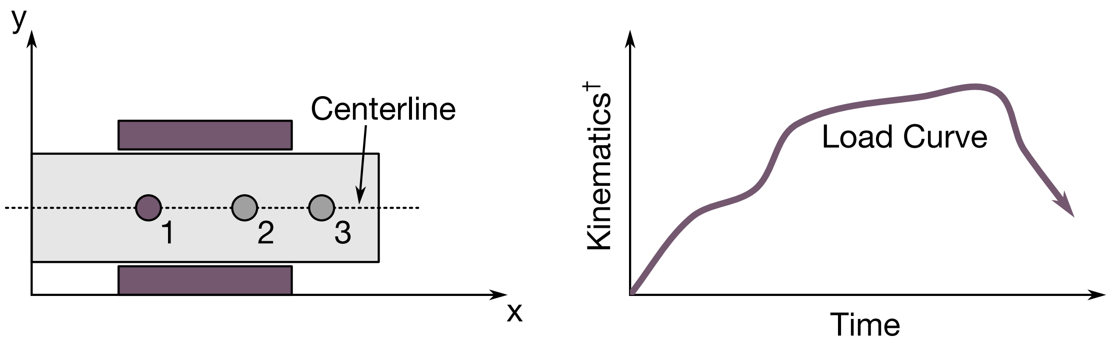


Figure 10-21. Translational motor joint. This joint is usually used in combination with a translational or a cylindrical joint. Node 1 and node 2 belong to the first rigid body and the second rigid body, resp. Furthermore, nodes 1 and 2 must be coincident. Node 3 may belong to either rigid body. The vector from node 2 to node 3 is the direction of relative motion. Node 4 is not used and can be left blank. The value of the load curve may specify any of several kinematic measures; see TYPE.

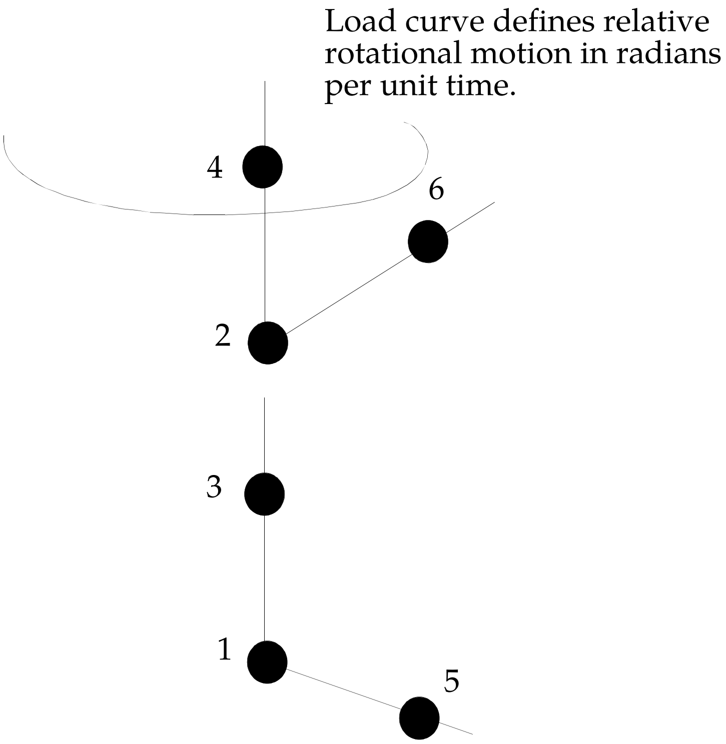

Load curve defines relative rotational motion in radians per unit time.
4
6
2
3
1
5

Figure 10-22. Rotational motor joint. This joint can be used in combination with other joints such as the revolute or cylindrical joints.

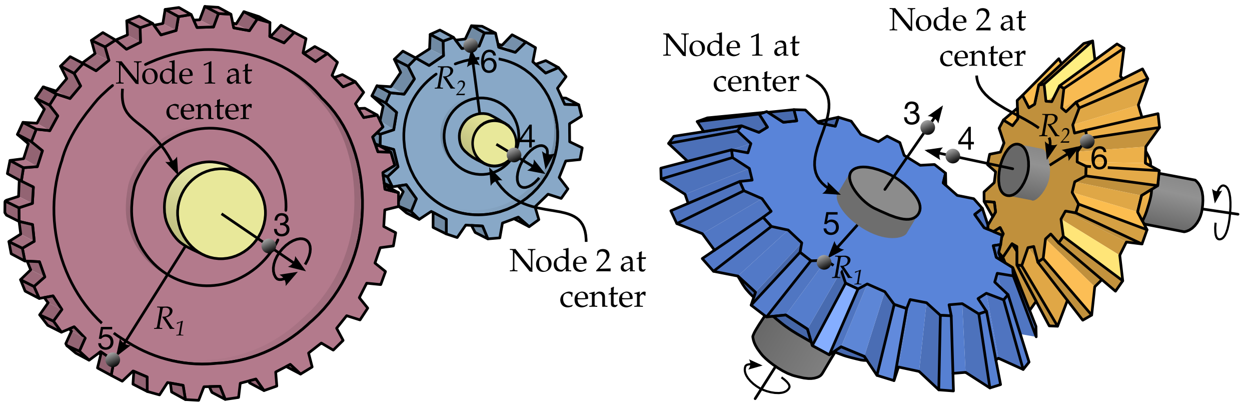


Figure 10-23. Gear joints. Nodal pairs (1, 3) and (2, 4) define axes that are orthogonal to the gears. Nodal pairs (1, 5) and (2, 6) define vectors in the plane of the gears. The ratio R_2/R_1 is specified but need not necessarily correspond to the geometry, if for instance the gear consists of spiral grooves. Note that the gear joint in itself does not maintain the contact point but this requires additional treatment, such as accompanying it with other joints.

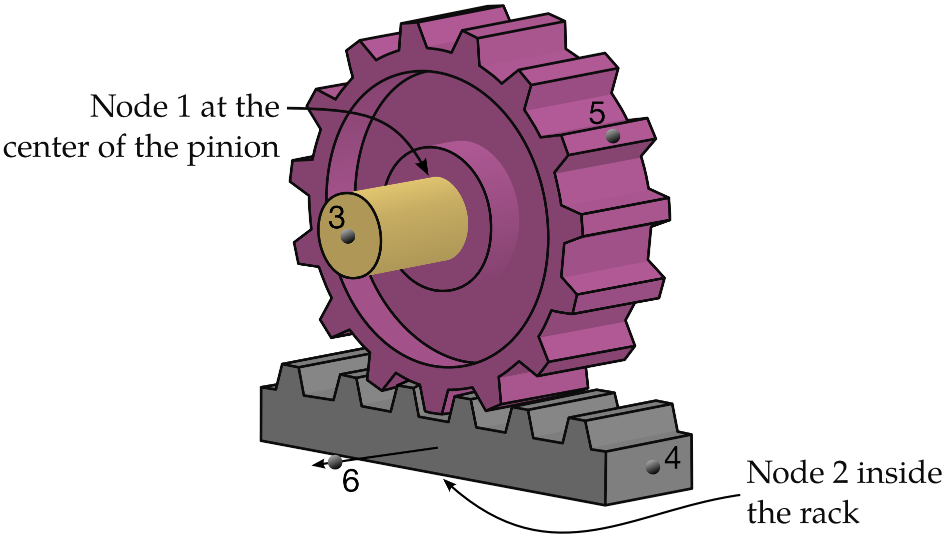


Figure 10-24. Rack and pinion joint. Nodal pair (1, 3) defines the axis of rotation of the first body (the pinion). Nodal pair (1, 5) is a vector in the plane of the pinion and is orthogonal to nodal pair (1, 3). Nodal pair (2, 4) defines the direction of travel for the second body (the rack). Nodal pair (2, 6) is parallel to the axis of the pinion and is thus parallel to nodal pair (1, 3). The value h is specified. The velocity of the rack is $h\omega_{\text{pinion}}$.

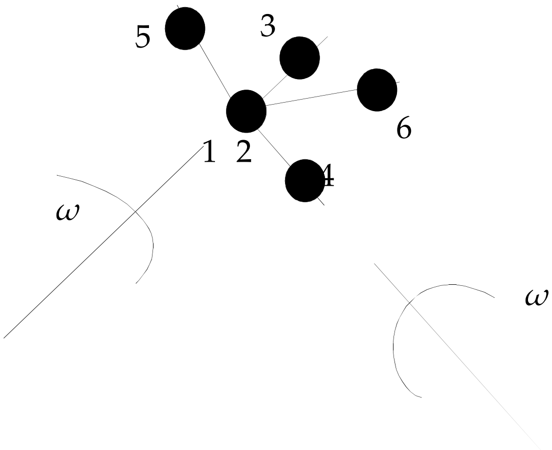


Figure 10-25. Constant velocity joint. Nodal pairs (1, 3) and (2, 4) define an axes for the constant angular velocity, and nodal pairs (1, 5) are orthogonal vectors. Here nodal points 1 and 2 must be coincident.

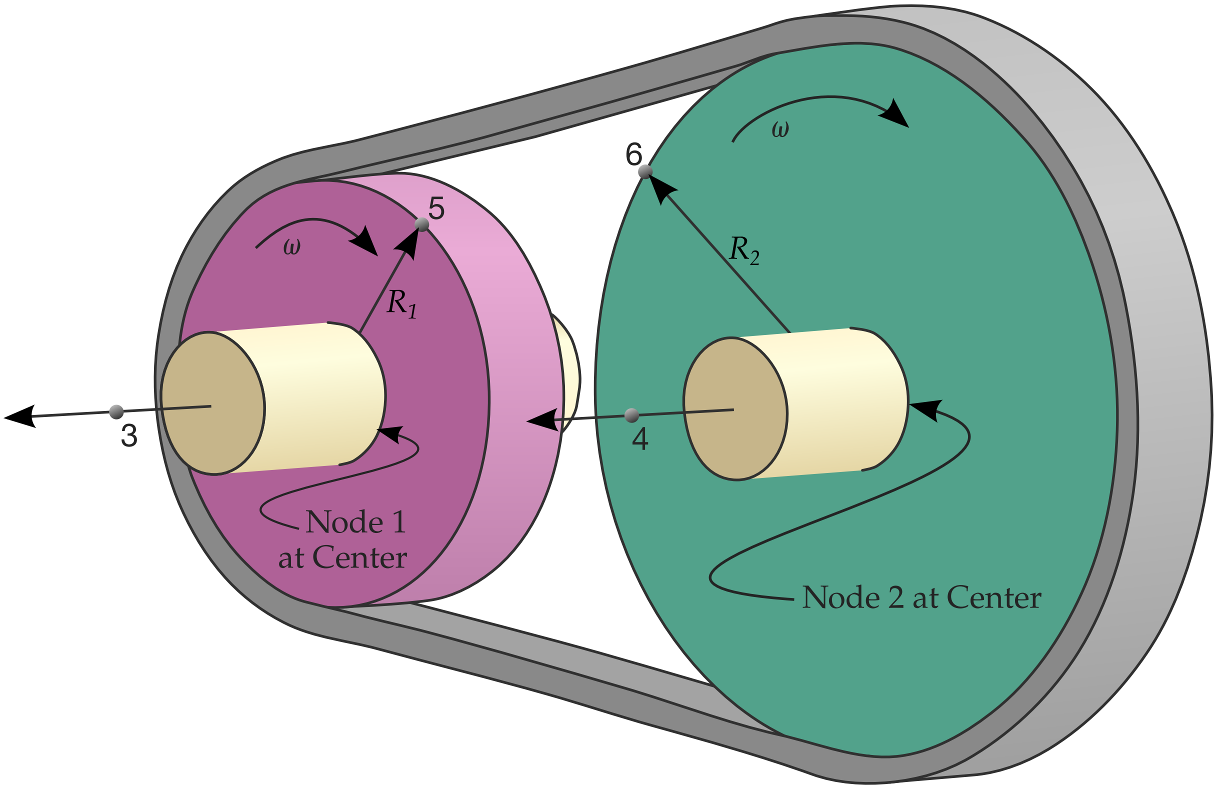


Figure 10-26. Pulley joint. Nodal pairs (1, 3) and (2, 4) define axes that are orthogonal to the pulleys. Nodal pairs (1, 5) and (2, 6) define vectors in the plane of the pulleys. The ratio R_2/R_1 is specified.

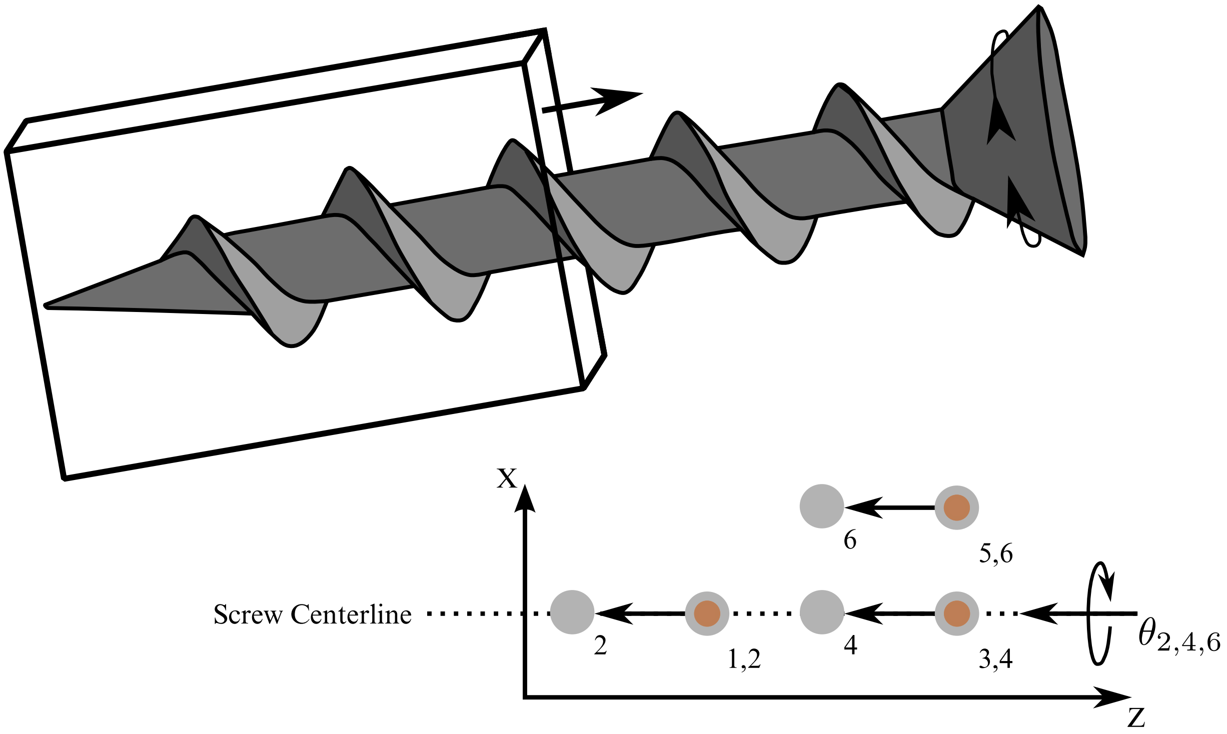


Figure 10-27. Screw joint. The second body translates in response to the spin of the first body. Nodal pairs (1, 3) and (2, 4) lie along the same axis and nodal pairs (1, 5) and (2, 6) are orthogonal vectors. The helix ratio, $\dot{x}/_{\omega}$, is specified.

Example 1:

```
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$  *CONSTRAINED_JOINT_PLANAR
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Define a planar joint between two rigid bodies.
$    - Nodes 91 and 94 are on rigid body 1.
$    - Nodes 21 and 150 are on rigid body 2.
$    - Nodes 91 and 21 must be coincident.
$        * These nodes define the origin of the joint plane.
$    - Nodes 94 and 150 must be coincident.
$        * To accomplish this, massless node 150 is artificially created at
$           the same coordinates as node 94 and then added to rigid body 2.
$        * These nodes define the normal of the joint plane (e.g., the
$           vector from node 91 to 94 defines the planes' normal).
$
*CONSTRAINED_JOINT_PLANAR
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$       n1        n2        n3        n4        n5        n6       rps
        91        21        94       150                     0.000E+00
$
$
*NODE
$    nid               x               y               z      tc      rc
     150            0.00            3.00            0.00       0       0
$
*CONSTRAINED_EXTRA_NODES_SET
$      pid      nsid
         2         6
$*SET_NODE_LIST
$      sid
         6
$     nid1
       150
$
$$$  request output for joint force data
$
*DATABASE_JNTFORC
$  dt/cycl      lcdt
    0.0001
$
```

Example 2:

```
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$  *CONSTRAINED_JOINT_REVOLUTE
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Create a revolute joint between two rigid bodies.  The rigid bodies must
$  share a common edge to define the joint along.  This edge, however, must
$  not have the nodes merged together.  Rigid bodies A and B will rotate
$  relative to each other along the axis defined by the common edge.
$
$  Nodes 1 and 2 are on rigid body A and coincide with nodes 9 and 10
$  on rigid body B, respectively.  (This defines the axis of rotation.)
$
```

```
$  The relative penalty stiffness on the revolute joint is to be 1.0,
$
*CONSTRAINED_JOINT_REVOLUTE
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$       n1        n2        n3        n4        n5        n6       rps      damp
         1         9         2        10                             1.0
$
$  Note:  A joint stiffness is not mandatory for this joint to work.
$         However, to see how a joint stiffness can be defined for this
$         particular joint, see the corresponding example listed in:
$            *CONSTRAINED_JOINT_STIFFNESS_GENERALIZED
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
```

CONSTRAINED_JOINT_COOR_*OPTION_{OPTION}_{OPTION}_{OPTION}

Available forms include (one is mandatory):

*CONSTRAINED_JOINT_COOR_SPHERICAL

*CONSTRAINED_JOINT_COOR_REVOLUTE

*CONSTRAINED_JOINT_COOR_CYLINDRICAL

*CONSTRAINED_JOINT_COOR_PLANAR

*CONSTRAINED_JOINT_COOR_UNIVERSAL

*CONSTRAINED_JOINT_COOR_TRANSLATIONAL

*CONSTRAINED_JOINT_COOR_LOCKING

*CONSTRAINED_JOINT_COOR_TRANSLATIONAL_MOTOR

*CONSTRAINED_JOINT_COOR_ROTATIONAL_MOTOR

*CONSTRAINED_JOINT_COOR_GEARS

*CONSTRAINED_JOINT_COOR_RACK_AND_PINION

*CONSTRAINED_JOINT_COOR_CONSTANT_VELOCITY

*CONSTRAINED_JOINT_COOR_PULLEY

*CONSTRAINED_JOINT_COOR_SCREW

If the force output data is to be transformed into a local coordinate use the option:

LOCAL

to define a joint ID and heading the following option is available:

ID

and to define failure for penalty-based joints (LMF = 0 in *CONTROL_RIGID) use:

FAILURE

The ordering of the bracketed options is arbitrary.

Purpose: Define a joint between two rigid bodies, see Figure. The connection coordinates are given instead of the nodal point IDs required in the previous section, *CONSTRAINED_JOINT_{*OPTION*}. Nodes are automatically generated for each coordinate and are constrained to the rigid body. Where coincident nodes are expected in the initial configuration, only one connection coordinate is needed since the connection coordinate for the second node, if given, is ignored. The created nodal ID's are chosen to exceed the maximum user ID. The coordinates of the joint nodes are specified on Cards 2 - 7. The input which follows Card 7 is identical to that in the previous section.

Card Format:

Cards 1 - 7:	required for all joint types
Card 8:	required for joint types: MOTOR, GEARS, RACK_AND_PINION, PULLEY, and SCREW
Optional Card:	required when LOCAL is specified in the keyword

In the first seven joint types above excepting the Universal joint, the coordinate points within the nodal pairs (1, 2), (3, 4), and (5, 6) (see Figures 10-14 through 10-20) should coincide in the initial configuration, and the nodal pairs should be as far apart as possible to obtain the best behavior. For the Universal Joint the nodes within the coordinate pair (3, 4) do not coincide, but the lines drawn between nodes (1, 3) and (2, 4) must be perpendicular.

For the Gear joint the nodes within the coordinate pair (1, 2) must not coincide.

When the penalty method is used (see *CONTROL_RIGID), at each time step, the relative penalty stiffness is multiplied by a function dependent on the step size to give the maximum stiffness that will not destroy the stability of the solution. LS-DYNA's explicit time integrator can become unstable when the penalty stiffness is too large. If instabilities occur, the recommended way to eliminate these problems is to decrease the time step or reduce the scale factor on the penalties.

For cylindrical joints, by setting node 3 to zero, it is possible to use a cylindrical joint to join a node that is not on a rigid body (node 1) to a rigid body (nodes 2 and 4).

ID Card. Additional card for the ID keyword option.

Optional	1	2	3	4	5	6	7	8
Variable	JID	HEADING						
Type	I	A70						

The heading is picked up by some of the peripheral LS-DYNA codes to aid in post-processing.

VARIABLE	DESCRIPTION
JID	Joint ID. This must be a unique number.
HEADING	Joint descriptor. It is suggested that unique descriptions be used.

Card 1	1	2	3	4	5	6	7	8
Variable	RBID_A	RBID_B	RPS	DAMP	TMASS	RMASS		
Type	I	I	F	F	F	F		

Card 2	1	2	3	4	5	6	7	8
Variable	X1	Y1	Z1					
Type	F	F	F					

Card 3	1	2	3	4	5	6	7	8
Variable	X2	Y2	Z2					
Type	F	F	F					

Card 4	1	2	3	4	5	6	7	8
Variable	X3	Y3	Z3					
Type	F	F	F					

Card 5	1	2	3	4	5	6	7	8
Variable	X4	Y4	Z4					
Type	F	F	F					

Card 6	1	2	3	4	5	6	7	8
Variable	X5	Y5	Z5					
Type	F	F	F					

Card 7	1	2	3	4	5	6	7	8
Variable	X6	Y6	Z6					
Type	F	F	F					

VARIABLE **DESCRIPTION**

RBID_A Part ID of rigid body A.

RBID_B Part ID of rigid body B.

RPS Relative penalty stiffness (default = 1.0).

DAMP Damping scale factor on default damping value. (Revolute and Spherical Joints):

EQ.0.0: default is set to 1.0,

GT.0.0 and LE.0.01: no damping is used.

TMASS Lumped translational mass. The mass is equally split between the first points defined for rigid bodies A and B.

RMASS Lumped rotational inertia. The inertia is equally split between the first points defined for rigid bodies A and B.

X1, Y1, Z1 Coordinate of point 1, in rigid body A. Define for all joint types.

VARIABLE	DESCRIPTION
X2, Y2, Z2	Coordinate of point 2, in rigid body B. If points 1 and 2 are coincident in the specified joint type, the coordinate for point 1 is used.
X3, Y3, Z31	Coordinate of point 3, in rigid body A. Define for all joint types.
X4, Y4, Z4	Coordinate of point 4, in rigid body B. If points 3 and 4 are coincident in the specified joint type, the coordinate for point 3 is used.
X5, Y5, Z5	Coordinate of point 5, in rigid body A. Define for all joint types.
X6, Y6, Z6	Coordinate of point 6, in rigid body B. If points 5 and 6 are coincident in the specified joint type, the coordinate for point 5 is used.

Rotational Properties Card. Additional card for joint types MOTOR, GEARS, RACK_AND_PINION, PULLEY, and SCREW.

Card 8	1	2	3	4	5	6	7	8
Variable	PARM	LCID	TYPE	R1				
Type	F	I	I	F				
Default	none							

VARIABLE	DESCRIPTION
PARM	Parameter, which a function of joint type: Gears: define R_2/R_1 Rack and Pinion: define h Pulley: define R_2/R_1 Screw: define $\dot{x}/\omega$ Motors: leave blank
LCID	Define load curve ID for MOTOR joints.

VARIABLE	DESCRIPTION
TYPE	Define integer flag for MOTOR joints as follows: EQ.0: translational/rotational velocity EQ.1: translational/rotational acceleration EQ.2: translational/rotational displacement
R1	Radius, R_1, for the gear and pulley joint type. If left undefined, nodal points 5 and 6 are assumed to be on the outer radius. R1 is the moment arm that goes into calculating the joint reaction forces. The ratio R2/R1 gives the transmitted moments, but not the forces. The force is moment divided by distance R1.

Local Card. Additional card for LOCAL keyword option.

Card 9	1	2	3	4	5	6	7	8
Variable	RAID	LST						
Type	I	I						
Default	0	0						

VARIABLE	DESCRIPTION
RAID	Rigid body or accelerometer ID. The force resultants are output in the local system of the rigid body or accelerometer.
LST	Flag for local system type: EQ.0: rigid body EQ.1: accelerometer

Failure Card 1. Additional card for the FAILURE keyword option.

Card 10	1	2	3	4	5	6	7	8
Variable	CID	TFAIL	COUPL					
Type	I	F	F					
Default	0	0	0.					

Failure Card 2. Additional card for the FAILURE keyword option.

Card 11	1	2	3	4	5	6	7	8
Variable	NXX	NYY	NZZ	MXX	MYY	MZZ		
Type	F	F	F	F	F	F		
Default	0	0	0	0	0	0		

VARIABLE	DESCRIPTION
CID	Coordinate ID for resultants in the failure criteria. If zero, the global coordinate system is used.
TFAIL	Time for joint failure. If zero, joint never fails.
COUPL	Coupling between the force and moment failure criteria. If COUPL is less than or equal to zero, the failure criteria is identical to the spotwelds. When COUPL is greater than zero, the force and moment results are considered independently. See the remarks in *CONSTRAINED_JOINT_{*OPTION*}.
NXX	Axial force resultant N_{xx_F}at failure. If zero, failure due to this component is not considered.
NYY	Force resultant N_{YY_F} at failure. If zero, failure due to this component is not considered.
NZZ	Force resultant N_{zz_F} at failure. If zero, failure due to this component is not considered.

VARIABLE	DESCRIPTION
MXX	Torsional moment resultant M_{XX_F} at failure. If zero, failure due to this component is not considered.
MYY	Moment resultant M_{YY_F} at failure. If zero, failure due to this component is not considered.
MZZ	Moment resultant M_{ZZ_F} at failure. If zero, failure due to this component is not considered.

*CONSTRAINED_JOINT_STIFFNESS_*OPTION*_*{OPTION}*

Available options include:

FLEXION-TORSION

GENERALIZED

TRANSLATIONAL

If desired a description of the joint stiffness can be provided with the option:

TITLE

which is written into the d3hsp and jntforc files.

Purpose: Define optional rotational and translational joint stiffness for joints defined by *CONSTRAINED_JOINT_*OPTION.* These definitions apply to all joints even though degrees of freedom that are considered in the joint stiffness capability may be constrained out in some joint types. The energy that is dissipated with the joint stiffness option is written for each joint in joint force file with the default name, jntforc. In the global energy balance this energy is included with the energy of the discrete elements, i.e., the springs and dampers.

Card Format:

The optional TITLE card and card 1 are common to all joint stiffness types.
Cards 2 to 4 are unique for each stiffness type.

Title Card. Additional card for the TITLE keyword option.

Optional	1	2	3	4	5	6	7	8
Variable	TITLE							
Type	A80							

Card 1	1	2	3	4	5	6	7	8
Variable	JSID	PIDA	PIDB	CIDA	CIDB	JID		
Type	I	I	I	I	I	I		
Default	none	none	none	none	CIDA	none		

VARIABLE	DESCRIPTION
TITLE	Description of joint stiffness for output files jntforc and d3hsp.
JSID	Joint stiffness ID
PIDA	Part ID for rigid body A, see *PART.
PIDB	Part ID for rigid body B, see *PART.
CIDA	Coordinate ID for rigid body A, see *DEFINE_COORDINATE_-OPTION. For the translational stiffness the local coordinate system must be defined by nodal points, *DEFINE_COORDINATE_NODES, since the first nodal point in each coordinate system is used to track the motion.
CIDB	Coordinate ID for rigid body B. If zero, the coordinate ID for rigid body A is used, see *DEFINE_COORDINATE_OPTION. For the translational stiffness the local coordinate system must be defined by nodal points, *DEFINE_COORDINATE_NODES, since the first nodal point in each coordinate system is used to track the motion.
JID	Joint ID for the joint reaction forces. If zero, tables can't be used in place of load curves for defining the frictional moments.

Card 2 for FLEXION-TORSION option.

Card 2	1	2	3	4	5	6	7	8
Variable	LCIDAL	LCIDG	LCIDBT	DLCIDAL	DLCIDG	DLCIDBT		
Type	I	I	I	I	I	I		
Default	none	1.0	none	none	1.0	none		

Card 3 for FLEXION-TORSION option.

Card 3	1	2	3	4	5	6	7	8
Variable	ESAL	FMAL	ESBT	FMBT				
Type	F	F	F	F				
Default	0.0	0.0	0.0	0.0				

Card 4 for FLEXION-TORSION option.

Card 4	1	2	3	4	5	6	7	8
Variable	SAAL	NSABT	PSABT					
Type	F	F	F					
Default	not used	not used	not used					

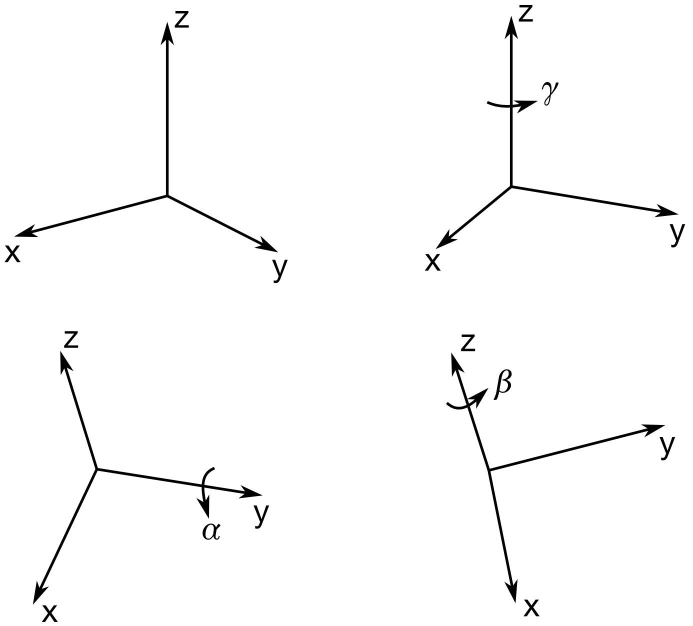


Figure 10-28. The angles α, β, γ align rigid body one with rigid body two for the FLEXION-TORSION option.

VARIABLE	DESCRIPTION
LCIDAL	Load curve ID for α-moment versus rotation in radians. See Figure 10-28 where it should be noted that $0 \leq \alpha \leq \pi$. If zero, the applied moment is set to zero. See *DEFINE_CURVE.
LCIDG	Load curve ID for γ versus a scale factor which scales the bending moment due to the α rotation. This load curve should be defined in the interval $-\pi \leq \gamma \leq \pi$. If zero the scale factor defaults to 1.0. See *DEFINE_CURVE.
LCIDBT	Load curve ID for β-torsion moment versus twist in radians. If zero the applied twist is set to zero. See *DEFINE_CURVE.
DLCIDAL	Load curve ID for α-damping moment versus rate of rotation in radians per unit time. If zero, damping is not considered. See *DEFINE_CURVE.
DLCIDG	Load curve ID for γ-damping scale factor versus rate of rotation in radians per unit time. This scale factor scales the α-damping moment. If zero, the scale factor defaults to one. See *DEFINE_CURVE.
DLCIDBT	Load curve ID for β-damping torque versus rate of twist. If zero damping is not considered. See *DEFINE_CURVE.

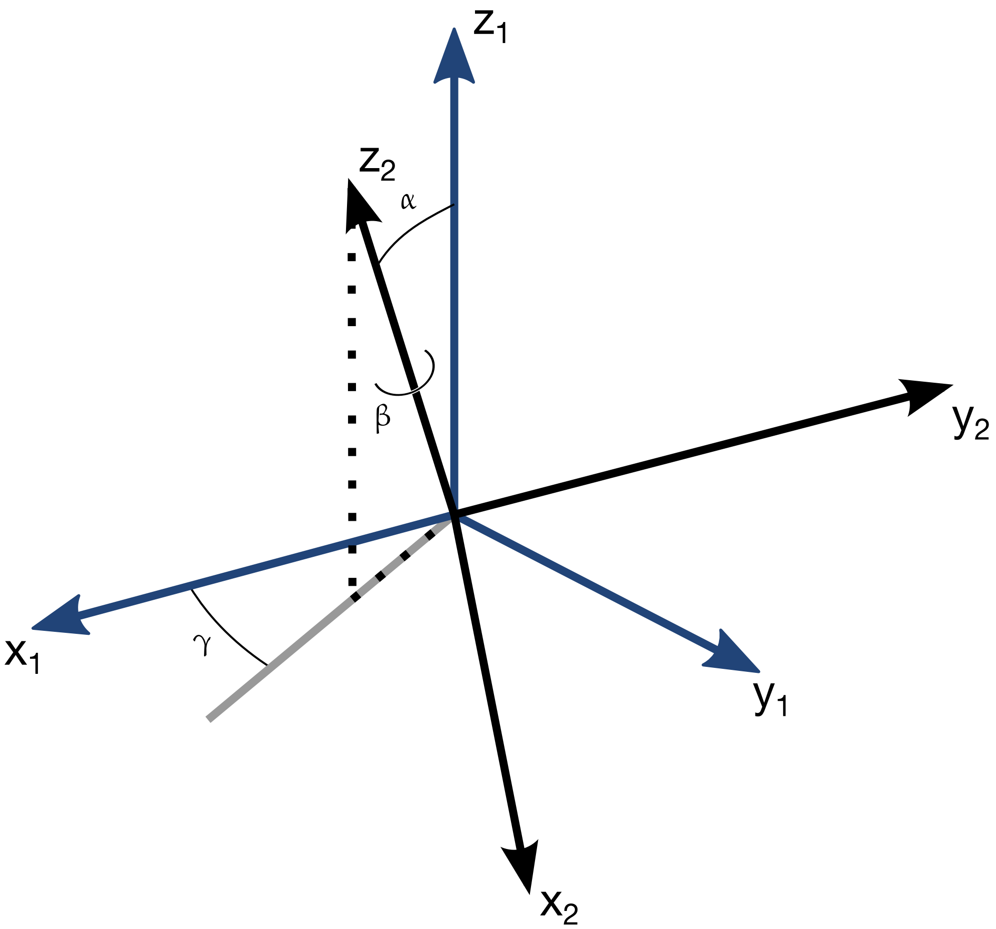


Figure 10-29. Flexion-torsion joint angles. If the initial positions of the local coordinate axes of the two rigid bodies connected by the joint do not coincide, the angles, α and γ, are initialized and torques will develop instantaneously based on the defined load curves. The angle β is also initialized but no torque will develop about the local axis on which β is measured. Rather, β will be measured relative to the computed offset.

VARIABLE **DESCRIPTION**

ESAL Elastic stiffness per unit radian for friction and stop angles for α rotation. If zero, friction and stop angles are inactive for α rotation. See Figure 10-31.

FMAL Frictional moment limiting value for α rotation. If zero, friction is inactive for α rotation. This option may also be thought of as an elastic-plastic spring. If a negative value is input then the absolute value is taken as the load curve or table ID defining the yield moment versus α rotation. A table permits the moment to also be a function of the joint reaction force and requires the specification of JID on Card 1. See Figure 10-31.

VARIABLE	DESCRIPTION
ESBT	Elastic stiffness per unit radian for friction and stop angles for β twist. If zero, friction and stop angles are inactive for β twist.
FMBT	Frictional moment limiting value for β twist. If zero, friction is inactive for β twist. This option may also be thought of as an elastic-plastic spring. If a negative value is input then the absolute value is taken as the load curve or table ID defining the yield moment versus β rotation. A table permits the moment to also be a function of the joint reaction force and requires the specification of JID on Card 1.
SAAL	Stop angle in degrees for α rotation where $0 \leq \alpha \leq \pi$. Ignored if zero. See Figure 10-31.
NSABT	Stop angle in degrees for negative β rotation. Ignored if zero.
PSABT	Stop angle in degrees for positive β rotation. Ignored if zero.

Remarks:

This option simulates a flexion-torsion behavior of a joint in a slightly different fashion than with the generalized joint option.

After the stop angles are reached the torques increase linearly to resist further angular motion using the stiffness values on Card 3. If the stiffness value is too low or zero, the stop will be violated.

The moment resultants generated from the moment versus rotation curve, damping moment versus rate-of-rotation curve, and friction are evaluated independently and are added together.

Card 2 for GENERALIZED stiffness option.

Card 2	1	2	3	4	5	6	7	8
Variable	LCIDPH	LCIDT	LCIDPS	DLCIDPH	DLCIDT	DLCIDPS		
Type	I	I	I	I	I	I		
Default	none	none	none	none	none	none		

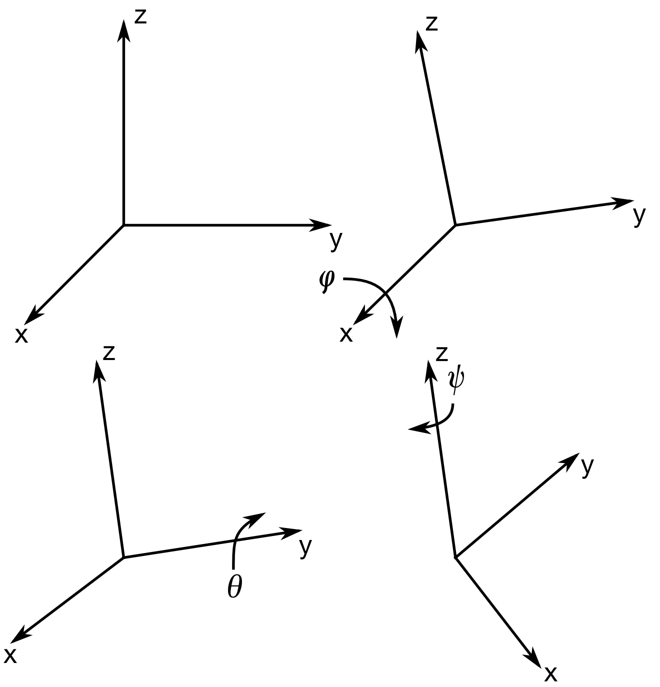


Figure 10-30. Definition of angles for the GENERALIZED joint stiffness.

Card 3 for GENERALIZED stiffness option.

Card 3	1	2	3	4	5	6	7	8
Variable	ESPH	FMPH	EST	FMT	ESPS	FMPS		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	0.0	0.0	0.0		

Card 4 for GENERALIZED stiffness option.

Card 4	1	2	3	4	5	6	7	8
Variable	NSAPH	PSAPH	NSAT	PSAT	NSAPS	PSAPS		
Type	F	F	F	F	F	F		
Default	0	0	0	0	0	0		

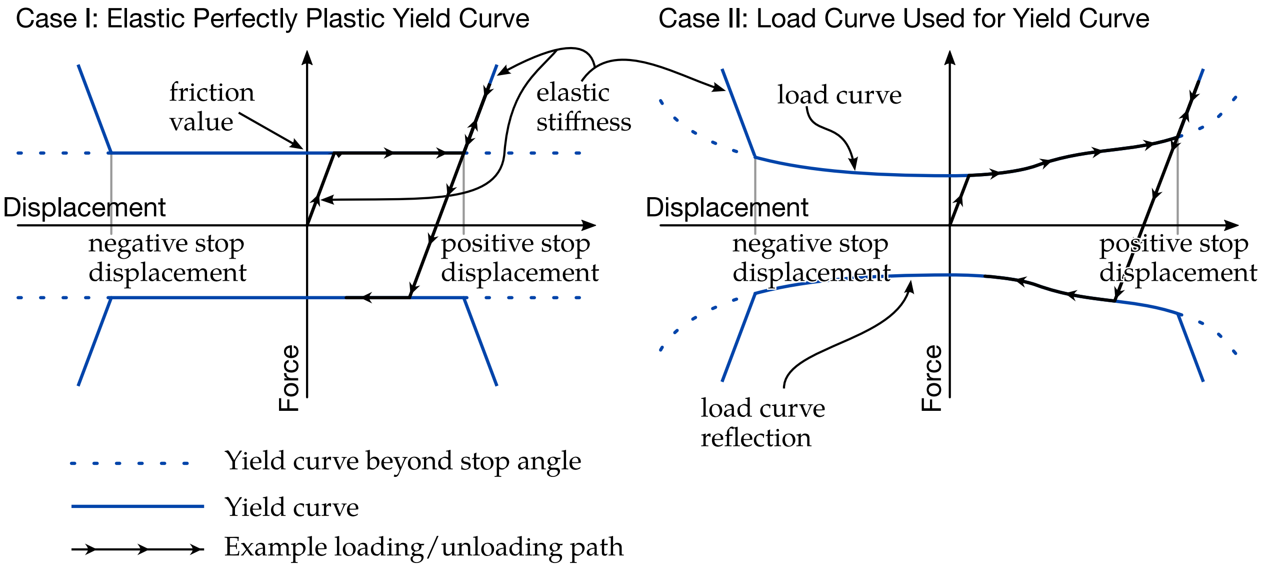

Case I: Elastic Perfectly Plastic Yield Curve
Case II: Load Curve Used for Yield Curve
friction value
elastic stiffness
load curve
Displacement
negative stop displacement
positive stop displacement
Displacement
negative stop displacement
positive stop displacement
Force
Force
load curve reflection
Yield curve beyond stop angle
Yield curve
Example loading/unloading path

Figure 10-31. *Friction model.* The friction model is motivated by plasticity and it is implemented for both rotational and translational joints. In the context of a rotational joint, the y-axis is to be interpreted as moment (rotational force) and the x-axis is to be interpreted as rotation. *Case I* (left) is activated by a positive friction value. *Case II* (right) is activated by a negative integer friction value, the absolute value of which specifies a load curve. *See the friction, elastic, and stop angle/displacement parameters from the input cards* (FM[var], ES[var], NSA[var], PSA[var]).

VARIABLE	DESCRIPTION
LCIDPH	Load curve ID for ϕ-moment versus rotation in radians. See Figure 10-30. If zero, the applied moment is set to 0.0. See *DEFINE_CURVE.
LCIDT	Load curve ID for θ-moment versus rotation in radians. If zero, the applied moment is set to 0.0. See *DEFINE_CURVE.
LCIDPS	Load curve ID for ψ-moment versus rotation in radians. If zero, the applied moment is set to 0.0. See *DEFINE_CURVE.
DLCIDPH	Load curve ID for ϕ-damping moment versus rate of rotation in radians per unit time. If zero, damping is not considered. See *DEFINE_CURVE.
DLCIDT	Load curve ID for θ-damping moment versus rate of rotation in radians per unit time. If zero, damping is not considered. See *DEFINE_CURVE.

VARIABLE **DESCRIPTION**

DLCIDPS

Load curve ID for ψ-damping torque versus rate of rotation in radians per unit time. If zero, damping is not considered. See *DEFINE_CURVE.

ESPH

Elastic stiffness per unit radian for friction and stop angles for ϕ rotation. If zero, friction and stop angles are inactive for ϕ rotation.

FMPH

Frictional moment limiting value for ϕ rotation. If zero, friction is inactive for ϕ rotation. This option may also be thought of as an elastic-plastic spring. If a negative value is input then the absolute value is taken as the load curve or table ID defining the yield moment versus ϕ rotation. A table permits the moment to also be a function of the joint reaction force and requires the specification of JID on card 1. See Figure 10-31.

EST

Elastic stiffness per unit radian for friction and stop angles for θ rotation. If zero, friction and stop angles are inactive for θ rotation. See Figure 10-31.

FMT

Frictional moment limiting value for θ rotation. If zero, friction is inactive for θ rotation. This option may also be thought of as an elastic-plastic spring. If a negative value is input then the absolute value is taken as the load curve or table ID defining the yield moment versus θ rotation. A table permits the moment to also be a function of the joint reaction force and requires the specification of JID on card 1.

ESPS

Elastic stiffness per unit radian for friction and stop angles for ψ rotation. If zero, friction and stop angles are inactive for ψ rotation.

FMPS

Frictional moment limiting value for ψ rotation. If zero, friction is inactive for ψ rotation. This option may also be thought of as an elastic-plastic spring. If a negative value is input then the absolute value is taken as the load curve or table ID defining the yield moment versus ψ rotation. A table permits the moment to also be a function of the joint reaction force and requires the specification of JID on card 1.

NSAPH

Stop angle in degrees for negative ϕ rotation. Ignored if zero. See Figure 10-31.

PSAPH

Stop angle in degrees for positive ϕ rotation. Ignored if zero.

VARIABLE	DESCRIPTION
NSAT	Stop angle in degrees for negative θ rotation. Ignored if zero.
PSAT	Stop angle in degrees for positive θ rotation. Ignored if zero.
NSAPS	Stop angle in degrees for negative ψ rotation. Ignored if zero.
PSAPS	Stop angle in degrees for positive ψ rotation. Ignored if zero.

Remarks:

After the stop angles are reached the torques increase linearly to resist further angular motion using the stiffness values on Card 3. Reasonable stiffness values have to be chosen. If the stiffness values are too low or zero, the stop will be violated.

If the initial local coordinate axes do not coincide, the angles, ϕ, θ, and ψ, will be initialized and torques will develop instantaneously based on the defined moment vs. rotation curves.

There are two methods available to calculate the rotation angles between the coordinate systems. For more information, see the JNTF parameter on *CONTROL_RIGID.

Card 2 for TRANSLATIONAL stiffness option.

Card 2	1	2	3	4	5	6	7	8
Variable	LCIDX	LCIDY	LCIDZ	DLCIDX	DLCIDY	DLCIDZ		
Type	I	I	I	I	I	I		
Default	none	none	none	none	none	none		

Card 3 TRANSLATIONAL stiffness option.

Card 3	1	2	3	4	5	6	7	8
Variable	ESX	FFX	ESY	FFY	ESZ	FFZ		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	0.0	0.0	0.0		

Card 4 for TRANSLATIONAL stiffness option.

Card 4	1	2	3	4	5	6	7	8
Variable	NSDX	PSDX	NSDY	PSDY	NSDZ	PSDZ		
Type	F	F	F	F	F	F		
Default	0	0	0	0	0	0		

VARIABLE	DESCRIPTION
LCIDX	Load curve ID for *x*-force versus *x*-distance between the origins of CIDA and CIDB based on the *x*-direction of CIDB. If zero, the applied force is set to 0.0. See *DEFINE_CURVE.
LCIDY	Load curve ID for *y*-force versus *y*-distance between the origins of CIDA and CIDB based on the *y*-direction of CIDB. If zero, the applied force is set to 0.0. See *DEFINE_CURVE.
LCIDZ	Load curve ID for *z*-force versus *z*-distance between the origins of CIDA and CIDB based on the *z*-direction of CIDB. If zero, the applied force is set to 0.0. See *DEFINE_CURVE.
DLCIDX	Load curve ID for *x*-damping force versus rate of *x*-translational displacement per unit time between the origins of CIDA and CIDB based on the *x*-direction of CIDB. If zero, damping is not considered. See *DEFINE_CURVE.

VARIABLE	DESCRIPTION
DLCIDY	Load curve ID for *y*-damping force versus rate of *y*-translational displacement per unit time between the origins of CIDA and CIDB based on the *y*-direction of CIDB. If zero, damping is not considered. See *DEFINE_CURVE.
DLCIDZ	Load curve ID for *z*-damping force versus rate of *z*-translational displacement per unit time between the origins of CIDA and CIDB based on the *z*-direction of CIDB. If zero, damping is not considered. See *DEFINE_CURVE.
ESX	Elastic stiffness for friction and stop displacement for *x*-translation. If zero, friction and stop angles are inactive for *x*-translation. See Figure 10-31.
FFX	Frictional force limiting value for *x*-translation. If zero, friction is inactive for *x*-translation. This option may also be thought of as an elastic-plastic spring. If a negative value is input then the absolute value is taken as the load curve ID defining the yield force versus *x*-translation. See Figure 10-31.
ESY	Elastic stiffness for friction and stop displacement for *y*-translation. If zero, friction and stop angles are inactive for *y*-translation.
FFY	Frictional force limiting value for *y*-translation. If zero, friction is inactive for y-translation. This option may also be thought of as an elastic-plastic spring. If a negative value is input then the absolute value is taken as the load curve ID defining the yield force versus *y*-translation.
ESZ	Elastic stiffness for friction and stop displacement for *z*-translation. If zero, friction and stop angles are inactive for *z*-translation.
FMZ	Frictional force limiting value for *z*-translation. If zero, friction is inactive for *z*-translation. This option may also be thought of as an elastic-plastic spring. If a negative value is input then the absolute value is taken as the load curve ID defining the yield force versus *z*-translation.
NSDX	Stop displacement for negative *x*-translation. Ignored if zero. See Figure 10-31.
PSDX	Stop displacement for positive *x*-translation. Ignored if zero.

VARIABLE	DESCRIPTION
NSDY	Stop displacement for negative *y*-translation. Ignored if zero.
PSDY	Stop displacement for positive *y*-translation. Ignored if zero.
NSDZ	Stop displacement for negative *z*-translation. Ignored if zero.
PSDZ	Stop displacement for positive *z*-translation. Ignored if zero.

Remarks:

After the stop displacements are reached the force increases linearly to resist further translational motion using the stiffness values on Card 3. Reasonable stiffness values must be chosen. If the stiffness values are too low or zero, the stop will be violated.

Example:

```
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$  *CONSTRAINED_JOINT_STIFFNESS_GENERALIZED
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Define a joint stiffness for the revolute joint described in
$    *CONSTRAINED_JOINT_REVOLUTE
$
$  Attributes of the joint stiffness:
$    - Used for defining a stop angle of 30 degrees rotation
$        (i.e., the joint allows a positive rotation of 30 degrees and
$           then imparts an elastic stiffness to prevent further rotation)
$    - Define between rigid body A (part 1) and rigid body B (part 2)
$    - Define a local coordinate system such that local x corresponds
$         to the joint’s axis of revolution and the angle phi is the angle
$         of rotation about that axis.
$    - The elastic stiffness per unit radian for the stop angle is 100.
$    - Variables left blank are not used during the simulation.
$
*CONSTRAINED_JOINT_STIFFNESS_GENERALIZED
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$     jsid      pida      pidb      cida      cidb
         1         1         2         5         5
$
$   lcidph     lcidt    lcidps   dlcidph    dlcidt   dlcidps

$
$     esph      fmps       est       fmt      esps      fmps
    100.0
$
$    nsaph     psaph      nsat      psat     nsaps     psaps
                30.0
$
$
*DEFINE_COORDINATE_NODES
$      cid        n1        n2        n3
```

```
         5         1         2         3
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
```

*CONSTRAINED_JOINT_USER_FORCE

Purpose: Define input data for a user subroutine to generate force resultants as a function of time and joint motion.

Card 1	1	2	3	4	5	6	7	8
Variable	FID	JID	NHISV					
Type	I	I	I					
Default	none	none	0					

User Subroutine Constants Cards. Define up to 48 optional user constants (6 cards total) for the user subroutine. This input is terminated after 48 constants are defined or when the next "*" keyword card is encountered.

Card 2	1	2	3	4	5	6	7	8
Variable	CONST1	CONST2	CONST3	CONST4	CONST5	CONST6	CONST7	CONST8
Type	F	F	F	F	F	I	I	F
Default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

VARIABLE	DESCRIPTION
FID	Joint user force ID.
JID	Joint ID for which this user force input applies.
NHISV	Number of history variables required for this definition. An array NHISV long is allocated and passed into the user subroutine. This array is updated in the user subroutine.
CONST*n*	A constant which is passed into the user subroutine.

CONSTRAINED_LAGRANGE_IN_SOLID_*{OPTION1}_{OPTION2}

Purpose: This command provides the coupling mechanism for modeling Fluid-Structure Interaction (FSI). The structure can be constructed from Lagrangian shell and/or solid entities. The multi-material fluids are modeled by ALE formulation.

Available options for *OPTION1* include:

<BLANK>

EDGE

This option may be used to allow the coupling between the edge of a shell part or part set and one or more ALE multi-material groups (AMMG). It accounts for the shell thickness in the coupling calculation. The edge thickness is the same as the shell thickness. This option only works when the Lagrangian slave set is defined as a part or a part set ID. It will not work for a slave segment set. One application of this option is a simulation of a Lagrangian blade (a shell part) cutting through some ALE material.

Available options for *OPTION2* include:

<BLANK>

TITLE

To define a coupling (card) ID number and title for each coupling card. If a title is not defined LS-DYNA will automatically create an internal title for this coupling definition. The ID number can be used to delete coupling action in a restart input deck via the *DELETE_FSI card.

Title Card. Additional card for the TITLE keyword option.

Optional	1	2	3	4	5	6	7	8
Variable	COUPID	TITLE						
Type	I	A70						

Card 1	1	2	3	4	5	6	7	8
Variable	SLAVE	MASTER	SSTYP	MSTYP	NQUAD	CTYPE	DIREC	MCOUP
Type	I	I	I	I	I	I	I	I
Default	none	none	0	0	0	2	1	0

Card 2	1	2	3	4	5	6	7	8
Variable	START	END	PFAC	FRIC	FRCMIN	NORM	NORMTYP	DAMP
Type	F	F	F	F	F	I	I	F
Default	0	1.0E10	0.1	0.0	0.5	0	0	0.0

Card 3	1	2	3	4	5	6	7	8
Variable	K	HMIN	HMAX	ILEAK	PLEAK	LCIDPOR	NVENT	IBLOCK
Type	F	F	F	I	F	I	I	I
Default	0.0	none	none	0	0.1	0	0	0

Card 4a. This card is required for CTYPE 11 & 12 but is otherwise optional.

Card 4a	1	2	3	4	5	6	7	8
Variable	IBOXID	IPENCHK	INTFORC	IALESOF	LAGMUL	PFACMM	THKF	
Type	I	I	I	I	F	I	F	
Default	0	0	0	0	0.0	0	0.0	

Porous Coupling Card 4b. This card applies only to CTYPE 11 & 12. If 4b is defined, 4a must be defined before 4b.

Card 4b	1	2	3	4	5	6	7	8
Variable	A1	B1	A2	B2	A3	B3		POREINI
Type	F	F	F	F	F	F		F
Default	0.0	0.0	0.0	0.0	0.0	0.0		0.0

Venting Geometry Card(s) 4c. These card(s) set venting geometry. It is repeated NVENT times (one card for defining each vent hole). It is defined only if NVENT > 0 in card 3. The last NVENT cards for *CONSTRAINED_LAGRANGE_IN_SOLID are taken to be Card(s) 4c, therefore, Cards 4a and 4b are not mandatory when Card(s) 4c are defined.

Card 4c	1	2	3	4	5	6	7	8
Variable	VENTSID	VENTYP	VTCOEF	POPPRES	COEFLC			
Type	I	I	I	F	I			
Default	0	0	0	0.0	0			

VARIABLE	DESCRIPTION
COUPID	Coupling (card) ID number. This ID can be used in a restart input deck to delete or reactivate this coupling action via the *DELETE_FSI card. If not defined, LSDYNA will assign an internal coupling ID based on the order of appearance in the input deck.
TITLE	A description of this coupling definition (A70).
SLAVE	Slave set ID defining a part, part set or segment set ID of the Lagrangian or slave structure (see *PART, *SET_PART or *SET_SEGMENT). See Remark 1.
MASTER	Master set ID defining a part or part set ID of the ALE or master solid elements (see *PART or *SET_PART, and see Remark 1).

VARIABLE **DESCRIPTION**

SSTYP

Slave set type of "SLAVE" (see Remark 1):

EQ.0: part set ID (PSID).

EQ.1: part ID (PID).

EQ.2: segment set ID (SGSID).

MSTYP

Master set type of "MASTER" (see Remark 1):

EQ.0: part set ID (PSID).

EQ.1: part ID (PID).

NQUAD

Number of coupling points distributed over each coupled Lagrangian surface segment. (see Remark 2)

EQ.0: NQUAD will be set by default to 2,

GT.0: An NQUAD × NQUAD coupling points distribution over each Lagrangian segment is defined,

LT.0: NQUAD is reset to a positive value. Coupling at nodes is obsolete.

CTYPE

Fluid-Structure coupling method. The constraint methods (1, 2, and 3) are not supported in MPP.

EQ.1: constrained acceleration.

EQ.2: constrained acceleration and velocity (default, see Remark 3).

EQ.3: constrained acceleration and velocity, normal direction only.

EQ.4: penalty coupling for shell (with or without erosion) and solid elements (without erosion).

> **NOTE:** For RIGID slave PARTS a penalty coupling method (CTYPE=4) must be used, see parameter CTYPE below.

EQ.5: penalty coupling allowing erosion in the Lagrangian entities (solid elements and thick shells).

EQ.6: penalty coupling designed for airbag modeling which automatically controls the DIREC parameter internally. It is equivalent to setting {CTYPE = 4; DIREC = 1} for unfolded region; and {CTYPE = 4; DIREC = 2}; in fold-

VARIABLE **DESCRIPTION**

ed region. For both cases: {ILEAK = 2; FRCMIN = 0.3}.

EQ.11: coupling designed to couple Lagrangian porous shell to ALE material. When this option is used, THKF, the 7th column parameter of optional card 4a and the first 2 parameters of optional card 4b must be defined. See *LOAD_BODY_POROUS and remark 13 below.

EQ.12: coupling designed to couple Lagrangian porous solid to ALE material. When this option is used, Ai & Bi parameters of optional card 4b must be defined (card 4a must be defined but can be blank). See *LOAD_-BODY_POROUS and Remark 14 below

DIREC

For CTYPE=4, 5, or 6

Coupling direction: (see Remark 4)

EQ.1: normal direction, compression and tension (default)

EQ.2: normal direction, compression only

EQ.3: all directions

For CTYPE=12

Flag to activate an element coordinate system:

EQ.0: The forces are applied in the global directions.

EQ.1: The forces are applied in a local system attached to the Lagrangian solid. The system is consistent with AOPT = 1 in *LOAD_BODY_POROUS. (See Remark 14).

MCOUP

For CTYPE = 4, 5, 6, 11, or 12

Multi-material option: (see Remark 5)

EQ.0: couple with all multi-material groups,

EQ.1: couple with material with highest density.

LT.0: MCOUP must be an integer. -MCOUP refers to a set ID of an ALE multi-material group. See *SET_MULTI-MATERIAL_GROUP.

START

Start time for coupling.

END

End time for coupling. If less than zero, coupling will be turned off during dynamic relaxation. After dynamic relaxation phase is finished, the absolute value will be taken as end time.

VARIABLE DESCRIPTION

PFAC

For CTYPE = 4,5 or 6

Penalty factor. PFAC is a scale factor for scaling the estimated stiffness of the interacting (coupling) system. It is used to compute the coupling forces to be distributed on the slave and master parts

GT.0: Fraction of estimated critical stiffness.

LT.0: PFAC must be an integer, and -PFAC is a load curve ID. The curve defines the coupling pressure on the y-axis as a function of the penetration along the *x*-axis. (See "How to Correct Leakage")

For CTYPE = 11 or 12

Time step factor

FRIC

Coefficient of friction (used with DIREC = 1 and 2 only).

FRCMIN

Minimum volume fraction of a coupled ALE multi-material group (AMMG) or fluid in a multi-material ALE element to activate coupling. Default value is 0.5. Reducing FRCMIN (typically, between 0.1 and 0.3) would turn on coupling earlier to prevent leakage in high velocity impact cases.

NORM

A Lagrangian segment will couple to fluid on only one side of the segment. NORM determines which side. See Remark 6.

EQ.0: Couple to fluid (AMMG) on head-side of Lagrangian segment normal vector.

EQ.1: Couple to fluid (AMMG) on tail-side of Lagrangian segment normal vector.

NORMTYP

Penalty coupling spring (or force) direction (DIREC = 1, or 2):

EQ.0: normal vectors are interpolated from nodal normals. (default).

EQ.1: normal vectors are interpolated from segment normals. This is sometimes a little more robust for sharp Lagrangian corners, and folds.

DAMP

Damping factor for penalty coupling. This is a coupling-damping scaling factor. Typically it may be between 0 and 1 (see Remark 7).

VARIABLE	DESCRIPTION
K	Thermal conductivity of a virtual fluid between the slave surface and the master material (see Remark 8).
HMIN	The absolute value is minimum air gap in heat transfer, h_{min} (See Remark 8). LT.0: turn on constraint based thermal nodal coupling between LAG structure and ALE fluids. GE.0: minimum air gap. If zero, default to 1.0e-6.
HMAX	Maximum air gap in heat transfer, h_{max}. There is no heat transfer above this value (see Remark 8).
ILEAK	Coupling leakage control flag (Remark 9): EQ.0: none (default), EQ.1: weak, leakage control is turned off if $$\text{penetrating volume fraction} > \text{FRCMIN} + 0.2$$ EQ.2: strong, with improved energy consideration. Leakage control is turned off if $$\text{penetrating volume fraction} > \text{FRCMIN} + 0.4$$
PLEAK	Leakage control penalty factor, 0 < PLEAK < 0.2 is recommended. This factor influences the additional coupling force magnitude to prevent leakage. It is conceptually similar to PFAC. Almost always, the default value (0.1) is adequate.
LCIDPOR	If this is a positive integer: A load curve ID (LCID) defining porous flow through coupling segment: $$\text{Abscissa} = x = (P_{\mathrm{up}} - P_{\mathrm{down}})$$ $$\text{Ordinate} = y = \text{relative porous fluid velocity}$$ Where P_{up} and P_{down} are, respectively, the upstream and downstream pressures across of the porous coupling segment. The relative porous velocity is the ALE fluid velocity relative to the moving Lagrangian segment. This experimental data curve must be provided by the user. If LCIDPOR is a negative integer: The porous flow is controlled by the parameters FLC, FAC, ELA under *MAT_FABRIC card.

VARIABLE **DESCRIPTION**

> **CAUTION:** The pressure under the FAC load curve is "absolute upstream pressure" (see Remark 10).

Abscissa = x = absolute upstream pressure

Ordinate = y = relative porous fluid velocity

For CTYPE = 11 or CTYPE = 12 and POREINI = 0.0:

LT.0: The load curve |LCIDPOR| is a factor versus time of the porous force computed by the Ergun equation (see Remark 13).

GT.0: The load curve LCIDPOR is a porous force versus velocity, which replaces the force computed by the Ergun equation (see Remark 13).

For CTYPE = 11 or CTYPE = 12 and POREINI > 0.0:

NE.0: The load curve |LCIDPOR| is a factor versus time of the porous force computed by the Ergun equation (see Remark 13).

NVENT The number of vent surface areas to be defined. Each venting flow surface is represented by one or more Lagrangian segments (or surfaces).

For airbag applications, this may be referred to as "isentropic" venting where the isentropic flow equation is used to compute the mass flow rate based on the ratio of the upstream and downstream pressures $P_{\mathrm{up}}/P_{\mathrm{down}}$.

For each of the NVENT vent surfaces, an additional card of format 4c defining the geometrical and flow properties for each vent surface will be read in.

The vented mass will simply be deleted from the system and cannot be visualized as in the case of physical venting (see Remark 11).

VARIABLE **DESCRIPTION**

IBLOCK

Flag to control the venting (or porous) flow blockage due to Lagrangian contact during ALE computation.

EQ.0: Off

EQ.1: On

The venting definition is defined in this command. However, the venting flow may be defined via either the LCIDPOR parameter in this command or via the *MAT_FABRIC parameters (FLC, FAC, ELA). However, note that FVOPT (blocking) parameter under *MAT_FABRIC applies only to CV computation.

IBOXID

A box ID defining a box region in space in which ALE coupling is activated.

GT.0: At time = 0.0, the Lagrangian segments inside this box are remembered. In subsequent coupling computation steps, there is no need to search for the Lagrangian segments again.

LT.0: At each FSI bucketsort, the Lagrangian segments inside this box are marked as active coupling segments. This makes the coupling operate more efficiently when structure mesh is approaching ALE domain, i.e. hydroplaning, bird strike, etc.

IPENCHK

Only for CTYPE = 4

Initial penetration check flag (See Remark 12):

EQ.0: Do not check for initial penetration.

EQ.1: Check and save initial ALE material penetration across a Lagrangian surface (d0), but do not activate coupling at t = 0. In subsequent steps (t > 0) the actual penetration is computed as follows:

$$\underbrace{\text{Actual Penetration}}_{d_a} = \underbrace{\text{Total Penetration}}_{d_T} - \underbrace{\text{Initial Penetration}}_{d_0}$$

INTFORC

A flag to turn on or off the output of ALE coupling pressure and forces on the slave Lagrangian segments (or surfaces).

EQ.0: Off

EQ.1: On

Note that the coupling pressures and forces are computed based

VARIABLE **DESCRIPTION**

on the coupling stiffness reponse to the ALE fluid penetration.

When INFORC = 1 and a *DATABASE_BINARY_FSIFOR (DBF) card is defined, LS-DYNA writes out the segment coupling pressure and forces to the binary interface force file for contour plotting. This interface force file must be given a name on the execution line, for example:

```
ls-dyna  i=inputfilename.k … h=interfaceforcefilename
```

The time interval between output is defined by "dt" in the DBF card. To plot the binary data in this file:

```
ls-prepost interfaceforcefilename
```

IALESOF An integer flag to turn ON/OFF a supplemental Lagrange multiplier FSI constraint which provides a coupling force in addition to the basic penalty coupling contribution. This is a hybrid coupling method.

EQ.0: OFF (default).

EQ.1: Turn ON the hybrid Lagrange-multiplier method. LAGMUL multiplier factor is read.

LAGMUL A Lagrange multiplier factor with a range between 0.0 and 0.05 may be defined. A typical value may be 0.01. This should never be greater than 0.1.

EQ.0: OFF (default).

GT.0: Turn ON the Lagrange-multiplier method and use LAGMUL as a coefficient for scaling the penalty factor.

PFACMM Mass-based penalty stiffness factor computational options. This works in conjunction with PFAC = constant (not a load curve). The coupling penalty stiffness (CPS) is computed based on an estimated effective coupling mass.

EQ.0: CPS $\propto$ PFAC $\times \min(m_{\text{slave}}, m_{\text{master}})$, default.

EQ.1: CPS $\propto$ PFAC $\times \max(m_{\text{slave}}, m_{\text{master}})$.

EQ.2: CPS $\propto$ PFAC $\times \sqrt{m_{\text{slave}} m_{\text{master}}}$, geometric-mean of the masses.

EQ.3: CPS $\propto$ PFAC $\times K_{\text{Lagrangian}}$ where K is the bulk modulus of the slave or Lagrangian part

VARIABLE	DESCRIPTION

THKF

For all CTYPE choices except 11:

A flag to account for the coupling thickness of the Lagrangian shell (slave) part.

LT.0: Use positive value of |THKF| for coupling segment thickness.

EQ.0: Do not consider coupling segment thickness.

GT.0: Coupling segment thickness scale factor.

For CTYPE = 11:

This thickness is required for volume calculation.

GT.0: (Fabric) Thickness scale factor. The base shell thickness is taken from the *PART definition.

LT.0: User-defined (Fabric) thickness. The fabric thickness is set to |THKF|.

A1

Viscous coefficient for the porous flow Ergun equation (see Remark 13).

GT.0:

For CTYPE = 11

$$\mathrm{A1} = A_n$$

which is the coefficient for normal-to-segment direction.

For CTYPE = 12

$$\mathrm{A1} = A_x$$

which is the coefficient for the x-direction in the coordinate system specified by DIREC.

LT.0: If POREINI = 0.0, the coefficient is time dependent through a load curve id defined by |A1|. If POREINI > 0.0, the coefficient is porosity dependent through a load curve id defined by |A1|. The porosity is defined by PORE (see POREINI).

B1

Inertial coefficient for the porous flow Ergun equation (see Remark 13).

GT.0:

For CTYPE = 11

$$\mathrm{B1} = B_n$$

VARIABLE	DESCRIPTION
	which is the coefficient for normal-to-segment direction. For CTYPE = 12 $$\mathrm{B1} = B_x$$ which is the coefficient for the x-direction of a coordinate system specified by DIREC. LT.0: If POREINI = 0.0, the coefficient is time dependent through a load curve id defined by |B1|. If POREINI > 0.0, the coefficient is porosity dependent through a load curve id defined by |B1|. The porosity is defined by PORE (see POREINI).
A2	For CTYPE = 12 Viscous coefficient for the porous flow Ergun equation (see Remark 14). GT.0: Coefficient for the y-direction of a coordinate systems specified by DIREC. $$\mathrm{A2} = A_y$$ LT.0: If POREINI = 0.0, the coefficient is time dependent through a load curve id defined by |A1|. If POREINI > 0.0, the coefficient is porosity dependent through a load curve id defined by |A2|. The porosity is defined by PORE (see POREINI).
B2	For CTYPE=12 Inertial coefficient for the porous flow Ergun equation (see Remark 14). GT.0: Coefficient for the y-direction of a coordinate system specified by DIREC. $$\mathrm{B2} = B_y$$ LT.0: If POREINI = 0.0 and B2 < 0, the coefficient is time dependent through a load curve id defined by |B2|. If POREINI > 0.0 and B2 < 0, the coefficient is porosity dependent through a load curve id defined by |B2|. The porosity is defined by PORE (see POREINI).
A3	For CTYPE = 12 Viscous coefficient for the porous flow Ergun equation (see Remark 14).

VARIABLE **DESCRIPTION**

GT.0: Coefficient for the z-direction of a coordinate system specified by DIREC.

$$A3 = A_z$$

LT.0: If POREINI = 0.0 and A3 < 0, the coefficient is time dependent through a load curve id defined by |A3|. If POREINI > 0.0 and A3 < 0, the coefficient is porosity dependent through a load curve id defined by |A3|. The porosity is defined by PORE (see POREINI).

B3

For CTYPE = 12

Inertial coefficient for the porous flow Ergun equation (see Remark 14).

GT.0: Coefficient for the z-direction of a coordinate system specified by DIREC.

$$B3 = B_z$$

LT.0: If POREINI = 0.0 and B3 < 0, the coefficient is time dependent through a load curve id defined by |B3|. If POREINI > 0.0 and B3 < 0, the coefficient is porosity dependent through a load curve id defined by |B3|. The porosity is defined by PORE (see POREINI).

POREINI

For CTYPE = 11 or CTYPE = 12

POREINI is the initial volume of pores in an element. The current volume is

$$\text{PORE} = \text{POREINI} \times \frac{v(t)}{v(t_0)}$$

where $v(t)$ and $v(t_0)$ are the current and initial element volumes respectively.

VENTSID

Set ID of the vent hole shape.

VENTYP

Vent surface area set ID type:

EQ.0: Part set ID (PSID).

EQ.1: Part ID (PID).

EQ.2: Segment set ID (SGSID).

VTCOEF

Flow coefficient for each vent surface area.

VARIABLE	DESCRIPTION
POPPRES	Venting pop pressure limit. If the pressure inside the airbag is lower than this pressure, then nothing is vented. Only when the pressure inside the airbag is greater than POPPRES that venting can begin.
COEFLC	A time-dependent multiplier load curve for correcting the vent flow coefficient, with values ranging from 0.0 to 1.0.

Best Practices:

Due to the complexity of this card, some comments on simple, efficient and robust coupling approach are given here. These are not rigid guidelines, but simply some experience-based observations.

1. **Definition (Fluid and Structure).** The term *fluid*, in the Fluid-Structure Interaction (FSI), refers to materials with ALE element formulation, not indicating the phase (solid, liquid or gas) of those materials. In fact, solid, liquid and gas can all be modeled by the ALE formulation. The term *structure* refers to materials with Lagrangian element formulation.

2. **Default Values (CTYPE and MCOUP).** In general, penalty coupling (CTYPE 4 & 5) is recommended, and MCOUP=negative integer is a better choice to define a specific ALE multi-material group (AMMG) to be coupled to the Lagrangian surface. At the minimum, all parameters on card 1 are to be specified, and the default values for most are good starting choices (except MCOUP).

3. **How to Correct Leakage.** If there is leakage, PFAC, FRCMIN, NORMTYPE and ILEAK are the 4 parameters that can be adjusted.

   a) For hard structure (steel) and very compressible fluid (air), PFAC may be set to 0.1 (or higher). PFAC = constant value.

   b) Next, keeping PFAC = constant and set PFACMM = 3 (optional card 4a). This option scales the penalty factor by the bulk modulus of the Lagrangian structure. This new approach has also shown to be effective for some airbag application.

   c) The next approach may be switching from constant PFAC to a load curve approach (i.e. PFAC = load curve, and PFACMM = 0). By looking at the pressure in the system near leakage original location, we can get a feel for the pressure required to stop it.

d) If leakage persists after some iterations on the coupling force controls, one can subsequently try to set ILEAK = 2 in combination with the other controls to prevent leakage.

e) If the modifications fail to stop the leakage, maybe the meshes have to be redesigned to allow better interactions between the Lagrangian and Ale materials.

In the example below, the underlined parameters are usually defined parameters. A full card definition is shown for reference.

```
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
*CONSTRAINED_LAGRANGE_IN_SOLID
$    SLAVE    MASTER     SSTYP     MSTYP     NQUAD     CTYPE     DIREC     MCOUP
         1        11         0         0         4         4         2      -123
$    START       END      PFAC      FRIC    FRCMIN      NORM  NORMTYPE      DAMP
       0.0       0.0       0.1      0.00       0.3         0         0       0.0
$       CQ      HMIN      HMAX     ILEAK     PLEAK   LCIDPOR     NVENT    IBLOCK
         0         0         0         0       0.0         0         0         0
$4A IBOXID   IPENCHK   INTFORC   IALESOF    LAGMUL    PFACMM      THKF
$        0         0         0         0         0         0         0
$4B     A1        B1        A2        B2        A3        B3
$      0.0       0.0       0.0       0.0       0.0       0.0
$4C VNTSID   VENTYPE  VENTCOEF   POPPRES  COEFLCID  (STYPE:0=PSID;1=PID;2=SGSID)
$        0         0         0       0.0         0
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|...8
```

Remarks:

1. **Meshing.** In order for a fluid-structure interaction (FSI) to occur, a Lagrangian (structure or slave) mesh must spatially overlap with an ALE (fluid or master) mesh. Each mesh should be defined with independent node IDs. LS-DYNA searches for the spatial intersection of between the Lagrangian and ALE meshes. Where the meshes overlap, there is a possibility that interaction may occur. In general, SLAVE, MASTER, SSTYP and MSTYPE are required definitions for specifying overlapping-domains coupling search.

2. **Number of Coupling Points.** The number of coupling points, NQUAD × NQUAD, is distributed over the surface of each Lagrangian segment. Generally, 2 or 3 coupling points per each Eulerian/ALE element width is adequate. Consequently, the appropriate NQUAD values must be estimated based on the *relative resolutions* between the Lagrangian and ALE meshes.

   For example, if 1 Lagrangian shell element spans 2 ALE elements, Then NQUAD for each Lagrangian segment should be 4 or 6. Alternatively, if 2 or 3 Lagrangian segments span 1 ALE element, then maybe NQUAD = 1 would be adequate.

   If either mesh compresses or expands during the interaction, the number of coupling points per ALE element will also change. The user must account for this and try to maintain at least 2 coupling points per each ALE element side length during the whole process to prevent leakage. Too many coupling points can result in instability, and not enough can result in leakage.

3. **The Constraint Method.** The constraint method violates kinetic energy balance. The penalty method is therefore recommended. Historically, CTYPE=2 was sometimes used to couple Lagrangian beam nodes to ALE or Lagrangian solids, e.g., for modeling rebar in concrete, or tire cords in rubber. For such constraint-based coupling of beams in solids, *CONSTRAINED_BEAM_IN_SOLID is now preferred.

4. **Coupling Direction.** DIREC=2 (compression only) may be generally a more stable and robust choice for coupling direction. However, the physics of the problem should dictate the coupling direction. DIREC=1 couples under both tension and compression. This is sometimes useful; for example, in the case of a suddenly accelerating liquid in a container. DIREC=3 is rarely appropriate because it models an extremely sticky fluid.

5. **Multi-material Coupling Option.** When MCOUP is a negative integer; for example MCOUP= -123, then an ALE multi-material set-ID (AMMSID) of 123 must exist. This is an ID defined by a *SET_MULTI-MATERIAL_GROUP_LIST card. This generally seems to be a better approach to couple to a specific set of AMMGs, and have a clearly defined fluid interface interacting with a Lagrangian surface. That way, any leakage may be visualized and the penalty force can be computed more precisely.

   The *Couple to all materials* option as activated by MCOUP = 0 is generally not recommended. LS-DYNA calculates the fluid coupling interface as the surface where the sum of coupled ALE materials occupies a volume fraction (Vf) equal to 50%. Since MCOUP = 0 couples to all materials, the sum of all coupled ALE materials is, in this case, trivially 100%. Consequently, when MCOUP = 0 there will not be a fluid interface with which to track leakage.

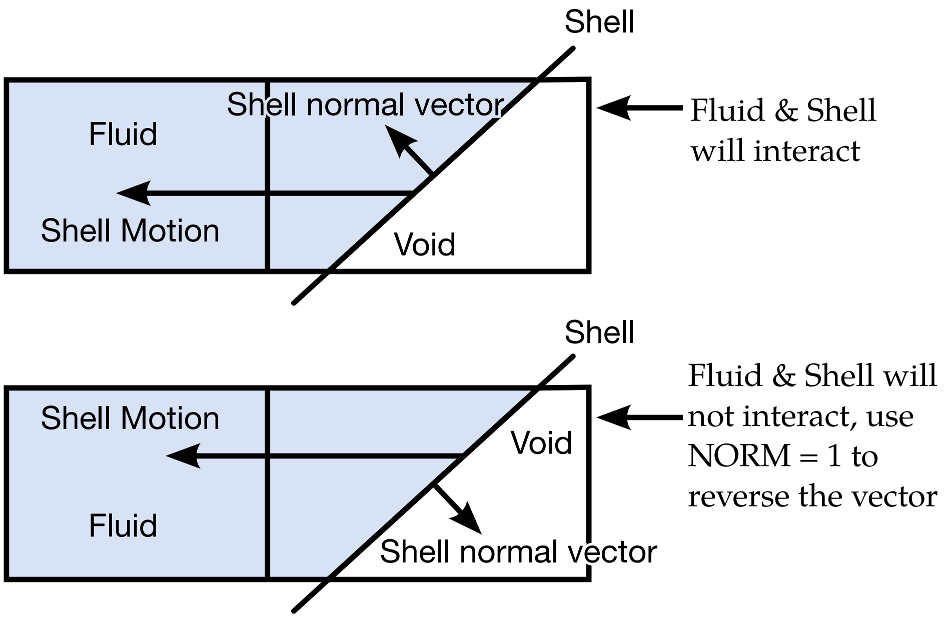

Shell
Fluid
Shell normal vector
Shell Motion
Void
Fluid & Shell will interact
Shell
Shell Motion
Void
Fluid
Shell normal vector
Fluid & Shell will not interact, use NORM = 1 to reverse the vector

Figure 10-32. Shell Motion

6. **Normal Vector Direction.** The normal vectors (NV) of a Lagrangian shell part are defined by the order of the nodes in *ELEMENT definitions, via the right hand rule, and for a segment set, the order of nodes defined in *SET_SEGMENT. Let the side pointed to by NV be “positive”. The penalty method measure penetration as the distance the ALE fluid penetrates from the positive side to the negative side of the Lagrangian segment. Only fluid on the positive side will be “seen” and coupled to.

   Therefore, all normal vectors of the Lagrangian segments should point uniformly toward the ALE fluid(s), AMMGs, to be coupled to. If NV point uniformly away from the fluid, coupling is not activated. In this case, coupling can be activated by setting NORM = 1. Sometimes a shell part or mesh is generated such that its normal vectors do not point uniformly in a consistent direction (all toward the inside or outside of a container, etc.) The user should always check for the normal vectors of any Lagrangian shell part interacting with any fluid. The NORM parameter may be used to flip the normal direction of all the segments included in the Lagrangian slave set. See Figure 10-32.

7. **Coupling-Damping Factor.** The user-input coupling-damping factor (DAMP) is used to scale down the critical-damping force (~ damper constant × velocity). For a mass-to-rigid-wall system connected by a parallel-spring-damper connector, we can obtain solution for a critically-damped case. DAMP is a factor for scaling down the amount of damping, with DAMP=1 being a critically-damped case.

8. **Heat Transfer.** The method used is similar to that done by *CONTACT_…_THERMAL_… card, except radiation heat transfer is not considered. A gap, l, is assumed to exist between the 2 materials undergoing heat transfer (one is Lagrangian and the other ALE). The convection heat transfer in the gap is assumed to approach simple conduction across the medium in the gap.

$$q = K\frac{dT}{dx} \sim h\Delta T \Rightarrow h \sim \frac{K}{l}$$

The heat flux is typically defined as an energy transfer rate per unit area, $q \sim \frac{[J/s]}{m^2}$. The constant K is the thermal conductivity of the material in the gap; h, is the equivalent convection heat transfer coefficient; and ΔT is the temperature difference between the master and slave sides. There are 3 possible scenarios:

$$h \sim \begin{cases} 0 & \text{HMAX} < l \\ K/l & \text{HMIN} \le l \le \text{HMAX} \\ K/\text{HMIN} & 0 < l < \text{HMIN} \end{cases}$$

The ALE fluid must be modelled using the ALE single material with void element formulation (ELFORM = 12) because the LS-DYNA thermal solver supports only one temperature per node. However, a workaround enables partial support for ELFORM =11. Rather than using the thermal solver's nodal temperature field, the ALE temperature is derived from element's internal energy using the heat capacity. The heat is then extracted from or added to the internal energy of ALE elements. This feature was implemented to calculate the heat exchange between a gas mixture, modeled with *MAT_GAS_MIXTURE and ALE multi-material formulation ELFORM = 11, and a Lagrangian container.

HMIN < 0 turns on constraint-based thermal nodal coupling between the Lagrangian surface nodes and ALE fluid nodes. This option only works with ALE single material with void element formulation (ELFORM = 12). Once a Lagrangian surface node is in contact with ALE fluid (gap = 0), the heat transfer described above is turned off. Instead the Lagrangian surface node temperature is constrained to the ALE fluid temperature field.

9. **Leakage Control.** The dominate force preventing leakage across a coupled Lagrangian surface should be the penalty associated with the coupling. Forces from the leakage control algorithm feature should be secondary. The *DATABASE_FSI keyword controls the "dbfsi" file, which reports both the coupling forces and the leakage control force contribution. It is useful for debugging and fine-tuning.

ILEAK = 2 conserves energy; thus, it is better for airbag applications. Leakage control should only be enabled when (1) coupling to a specific AMMG (MCOUP as a negative integer) is activated, and (2) the fluid interface is clearly defined and tracked through the *ALE_MULTI-MATERIAL_GROUP card.

10. **Pressure Definition in Porous Flow.** There are currently two methods to model porous flow across a Lagrangian shell structure. Both methods involve defining an empirical data curve of relative porous gas velocity as a function of system pressure. However the pressure definitions are slightly different depending on the choice of parameter defined:

    a) When porous flow is modelled using the LCIDPOR parameter (part of *this* keyword), the velocity response curve expected to be given in terms of the pressure difference: $P_{\mathrm{upstream}} - P_{\mathrm{downstream}}$.

    b) When LCIDPOR is negative, porous flow is modelled using the *MAT_-FABRIC material model. The FAC field in *MAT_FABRIC contains a load curve ID given in terms of absolute upstream pressure, rather than in terms of the pressure difference.

    The *AIRBAG_ALE keyword assumes that the curve referenced by FAC in *MAT_FABRIC is given in terms of absolute upstream pressure. These absolute pressure data are *required* for the CV phase. During the ALE phase, LS-DYNA automatically shifts the FAC curve left (negative) by 1 atmospheric pressure for the porous coupling calculation, which uses gauge pressure, rather than absolute pressure.

    The mass flowing across a porous Lagrangian surface can be tracked by the “pleak” parameter of the optional “dbfsi” ASCII output file, which may be enabled with the *DATABASE_FSI keyword.

11. **Venting.** There are 2 methods to model (airbag) venting. The accumulated mass output of both may be tracked via the *DATABASE_FSI card (“pleak” parameter in the “dbfsi” ASCII output file).

    a) **Isentropic Venting.** In isentropic venting, (define NVENT on card 3) the flow crossing the vent hole surface is estimated from the isentropic equation. All airbag shell normal vectors should point uniformly in the same direction: typically, inward. The shell elements for the vent holes, included in the Lagrangian coupling set, should also point in the same direction as the airbag meaning usually inward. For more details on isentropic venting see *AIRBAG_WANG_NEFSKE mass flow rate equation for option OPT EQ.1 and 2.

    b) **Physical Venting.** Physical venting models involve holes in the Lagrangian structure (usually airbags). The shell parts representing the vent holes may be either excluded from the Lagrangian coupling set, or, if included, have normal vectors reversed from the rest of the airbag. Typically, this means the holes having outward facing normal vectors, since the rest of the airbag has inward pointing normal vectors. With either approach the holes produce no coupling force to stop fluid leakage.

When a particular AMMG is present on both sides of the same Lagrangian shell surface, penalty coupling can break down. Therefore, It is recommended that *ALE_FSI_SWITCH_MMG_ID be used to switch the AMMG ID of the vented gas so that the vented gas outside the bag does not lead to leakage.

12. **Initial Penetration Check.** Typically, penetration check (IPENCHK) should only be used if there is high coupling force applied at t=0. For example, consider a Lagrangian container, filled with non-gaseous fluid (i.e. ALE liquid or solid) via the *INITIAL_VOLUME_FRACTON_GEOMETRY command. Sometimes due to mesh resolution or complex container geometry, there is initial penetration of the fluid across the container surface. This can give rise to a sharp and immediate coupling force on the fluid at t=0. Turning on IPENCHK may help eliminate this spike in coupling force.

13. **Porous Flow for Shell Elements.** For shell, CTYPE=11, the Ergun-type empirical porous flow equation is applied to the normal flow direction across the porous surface. The pressure gradient along the segment normal direction is

$$\frac{dP}{dx_n} = A_n(\varepsilon,\mu)V_n + B_n(\varepsilon,\rho)|V_n|V_n$$

where the subscript "n" refers to the direction normal to the porous Lagrangian shell surface and where,

a) V_n is the relative normal-to-porous-shell-surface fluid velocity component.

b) $A_n(\varepsilon,\mu) = A_1(\varepsilon,\mu)$ is a viscous coefficient of the Ergun-type porous flow equation. As applied here it should contain the fluid dynamic viscosity, μ, and shell porosity, ε information.

c) $B_n(\varepsilon,\rho) = B_1(\varepsilon,\rho)$ is an inertial coefficient of the Ergun-type porous flow equation. As applied here it should contain the fluid density, ρ, and shell porosity, ε, information.

The force increment applied per segment is

$$F_n = \frac{d\rho}{dx_n} \times \mathrm{THKF} \times S,$$

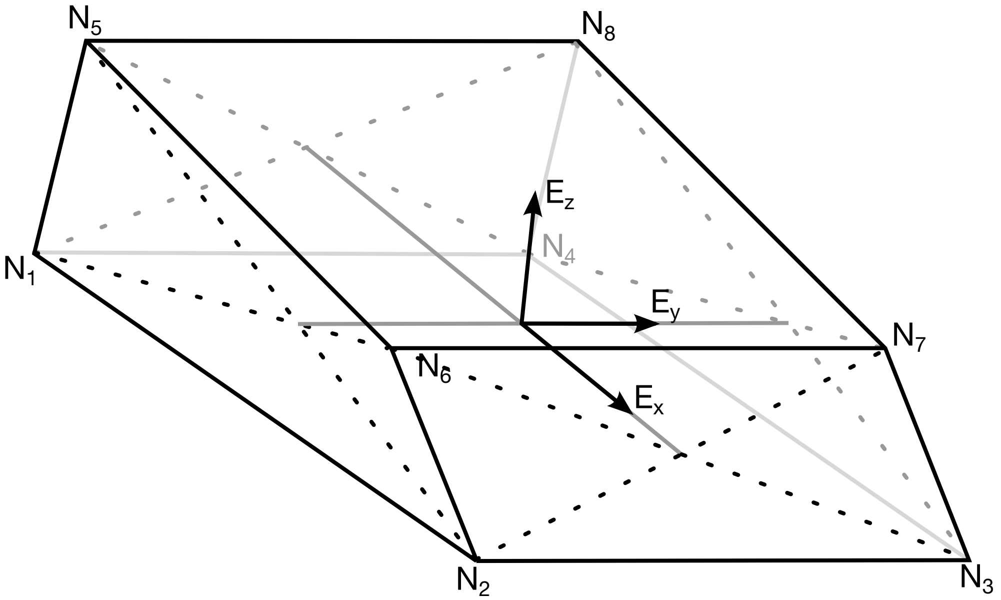


Figure 10-33. The E_x direction is aligned along the line segment connecting the centers of the 2-3-6-7 and the 1-4-8-5 faces. The E_y direction is orthogonal to the E_x direction and in the plane containing both E_x and containing the segment connecting the centers of the 1-2-6-5 and 3-4-8-7 faces. The E_z is normal to this plane.

where, S is the segment surface area.

If *DEFINE_POROUS_LAGRANGIAN defines the porous properties of a slave element, the porous forces are computed with an equation similar to the one used in *LOAD_BODY_POROUS

> **NOTE:** $A_i(\varepsilon,\mu)$, $B_i(\varepsilon,\rho)$, and THKF are required input for porous shell coupling.

14. **Porous Flow for Solid Elements.** For porous solid, CTYPE=12, the pressure gradient along each global direction (i) can be computed similarly.

$$\frac{dP}{dx_i} = A_i(\varepsilon,\mu)V_i + B_i(\varepsilon,\rho)|V_i|V_i \text{ for } i = 1,2,3$$

Where,

a) V_i is the relative fluid velocity component through the porous solid in the 3 global directions.

b) $A_i(\varepsilon,\mu)$ is a viscous coefficient of the Ergun-type porous flow equation in the ith direction. As applied here it should contain the fluid dynamic viscosity, μ, and shell porosity, ε, information.

c) $B_i(\varepsilon,\rho)$ is an inertial coefficient of the Ergun-type porous flow equation in the i^th^ direction. As applied here it should contain the fluid density (ρ) and solid porosity (ε) information.

> **NOTE:** $A_i(\varepsilon,\mu)$, and $B_i(\varepsilon,\rho)$ are required input for porous solid coupling.

If DIREC = 1, the pressure gradient in a solid is applied in a local reference coordinate system defined in Figure 10-33. If *DEFINE_POROUS_LAGRANGIAN defines the porous properties of a slave element, the local system can be adapted and the porous forces are computed with an equation similar to the one used in *LOAD_BODY_POROUS.

*CONSTRAINED_LINEAR_GLOBAL

Purpose: Define linear constraint equations between displacements and rotations, which can be defined in global coordinate systems.

Card 1	1	2	3	4	5	6	7	8
Variable	LCID							
Type	I							
Default	none							

DOF Card. Define one card for each constrained degree-of-freedom. Input is terminated when a "*" card is found.

Card 2	1	2	3	4	5	6	7	8
Variable	NID	DOF	COEF					
Type	I	I	I					
Default	none	0	0					
Remark	1							

VARIABLE	DESCRIPTION
LCID	Linear constraint definition ID. This ID can be used to identify a set to which this constraint is a member.
NID	Node ID

VARIABLE	DESCRIPTION
DOF	Degree of freedom in the global coordinate system; EQ.1: displacement along global x-direction EQ.2: displacement along global y-direction EQ.3: displacement along global z-direction EQ.4: global rotation about global x-axis EQ.5: global rotation about global y-axis EQ.6: global rotation about global z-axis
COEF	Nonzero coefficient, C_k

Remarks:

Nodes of a nodal constraint equation cannot be members of another constraint equation or constraint set that constrain the same degrees-of-freedom, a tied interface, or a rigid body; i.e. nodes cannot be subjected to multiple, independent, and possibly conflicting constraints. Also care must be taken to ensure that single point constraints applied to nodes in a constraint equation do not conflict with the constraint sets constrained degrees-of-freedom.

In this section linear constraint equations of the form:

$$\sum_{k=1}^{n} C_k u_k = C_0$$

can be defined, where u_k are the displacements and C_k are user defined coefficients. Unless LS-DYNA is initialized by linking to an implicit code to satisfy this equation at the beginning of the calculation, the constant C_0 is assumed to be zero. The first constrained degree-of-freedom is eliminated from the equations-of-motion:

$$u_1 = C_0 - \sum_{k=2}^{n} \frac{C_k}{C_1} u_k$$

its velocities and accelerations are given by

$$\dot{u}_1 = -\sum_{k=2}^{n} \frac{C_k}{C_1} \dot{u}_k$$

$$\ddot{u}_1 = -\sum_{k=2}^{n} \frac{C_k}{C_1} \ddot{u}_k,$$

respectively. In the implementation a transformation matrix, $\mathbf{L}$, is constructed relating the unconstrained, $\mathbf{u}$, and constrained, $\mathbf{u}_c$, degrees-of-freedom. The constrained accelerations used in the above equation are given by:

$$\ddot{\mathbf{u}}_c = [\mathbf{L}^\mathrm{T}\mathbf{M}\mathbf{L}]^{-1}\mathbf{L}^\mathrm{T}\mathbf{F}$$

where **M** is the Diagonal lumped mass matrix and **F** is the right hand side force vector. This requires the inversion of the condensed mass matrix which is equal in size to the number of constrained degrees-of-freedom minus one.

Example:

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$  *CONSTRAINED_LINEAR_GLOBAL
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Constrain nodes 40 and 42 to move identically in the z-direction.
$
$  When the linear constraint equation is applied, it goes like this:
$
$     0 = C40uz40 + C42uz42
$
$       = uz40 - uz42
$
$     uz40 = uz42
$
$   where,
$     C40  =  1.00  coefficient for node 40
$     C42  = -1.00  coefficient for node 42
$     uz40 = displacement of node 40 in z-direction
$     uz42 = displacement of node 42 in z-direction
$
$
*CONSTRAINED_LINEAR
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$        i
$       id
         2
$
$      nid       dof      coef
        40         3     1.00
        42         3    -1.00
$
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
```

*CONSTRAINED_LINEAR_LOCAL

Purpose: Define linear constraint equations between displacements and rotations, which can be defined in a local coordinate system. Each node may have a unique coordinate ID.

Card 1	1	2	3	4	5	6	7	8
Variable	LCID							
Type	I							
Default	none							

DOF Cards. Define one card for each constrained degree-of-freedom. Input is terminated at next "*" card.

Card 2	1	2	3	4	5	6	7	8
Variable	NID	DOF	CID	COEF				
Type	I	I	I	I				
Default	none	0	0	0				
Remark	1							

VARIABLE	DESCRIPTION
LCID	LCID for linear constraint definition. This ID can be used to identify a set to which this constraint is a member.
NID	Node ID

VARIABLE	DESCRIPTION
DOF	Degree of freedom in the local coordinate system; EQ.1: displacement along local x-direction EQ.2: displacement along local y-direction EQ.3: displacement along local z-direction EQ.4: local rotation about local x-axis EQ.5: local rotation about local y-axis EQ.6: local rotation about local z-axis
CID	Local coordinate system ID number. If the number is zero, the global coordinate system is used.
COEF	Nonzero coefficient, C_k

Remarks:

In this section linear constraint equations of the form:

$$\sum_{k=1}^{n} C_k\, u_k^L = C_0$$

can be defined, where u_k^L are the displacements in the local coordinate systems and C_k are user defined coefficients. Unless LS-DYNA is initialized by linking to an implicit code to satisfy this equation at the beginning of the calculation, the constant C_0 is assumed to be zero. The first constrained degree-of-freedom is eliminated from the equations-of-motion:

$$u_1^L = C_0 - \sum_{k=2}^{n} \frac{C_k}{C_1} u_k^L$$

Its velocities and accelerations are given by

$$\dot{u}_1^L = -\sum_{k=2}^{n} \frac{C_k}{C_1} \dot{u}_k^L$$

$$\ddot{u}_1^L = -\sum_{k=2}^{n} \frac{C_k}{C_1} \ddot{u}_k^L$$

respectively. The local displacements are calculated every time step using the local coordinate systems defined by the user. More than one degree of freedom for a node can be constrained by specifying a card for each degree of freedom.

WARNING: Nodes of a nodal constraint equation cannot be members of another constraint equation or constraint set that contains the same degrees-of-freedom, tied interface, or rigid bodies.

Nodes must not be subject to multiple, independent, and possibly conflicting constraints. Furthermore, care must be taken to ensure that single point constraints applied to nodes in a constraint equation do not conflict with the constraint set's constrained degrees-of-freedom.

*CONSTRAINED_LOCAL

Purpose: Define a local boundary constraint plane.

Card 1	1	2	3	4	5	6	7	8
Variable	TC	RC	DIR	X	Y	Z	CID	
Type	1	1	1	F	F	F	1	
Default	0	0	0	0	0	0	none	

VARIABLE	DESCRIPTION

TC Translational Constraint in local system:

EQ.1: constrained x translation,

EQ.2: constrained y translation,

EQ.3: constrained z translation,

EQ.4: constrained x and y translations,

EQ.5: constrained y and z translations,

EQ.6: constrained x and z translations,

EQ.7: constrained x, y, and translations.

RC Rotational Constraint in local system:

EQ.1: constrained x-rotation,

EQ.2: constrained y-rotation,

EQ.3: constrained z-rotation,

EQ.4: constrained x and y rotations,

EQ.5: constrained y and z rotations,

EQ.6: constrained z and x rotations,

EQ.7: constrained x, y, and z rotations.

VARIABLE	DESCRIPTION
DIR	Direction of normal for local constraint plane. EQ.1: local x, EQ.2: local y, EQ.3: local z.
X	Local x-coordinate of a point on the local constraint plane.
Y	Local y-coordinate of a point on the local constraint plane.
Z	Local z-coordinate of a point on the local constraint plane.
CID	Coordinate system ID for orientation of the local coordinate system.

Remarks:

Nodes within a mesh-size-dependent tolerance are constrained on a local plane. This option is recommended for use with r-method adaptive remeshing where nodal constraints are lost during the remeshing phase.

*CONSTRAINED_MULTIPLE_GLOBAL

Purpose: Define global multi-point constraints for imposing periodic boundary condition in displacement field.

Card 1	1	2	3	4	5	6	7	8
Variable	ID							
Type	I							
Default								

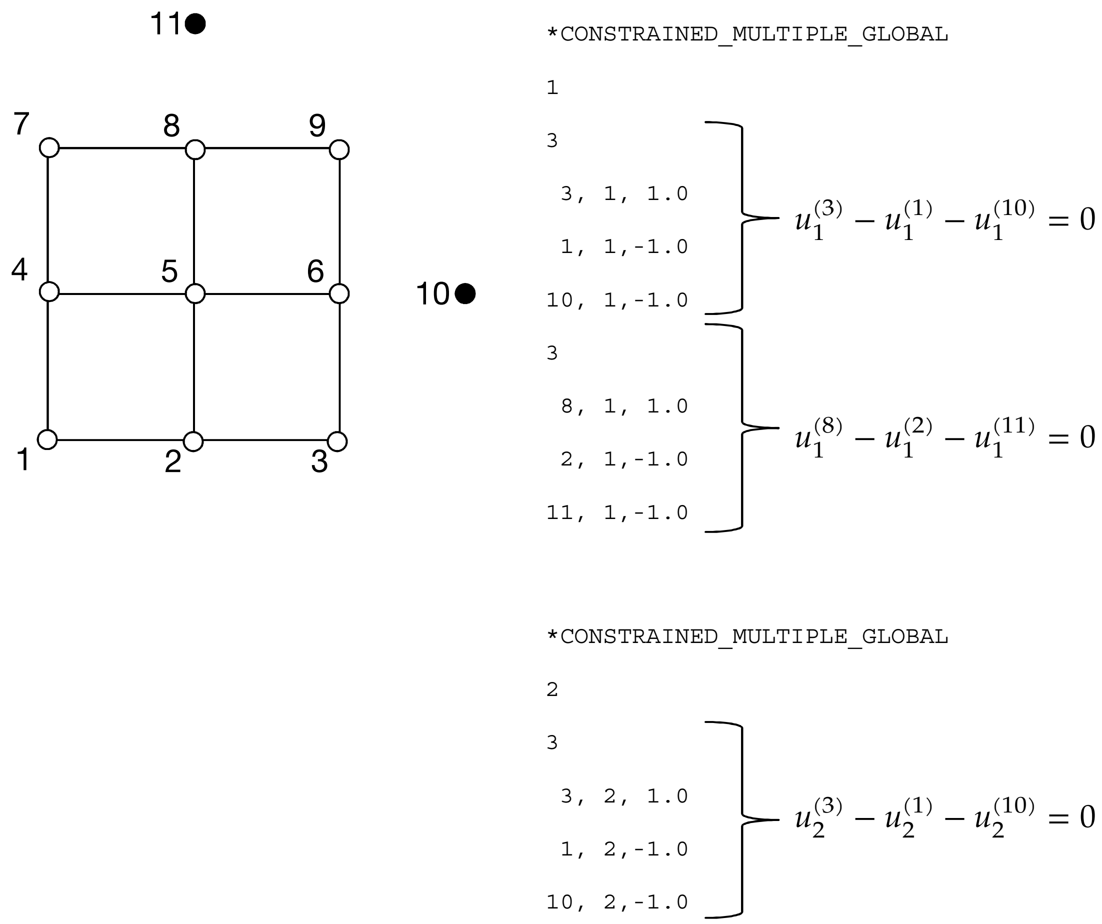
NOTE: For each constraint equation include a set of cards consisting of (1) a C*onstraint Equation Definition Card* and (2) NMP *Coefficient Cards*.

Constraint Equation Definition Card.

Card 2	1	2	3	4	5	6	7	8
Variable	NMP							
Type	I							
Default								

Coefficient Cards. The next NMP cards adhere to this format. Each card sets a single coefficient in the constraint equation.

Card 3	1	2	3	4	5	6	7	8
Variable	NID	DIR	COEF					
Type	I	I	F					
Default								



```
*CONSTRAINED_MULTIPLE_GLOBAL
1
3
 3, 1, 1.0
 1, 1,-1.0
10, 1,-1.0
3
 8, 1, 1.0
 2, 1,-1.0
11, 1,-1.0
```

$$u_1^{(3)} - u_1^{(1)} - u_1^{(10)} = 0$$

$$u_1^{(8)} - u_1^{(2)} - u_1^{(11)} = 0$$

```
*CONSTRAINED_MULTIPLE_GLOBAL
2
3
 3, 2, 1.0
 1, 2,-1.0
10, 2,-1.0
```

$$u_2^{(3)} - u_2^{(1)} - u_2^{(10)} = 0$$

Figure 10-34. Simple example.

VARIABLE	DESCRIPTION
ID	Constraint set identification. All constraint sets should have a unique set ID.
NMP	Number of nodes to be constrained mutually.
NID	Nodal ID
DIR	Direction in three-dimensional space to be constrained EQ.1: x direction EQ.2: y direction EQ.3: z direction LT.0: Extra DOFs for user defined element formulation (e.g. -1: the 1st extra DOF; -2: the 2nd extra DOF; …)

VARIABLE	DESCRIPTION
COEF	Coefficient α_{NID} in constraint equation: $$\sum_{\mathrm{NID}} \alpha_{\mathrm{NID}} u_{\mathrm{DIR}}^{(\mathrm{NID})} = 0.$$

Remarks:

1. Defining multi-point constraints by this keyword can be demonstrated by the following example: a two-dimensional unit square with four quadrilateral elements and 11 nodes as shown in the figure below, where the nodes #10 and #11 are two dummy nodes serving as control points.

CONSTRAINED_NODAL_RIGID_BODY_*{OPTION}_{OPTION}_{OPTION}

Available options include:

<BLANK>

SPC

INERTIA

TITLE

If the center of mass is constrained use the **SPC** option. If the inertial properties are defined rather than computed use the **INERTIA** option. A description for the nodal rigid body can be defined with the **TITLE** option.

Purpose: Define a nodal rigid body. This is a rigid body which consists of the defined nodes. If the INERTIA option is not used, then the inertia tensor is computed from the nodal masses. Arbitrary motion of this rigid body is allowed. If the INERTIA option is used, constant translational and rotational velocities can be defined in a global or local coordinate system.

The first node in the nodal rigid body definition is treated as the master for the case where DRFLAG and RRFLAG are nonzero. The first node always has six degrees-of-freedom. The release conditions applied in the global system are sometimes convenient in small displacement linear analysis, but, otherwise, are not recommended. It is strongly recommended, especially for implicit calculations, that release conditions are only used for a two noded nodal rigid body.

Card Format:

Card 1:	required
Card 2:	required for SPC option
Card 3 - 5:	required for INERTIA option
Card 6:	required if a local coordinate system is used to specify the inertia tensor when the INERTIA option is set

Remarks:

1. Unlike the *CONSTRAINED_NODE_SET which permits only constraints on translational motion, here the equations of rigid body dynamics are used to update the motion of the nodes and therefore rotations of the nodal sets are admissible. Mass properties are determined from the nodal masses and coor-

dinates. Inertial properties are defined if and only if the INERTIA option is specified.

Title Card. Additional card for the TITLE keyword option.

Optional	1	2	3	4	5	6	7	8
Variable	TITLE							
Type	A80							

Card 1	1	2	3	4	5	6	7	8
Variable	PID	CID	NSID	PNODE	IPRT	DRFLAG	RRFLAG	
Type	I	I	I	I	I	I	I	
Default	none	none	none	0	0	0	0	

Center of Mass Constraint Card. Additional card for the SPC keyword option.

Card 2	1	2	3	4	5	6	7	8
Variable	CMO	CON1	CON2					
Type	F	F	F					
Default	0	0	0					

VARIABLE	DESCRIPTION
PID	Part ID of the nodal rigid body.
CID	Optional coordinate system ID for the rigid body local system, see *DEFINE_COORDINATE_OPTION. Output of the rigid body data and the degree-of- freedom releases are done in this local system. This local system rotates with the rigid body.

VARIABLE	DESCRIPTION
NSID	Nodal set ID, see *SET_NODE_OPTION. This nodal set defines the rigid body. If NSID = 0, then NSID = PID, i.e., the node set ID and the part ID are assumed to be identical.
PNODE	An optional node (a massless node is allowed) used for post processing rigid body data. If the PNODE is not located at the rigid body's center of mass, then the initial coordinates of PNODE will be reset to the center of mass. If CID is defined, the velocities and accelerations of PNODE will be output in the local system to the d3plot and d3thdt files unless PNODE is specified as a negative number, in which case the global system is used.
IPRT	Print flag. For nodal rigid bodies the following values apply: EQ.1: write data into rbdout EQ.2: do not write data into rbdout Printing is suppressed for two noded rigid bodies unless IPRT is set to unity. This is to avoid excessively large rbdout files when many, two-noded welds are used.
DRFLAG	Displacement release flag for all nodes except the first node in the definition. EQ.-7: release x, y, and z displacement in global system EQ.-6: release z and x displacement in global system EQ.-5: release y and z displacement in global system EQ.-4: release x and y displacement in global system EQ.-3: release z displacement in global system EQ.-2: release y displacement in global system EQ.-1: release x displacement in global system EQ.0: off for rigid body behavior EQ.1: release x displacement in rigid body local system EQ.2: release y displacement in rigid body local system EQ.3: release z displacement in rigid body local system EQ.4: release x and y displacement in rigid body local system EQ.5: release y and z displacement in rigid body local system EQ.6: release z and x displacement in rigid body local system EQ.7: release x, y, and z displacement in rigid body local

VARIABLE **DESCRIPTION**

system

RRFLAG

Rotation release flag for all nodes except the first node in the definition.

EQ.-7: release x, y, and z rotations in global system

EQ.-6: release z and x rotations in global system

EQ.-5: release y and z rotations in global system

EQ.-4: release x and y rotations in global system

EQ.-3: release z rotation in global system

EQ.-2: release y rotation in global system

EQ.-1: release x rotation in global system

EQ.0: off for rigid body behavior

EQ.1: release x rotation in rigid body local system

EQ.2: release y rotation in rigid body local system

EQ.3: release z rotation in rigid body local system

EQ.4: release x and y rotations in rigid body local system

EQ.5: release y and z rotations in rigid body local system

EQ.6: release z and x rotations in rigid body local system

EQ.7: release x, y, and z rotations in rigid body local system

CMO

Center of mass constraint option, CMO:

EQ.+1.0: constraints applied in global directions,

EQ.0.0: no constraints,

EQ.-1.0: constraints applied in local directions (SPC constraint).

CON1

First constraint parameter:

<u>If CMO=+1.0, then specify</u> global translational constraint:

EQ.0: no constraints,

EQ.1: constrained x displacement,

EQ.2: constrained y displacement,

EQ.3: constrained z displacement,

EQ.4: constrained x and y displacements,

VARIABLE **DESCRIPTION**

EQ.5: constrained y and z displacements,

EQ.6: constrained z and x displacements,

EQ.7: constrained x, y, and z displacements.

If CM0 = -1.0, then specify local coordinate system ID. See *DEFINE_COORDINATE_*OPTION*: This coordinate system is fixed in time

CON2 Second constraint parameter:

If CMO=+1.0, then specify global rotational constraint:

EQ.0: no constraints,

EQ.1: constrained x rotation,

EQ.2: constrained y rotation,

EQ.3: constrained z rotation,

EQ.4: constrained x and y rotations,

EQ.5: constrained y and z rotations,

EQ.6: constrained z and x rotations,

EQ.7: constrained x, y, and z rotations.

If CM0 = -1.0, then specify local (SPC) constraint:

EQ.000000: no constraint,

EQ.100000: constrained x translation,

EQ.010000: constrained y translation,

EQ.001000: constrained z translation,

EQ.000100: constrained x rotation,

EQ.000010 : constrained y rotation,

EQ.000001: constrained z rotation.

Any combination of local constraints can be achieved by adding the number 1 into the corresponding column.

Inertia Card 1. Additional card for the INERTIA keyword option.

Card 3	1	2	3	4	5	6	7	8
Variable	XC	YC	ZC	TM	IRCS	NODEID		
Type	F	F	F	F	I	I		
Default	0	0	0	0	0	0		

VARIABLE	DESCRIPTION
XC	*x*-coordinate of center of mass. If nodal point, NODEID, is defined, XC, YC, and ZC are ignored and the coordinates of the nodal point, NODEID, are taken as the center of mass.
YC	*y*-coordinate of center of mass
ZC	*z*-coordinate of center of mass
TM	Translational mass
IRCS	Flag for inertia tensor reference coordinate system: EQ.0: global inertia tensor, EQ.1: local inertia tensor is given in a system defined by the orientation vectors as given below.
NODEID	Optional nodal point defining the CG of the rigid body. If this node is not a member of the set NSID above, its motion will not be updated to correspond with the nodal rigid body after the calculation begins. PNODE and NODEID can be identical if and only if PNODE physically lies at the mass center at time zero.

Inertia Card 2. Second Additional card for the INERTIA keyword option.

Card 4	1	2	3	4	5	6	7	8
Variable	IXX	IXY	IXZ	IYY	IYZ	IZZ		
Type	F	F	F	F	F	F		
Default	none	0	0	none	0	0		

VARIABLE	DESCRIPTION
IXX	I_{xx}, xx component of inertia tensor
IXY	I_{xy}, xy component of inertia tensor
IXZ	I_{xz}, xz component of inertia tensor
IYY	I_{yy}, yy component of inertia tensor
IYZ	I_{yz}, yz component of inertia tensor
IZZ	I_{zz}, zz component of inertia tensor

Inertia Card 3. Third additional card for the INERTIA keyword option.

Card 5	1	2	3	4	5	6	7	8
Variable	VTX	VTY	VTZ	VRX	VRY	VRZ		
Type	F	F	F	F	F	F		
Default	0	0	0	0	0	0		

VARIABLE	DESCRIPTION
VTX	x-rigid body initial translational velocity in global coordinate system.
VTY	y-rigid body initial translational velocity in global coordinate system.

VARIABLE **DESCRIPTION**

VTZ z-rigid body initial translational velocity in global coordinate system.

VRX x-rigid body initial rotational velocity in global coordinate system.

VRY y-rigid body initial rotational velocity in global coordinate system.

VRZ z-rigid body initial rotational velocity in global coordinate system.

Remarks:

The velocities defined above can be overwritten by the *INITIAL_VELOCITY card.

Local Inertia Tensor Card. Additional card required for IRCS = 1 (see Inertia Card 1). Define two local vectors or a local coordinate system ID.

Card 6	1	2	3	4	5	6	7	8
Variable	XL	YL	ZL	XLIP	YLIP	ZLIP	CID2	
Type	F	F	F	F	F	F	I	
Default	none	none	none	none	none	none	none	

VARIABLE **DESCRIPTION**

XL x-coordinate of local x-axis. Origin lies at (0,0,0).

YL y-coordinate of local x-axis

ZL z-coordinate of local x-axis

XLIP x-coordinate of local in-plane vector

YLIP y-coordinate of local in-plane vector

ZLIP z-coordinate of local in-plane vector

CID2 Local coordinate system ID, see *DEFINE_COORDINATE_.... With this option leave fields 1-6 blank.

Remarks:

The local coordinate system is set up in the following way. After the local x-axis is defined, the local *z*-axis is computed from the cross-product of the local *x*-axis vector with the given in-plane vector. Finally, the local *y*-axis is determined from the cross-product of the local *z*-axis with the local *x*-axis. The local coordinate system defined by CID has the advantage that the local system can be defined by nodes in the rigid body which makes repositioning of the rigid body in a preprocessor much easier since the local system moves with the nodal points.

Example:

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$  *CONSTRAINED_NODAL_RIGID_BODY
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Define a rigid body consisting of the nodes in nodal set 61.
$
$  This particular example was used to connect three separate deformable
$  parts.  Physically, these parts were welded together.  Modeling wise,
$  however, this joint is quit messy and is most conveniently modeled
$  by making a rigid body using several of the nodes in the area.  Physically,
$  this joint was so strong that weld failure was never of concern.
$
*CONSTRAINED_NODAL_RIGID_BODY
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      pid       cid      nsid
        45                  61
$
$     nsid = 61   nodal set ID number, requires a *SET_NODE_option
$      cid        not used in this example, output will be in global coordinates
$
$
*SET_NODE_LIST
$      sid
        61
$     nid1      nid2      nid3      nid4      nid5      nid6      nid7      nid8
       823      1057      1174      1931      2124      1961      2101
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

*CONSTRAINED_NODE_INTERPOLATION

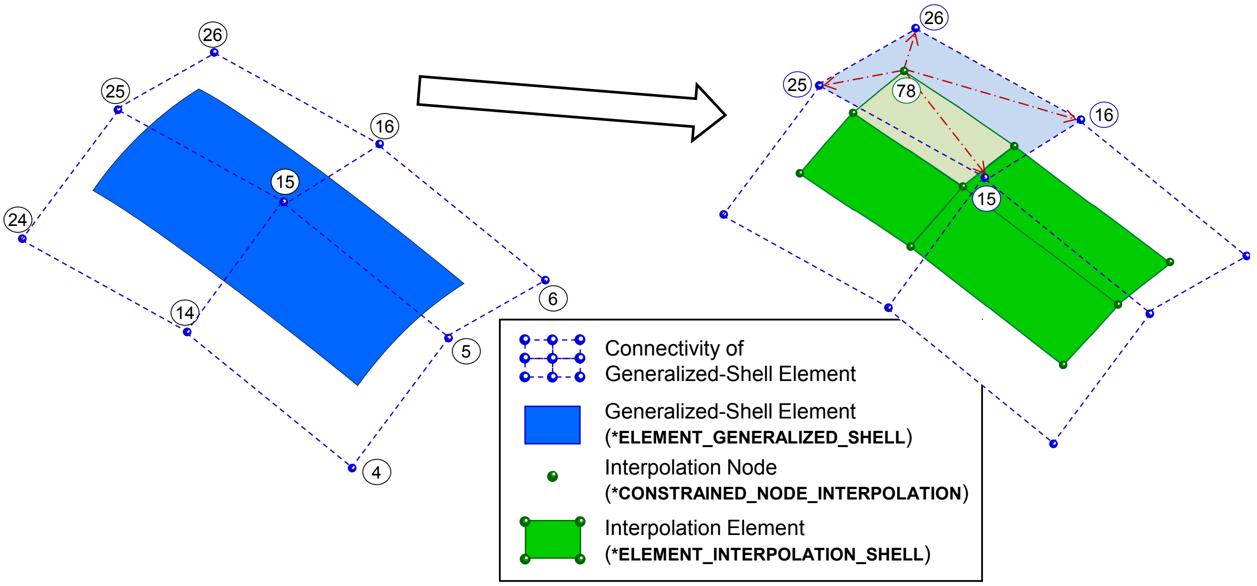
Purpose: Define constrained nodes for the use of *ELEMENT_INTERPOLATION_-SHELL and *ELEMENT_INTERPOLATION_SOLID to model contact and to visualize the results of generalized elements (see *ELEMENT_GENERALIZED_SHELL/SOLID). The displacements of these nodes are dependent of their corresponding master nodes.

Card 1	1	2	3	4	5	6	7	8
Variable	NID	NUMMN						
Type	I	I						
Default	none	none						

Weighting Factor Cards. For each of the NUMMN master nodes (see *ELEMENT_-GENERALIZED_SHELL/SOLID) NID depends on set a MN and W entry. Each Weighting Factor Card can accommodate four master nodes. Add as many Weighting Factor Cards as needed.

Card 2	1	2	3	4	5	6	7	8
Variable	MN1	W1	MN2	W2	MN3	W3	MN4	W4
Type	I	F	I	F	I	F	I	F
Default	none	none	none	none	none	none	none	none

Card 3	1	2	3	4	5	6	7	8
Variable	MN5	W5	Etc.	Etc.	Etc.	Etc.	Etc.	Etc.
Type	I	F	I	F	I	F	I	F
Default	none	none	none	none	none	none	none	none


26
25
16
15
24
14
6
5
4
26
25
78
16
15
Connectivity of Generalized-Shell Element
Generalized-Shell Element (*ELEMENT_GENERALIZED_SHELL)
Interpolation Node (*CONSTRAINED_NODE_INTERPOLATION)
Interpolation Element (*ELEMENT_INTERPOLATION_SHELL)

```
*CONSTRAINED_NODE_INTERPOLATION
$---+--NID----+NUMCN----+----3----+----4----+----5----+----6----+----7----+----8
        78         4
$---+--CN1----+---W1----+--CN2----+---W2----+--CN3----+---W3----+--CN4----+---W4
        26      0.35        25      0.32        15      0.18        16      0.15
```

Figure 10-35. Example of a *CONSTRAINED_NODE_INTERPOLATION card

VARIABLE	DESCRIPTION
NID	Node ID of the interpolation node as defined in *NODE (see Remark 1).
NUMMN	Number of master nodes, this constrained node depends on.
MN*i*	Node ID of master node *i*.
W*i*	Weighting factor of master node *i*.

Remarks:

1. The coordinates of an interpolation node have to be defined in *NODE. In there the translational and rotational constraints TC = 7. and RC = 7. need to be set.

2. The displacements of the interpolation node, $\boldsymbol{d}_{\mathrm{IN}}$, are interpolated based on the displacements of the corresponding master nodes, $\boldsymbol{d}_i$, and the appropriate weighting factors w_i. The interpolation is computed as follows:

$$\boldsymbol{d}_{\mathrm{IN}} = \sum_{i=1}^{\mathrm{NUMMN}} w_i \boldsymbol{d}_i .$$

CONSTRAINED_NODE_SET_*{OPTION}

To define an ID for the constrained node set the following option is available:

<BLANK>

ID

If the ID is defined an additional card is required.

Purpose: Define nodal constraint sets for translational motion in global coordinates. No rotational coupling. See Figure 10-36. Nodal points included in the sets should not be subjected to any other constraints including prescribed motion, e.g., with the *BOUNDARY_PRESCRIBED_MOTION options.

ID Card. Additional card for ID keyword option.

Card 1	1	2	3	4	5	6	7	8
Variable	CNSID							
Type	I							
Default	0							

Card 2	1	2	3	4	5	6	7	8
Variable	NSID	DOF	TF					
Type	I	I	F					
Default	none	none	1.E+20					
Remarks	1		2					

VARIABLE	DESCRIPTION
CNSID	Optional constrained node set ID.
NSID	Nodal set ID, see *SET_NODE_*OPTION*.

*CONTRAINED_NODE_SET

Since no rotation is permitted, this option should *not* be used to model rigid body behavior involving rotations

*CONSTRAINED_NODAL_RIGID_BODY *CONSTRAINED_SPOTWELD

Behavior is like a rigid beam. These options *may* be used to model spotwelds.

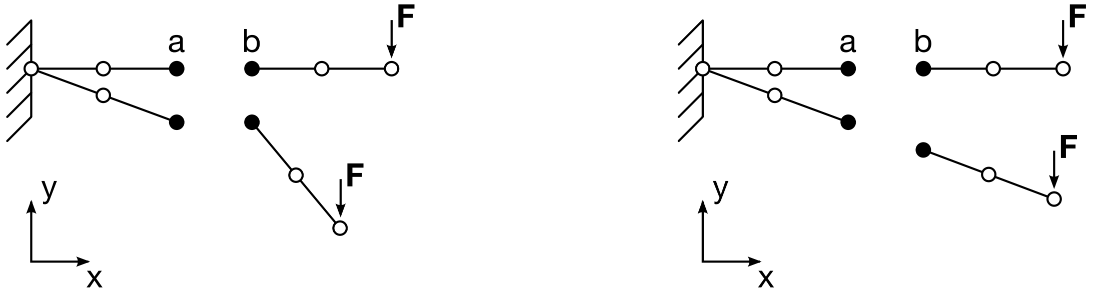


Figure 10-36. Two different ways to constrain node *a* and *b*. For rigid-body type situations this card, *CONSTRAINED_NODE_SET may lead to unphysical results.

VARIABLE	DESCRIPTION
DOF	Applicable degrees-of-freedom: EQ.1: x-translational degree-of-freedom, EQ.2: y-translational degree-of-freedom, EQ.3: z-translational degree-of-freedom, EQ.4: x and y-translational degrees-of-freedom, EQ.5: y and z-translational degrees-of-freedom, EQ.6: z and x-translational degrees-of-freedom, EQ.7: x, y, and z-translational degrees-of-freedom.
TF	Failure time for nodal constraint set.

Remarks:

1. The masses of the nodes are summed up to determine the total mass of the constrained set. It must be noted that the definition of a nodal rigid body is not possible with this input For nodal rigid bodies the keyword input: *CONSTRAINED_NODAL_RIGID_BODY_*OPTION*, must be used.

2. When the failure time, TF, is reached the nodal constraint becomes inactive and the constrained nodes may move freely.

Example:

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$  *CONSTRAINED_NODE_SET
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Constrain all the nodes in a nodal set to move equivalently
$  in the z-direction.
$
*CONSTRAINED_NODE_SET
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
      nsid       dof        tf
         7         3      10.0
$
$     nsid = 7    nodal set ID number, requires a *SET_NODE_option
$      dof = 3    nodal motions are equivalent in z-translation
$      tf  = 3    at time=10.  the nodal constraint is removed
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
```

CONSTRAINED_NODE_TO_NURBS_PATCH_*{OPTION}

Purpose: To add additional massless nodes to the surface of a NURBS patch. The motion of the nodes is governed by the NURBS patch. Forces applied to the nodes are distributed to the NURBS patch. Penalty method is used to handle the displacement boundary conditions CON (see below) on the specified nodes.

To specify node sets instead of individual nodes use the option:

SET

Card 1	1	2	3	4	5	6	7	8
Variable	PATCHID	NSID	CON	CID	SF			
Type	I	I	I	I	F			
Default	none	none	0		1.0			

VARIABLE	DESCRIPTION
PATCHID	Patch ID.
NSID	Nodal set ID or node ID depending on the *OPTION*.
CON	Constraint parameter for extra node(s) of NSID. Its definition is same as that of CON2 when CM0 = -1 as described in MAT_RIGID. For example "1110" means constrained *z*-translation, *x*-rotation and *y*-rotation.
CID	Coordinate system ID for constraint
SF	Penalty force scale factor for the penalty-based constraint

*CONSTRAINED_POINTS

Purpose: Constrain two points with the specified coordinates connecting two shell elements at locations other than nodal points. In this option, the penalty method is used to constrain the translational and rotational degrees-of-freedom of the points. Force resultants are written into the swforc ASCII file for post-processing.

Card 1	1	2	3	4	5	6	7	8
Variable	CID							
Type	I							
Default	none							

Card 2	1	2	3	4	5	6	7	8	9	10
Variable	EID1	X1		Y1		Z1				
Type	I	F		F		F				
Default	none	0.		0.		0.				

Card 3	1	2	3	4	5	6	7	8	9	10
Variable	EID2	X2		Y2		Z2				
Type	I	F		F		F				
Default	none	0.		0.		0.				

Card 4	1	2	3	4	5	6	7	8
Variable	PSF	FAILA	FAILS	FAILM				
Type	F	F	F	F				
Default	1.0	0.0	0.0	0.0				

VARIABLE	DESCRIPTION
CID	Constrained points ID.
EID*i*	Shell element ID, *i* = 1, 2.
X*i*, Y*i*, Z*i*	Coordinates of the constrained points, *i* = 1, 2.
PSF	Penalty scale factor (Default = 1.0).
FAILA	Axial force resultant failure value, no failure if zero.
FAILS	Shear force resultant failure value, no failure if zero.
FAILM	Moment resultant failure value, no failure if zero.

*CONSTRAINED_RIGID_BODIES

Purpose: Merge two rigid bodies. One rigid body, called slave rigid body, is merged to the other one called a master rigid body. This command applies to parts comprised of *MAT_RIGID but not to nodal rigid bodies (*CONSTRAINED_NODAL_RIGID_BODY).

Card 1	1	2	3	4	5	6	7	8
Variable	PIDM	PIDS	IFLAG					
Type	I	I	I					
Default	none	none	0					

VARIABLE	DESCRIPTION
PIDM	Master rigid body part ID, see *PART.
PIDS	Slave rigid body part ID, see *PART.
IFLAG	This flag is meaningful *if and only if* the inertia properties of the Part, PIDM, are defined in PART_INERTIA. EQ.1: Update the center-of-gravity, the translational mass, and the inertia matrix of PIDM to reflect its merging with the slave rigid body (PIDS). EQ.0: The merged PIDS will not affect the properties defined in PART_INERTIA for PIDM since it is assumed the properties already account for merged parts. The inertia properties of PIDS will be computed from its nodal masses if the properties are not defined in a PART_INERTIA definition.

Remarks:

1. The slave rigid body is merged to the master rigid body. The inertial properties computed by LS-DYNA are based on the combination of the master rigid body plus all the rigid bodies which are slaved to it unless the inertial properties of the master rigid body are defined via the *PART_INERTIA keyword in which case those properties are used for the combination of the master and slave rigid bodies. Note that a master rigid body may have many slaves.

2. Independent rigid bodies must not share common nodes since each rigid body updates the motion of its nodes independently of the other rigid bodies. If common nodes exist between rigid bodies the rigid bodies sharing the nodes must be merged.

3. It is also possible to merge rigid bodies that are completely separated and share no common nodal points or boundaries. All actions valid for the master rigid body, e.g., constraints, given velocity, are now also valid for the newly-created rigid body.

Example:

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$  *CONSTRAINED_RIGID_BODIES
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Rigidly connect parts 35, 70, 71, and 72 to part 12.
$  All parts must be defined as rigid.
$
$  This example is used to make a single rigid body out of the five parts
$  that compose the back end of a vehicle.  This was done to save cpu time
$  and was determined to be valid because the application was a frontal
$  impact with insignificant rear end deformations.  (The cpu time saved
$  was from making the parts rigid, not from merging them - merging was
$  more of a convenience in this case for post processing, for checking
$  inertial properties, and for joining the parts.)
$
*CONSTRAINED_RIGID_BODIES
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$     pidm      pids
        12        35
        12        70
        12        71
        12        72
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

*CONSTRAINED_RIGID_BODY_INSERT

Purpose: This keyword is for modeling die inserts. One rigid body, called slave rigid body, is constrained to move with another rigid body, called the master rigid body, in all directions except for one.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	PIDM	PIDS	COORDID	IDIR			
Type	I	I	I	I	I			
Default	none	none	none	none	3			

Card 2	1	2	3	4	5	6	7	8
Variable	MFLAG	MCID	DEATHM					
Type	I	I	F					
Default	0	0	0.0					

Card 3	1	2	3	4	5	6	7	8
Variable	PARTB	DEATHB						
Type	I	F						
Default	0	0.0						

VARIABLE	DESCRIPTION
ID	Insert ID
PIDM	Master (die) rigid body part ID, see *PART.
PIDS	Slave (die insert) rigid body part ID, see *PART.

VARIABLE DESCRIPTION

COORDID

Coordinate ID. The x direction is the direction the insert moves independently of the die.

IDIR

The direction the insert moves independently of the die. If unspecified, it defaults to the local z direction, IDIR = 3.

MFLAG

Motion flag.

EQ.0: Relative motion is unconstrained.

EQ.1: The displacement of the insert relative to the die is imposed.

EQ.2: The velocity of the insert relative to the die is imposed.

EQ.3: The acceleration of the insert relative to the die is imposed.

MCID

Curve defining the motion of the die insert relative to the die.

DEATHM

Death time of the imposed motion. If it is equal to 0.0, the motion is imposed for the entire analysis.

PARTB

Part ID for a discrete beam connected between the insert and die.

DEATHB

Death time for the discrete beam specified by BPART.

Remarks:

1. This capability is supported by both the implicit and explicit time integrators; however, the joint death time DEATHM feature works only for explicit integration with the penalty method.

2. The translational joint constraining the die and the die insert are automatically generated. The joint reaction forces will appear in the jntforc output file.

3. The translational motor constraining the remaining translational degree of freedom is also automatically generated, and its reaction forces also appear in the jntforc output file.

4. The automatically generated beam has its data written to the d3plot file, and all of the optional appropriate output files.

*CONSTRAINED_RIGID_BODY_STOPPERS

Purpose: Rigid body stoppers provide a convenient way of controlling the motion of rigid tooling in metalforming applications. The motion of a "master" rigid body is limited by load curves. This option will stop the motion based on a time dependent constraint. The stopper overrides prescribed motion boundary conditions (except relative displacement) operating in the same direction for both the master and slaved rigid bodies. See Figure 10-37.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	LCMAX	LCMIN	PSIDMX	PSIDMN	LCVMNX	DIR	VID
Type	I	I	I	I	I	I	I	I
Default	required	0	0	0	0	0	required	0

Card 2	1	2	3	4	5	6	7	8
Variable	TB	TD						
Type	F	F						
Default	0	1021						

VARIABLE	DESCRIPTION
PID	Part ID of master rigid body, see *PART.
LCMAX	Load curve ID defining the maximum coordinate or displacement as a function of time. See *DEFINE_CURVE: LT.0: Load Curve ID |LCMAX| provides an upper bound for the displacement of the rigid body EQ.0: no limitation of the maximum displacement. GT.0: Load Curve ID LCMAX provides an upper bound for the position of the rigid body center of mass

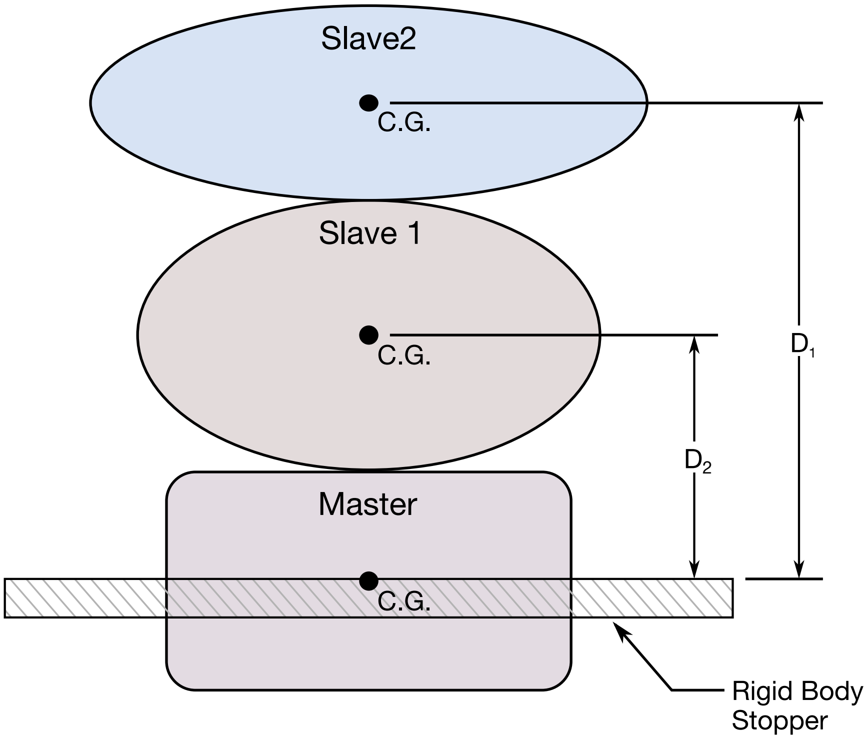


Figure 10-37. When the master rigid body reaches the rigid body stopper, the velocity component into the stopper is set to zero. Slave rigid bodies 1 and 2 also stop if the distance between their mass centers and the master rigid body is less than or equal to the input values D1 and D2, respectively.

VARIABLE	DESCRIPTION
LCMIN	Load curve ID defining the minimum coordinate or displacement as a function of time. See *DEFINE_CURVE: LT.0: Load Curve ID |LCMIN| defines a lower bound for the displacement of the rigid body EQ.0: no limitation of the minimum displacement. GT.0: Load Curve ID LCMIN defines a lower bound for the position of the rigid body center of mass
PSIDMX	Optional part set ID of rigid bodies that are slaved in the maximum coordinate direction to the master rigid body. The part set definition, (see *SET_PART_COLUMN) may be used to define the closure distance (D_1 and D_2 in Figure 10-37) which activates the constraint. The constraint does not begin to act until the master rigid body stops. If the distance between the master rigid body is greater than or equal to the closure distance, the slave rigid body motion away from the master rigid body also stops. However, the slaved rigid body is free to move towards the

VARIABLE DESCRIPTION

master. If the closure distance is input as zero (0.0) then the slaved rigid body stops when the master stops.

PSIDMN

Optional part set ID of rigid bodies that are slaved in the minimum coordinate direction to the master rigid body. The part set definition, (see *SET_PART_COLUMN) may be used to define the closure distance (D_1 and D_2 in Figure 10-37) which activates the constraint. The constraint does not begin to act until the master rigid body stops. If the distance between the master rigid body is less than or equal to the closure distance, the slave rigid body motion towards the master rigid body also stops. However, the slaved rigid body is free to move away from the master. If the closure distance is input as zero (0.0) then the slaved rigid body stops when the master stops.

LCVMX

Load curve ID which defines the maximum absolute value of the velocity as a function of time that is allowed for the master rigid body. See *DEFINE_CURVE:

EQ.0: no limitation on the velocity.

DIR

Direction stopper acts in:

EQ.1: x-translation,

EQ.2: y-translation,

EQ.3: z-translation,

EQ.4: arbitrary, defined by vector VID (see below),

EQ.5: x-axis rotation,

EQ.6: y-axis rotation,

EQ.7: z-axis rotation,

EQ.8: arbitrary, defined by vector VID (see below).

VID

Vector for arbitrary orientation of stopper, see *DEFINE_VECTOR.

TB

Time at which stopper is activated.

TD

Time at which stopper is deactivated.

Remarks:

The optional definition of part sets in minimum or maximum coordinate direction allows the motion to be controlled in arbitrary direction.

*CONSTRAINED_RIVET_{OPTION}

To define an ID for the rivet, the following option is available:

<BLANK>

ID

If the ID is defined an additional card is required.

Purpose: Define massless rivets between non-contiguous nodal pairs. The nodes must not have the same coordinates. The action is such that the distance between the two nodes is kept constant throughout any motion. No failure can be specified.

ID Card. Additional card for the ID keyword option.

ID	1	2	3	4	5	6	7	8
Variable	RID							
Type	I							
Default	0							

Card 1	1	2	3	4	5	6	7	8
Variable	N1	N2	TF					
Type	I	I	F					
Default	none	none	1.E+20					
Remarks	1		2					

VARIABLE	DESCRIPTION
RID	Optional rivet ID.
N1	Node ID
N2	Node ID

VARIABLE	DESCRIPTION
TF	Failure time for nodal constraint set.

Remarks:

1. Nodes connected by a rivet cannot be members of another constraint set that constrain the same degrees-of-freedom, a tied interface, or a rigid body, i.e., nodes cannot be subjected to multiple, independent, and possibly conflicting constraints. Also care must be taken to ensure that single point constraints applied to nodes in a constraint set do not conflict with the constraint sets constrained degrees-of-freedom.

2. When the failure time, TF, is reached the rivet becomes inactive and the constrained nodes may move freely.

Example:

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$  *CONSTRAINED_RIVET
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Connect node 382 to node 88471 with a massless rivet.
$
*CONSTRAINED_RIVET
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$       n1        n2        tf
       382     88471       0.0
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

*CONSTRAINED_SHELL_TO_SOLID

Purpose: Define a tie between a shell edge and solid elements. Nodal rigid bodies can perform the same function and may also be used.

Card 1	1	2	3	4	5	6	7	8
Variable	NID	NSID						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
NID	Shell node ID
NSID	Solid nodal set ID, see *SET_NODE_*OPTION*.

Remarks:

The shell-brick interface, an extension of the tied surface capability, ties regions of hexahedron elements to regions of shell elements. A shell node may be tied to up to

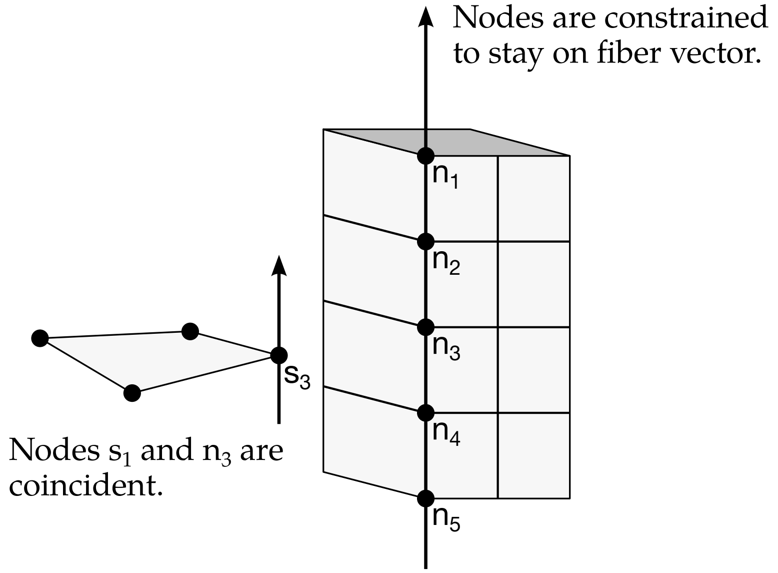


Figure 10-38. The interface between shell elements and solids ties shell node s1 to a line of nodes on the solid elements n1-n5. It is very important for the nodes to be aligned.

nine brick nodes lying along the tangent vector to the nodal fiber. See Figure 10-38. During the calculation, the brick nodes thus constrained, must lie along the fiber but can move relative to each other in the fiber direction. The shell node stays on the fiber at the same relative spacing between the first and last brick node. The brick nodes must be input in the order in which they occur, in either the plus or minus direction, as one moves along the shell node fiber.

This feature is intended to tie four node shells to eight node shells or solids; it is not intended for tying eight node shells to eight node solids.

Example:

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$  *CONSTRAINED_SHELL_TO_SOLID
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Tie shell element, at node 329, to a solid element at node 203.
$    - nodes 329 and 203 are coincident
$
$  Additionally, define a line of nodes on the solids elements, containing
$  node 203, that must remain in the same direction as the fiber of the shell
$  containing node 329.  In other words:
$
$    - Nodes 119, 161, 203, 245 and 287 are nodes on a solid part that
$       define a line on that solid part.
$    - This line of nodes will be constrained to remain linear throughout
$       the simulation.
$    - The direction of this line will be kept the same as the fiber of the
$       of the shell containing node 329.
$
*CONSTRAINED_SHELL_TO_SOLID
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      nid      nsid
       329         4
$
*SET_NODE_LIST
$      sid
         4
$     nid1      nid2      nid3      nid4      nid5      nid6      nid7      nid8
       119       161       203       245       287
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

*CONSTRAINED_SPLINE

Purpose: Define an elastic cubic spline interpolation constraint. The displacements and slopes at the end points are continuous. The first and last nodes, which define the constraint, must be independent. The degrees-of-freedom of interior nodes may be either dependent or independent.

Card 1	1	2	3	4	5	6	7	8
Variable	SPLID	DLRATIO						
Type	I	I						
Default	0	0.10						

Node Cards. Include one card per independent/dependent node. The first and last nodes must be independent. The next "*" card terminates this input.

Card 2	1	2	3	4	5	6	7	8
Variable	NID	DOF						
Type	I	I						
Default	0	0						

VARIABLE	DESCRIPTION
SPLID	Spline constraint ID.
DLRATIO	Ratio of bending to torsional stiffness for an elastic tubular beam which connects the independent degrees-of-freedom. The default value is set to 0.10.
NID	Independent/dependent node ID. For explicit problems this node should not be a member of a rigid body, or elsewhere constrained in the input.

VARIABLE **DESCRIPTION**

DOF

Degrees-of-freedom. The list of dependent degrees-of-freedom consists of a number with up to six digits, with each digit representing a degree of freedom. For example, the value 1356 indicates that degrees of freedom 1, 3, 5, and 6 are controlled by the constraint. The default is 123456. Digit: degree of freedom ID's:

EQ.1: x

EQ.2: y

EQ.3: z

EQ.4: rotation about x axis

EQ.5: rotation about y axis

EQ.6: rotation about z axis

CONSTRAINED_SPOTWELD_*{OPTION}_{OPTION}

If it is desired to use a time filtered force calculation for the forced based failure criterion then the following option is available:

<BLANK>

FILTERED_FORCE

and one additional card must be defined below. To define an ID for the spotweld the following option is available:

<BLANK>

ID

If the ID is defined an additional card is required. The ordering of the options is arbitrary.

Purpose: Define massless spot welds between non-contiguous nodal pairs.

The spot weld is a rigid beam that connects the nodal points of the nodal pairs; thus, nodal rotations and displacements are coupled. The spot welds must be connected to nodes having rotary inertias, i.e., beams or shells. If this is not the case, for example, if the nodes belong to solid elements, use the option: *CONSTRAINED_RIVET. During implicit calculations this case is treated like a rivet, constraining only the displacements. Note that shell elements do not have rotary stiffness in the normal direction and, therefore, this component cannot be transmitted.

Spot welded nodes must not have the same coordinates. Coincident nodes in a spot weld can be handled by the *CONSTRAINED_NODAL_RIGID_BODY option. Brittle and ductile failures are supported by this model. Brittle failure is based on the resultant forces acting on the weld, and ductile failure is based on the average plastic strain value of the shell elements which include the spot welded node. Spot welds, which are connected to massless nodes, are automatically deleted in the initialization phase and a warning message is printed in the messag file and the d3hsp file.

Warning. The accelerations of spot welded nodes are output as zero into the various databases, but if the acceleration of spotwelded nodes are required, use either the *CONSTRAINED_GENERALIZED_WELD or the *CONSTRAINED_NODAL_RIGID_BODY input. However, if the output interval is frequent enough accurate acceleration time histories can be obtained from the velocity time history by differentiation in the post-processing phase.

ID Card. Additional card for the ID keyword option.

ID	1	2	3	4	5	6	7	8
Variable	WID							
Type	I							
Default	0							

Card 1	1	2	3	4	5	6	7	8
Variable	N1	N2	SN	SS	N	M	TF	EP
Type	I	I	F	F	F	F	F	F
Default	none	none	optional	optional	none	none	1.E+20	1.E+20
Remarks	1.		2.				3	4

Filter Card. Additional card for the FILTERED_FORCE keyword option.

Card 2	1	2	3	4	5	6	7	8
Variable	NF	TW						
Type	I	F						
Default	none	none						

VARIABLE	DESCRIPTION
WID	Optional weld ID.
N1	Node ID
N2	Node ID
SN	Normal force at spotweld failure (see Remark 2 below).

VARIABLE	DESCRIPTION
SS	Shear force at spotweld failure (see Remark 2 below).
N	Exponent for normal spotweld force (see Remark 2 below).
M	Exponent for shear spotweld force (see Remark 2 below).
TF	Failure time for nodal constraint set.
EP	Effective plastic strain at failure.
NF	Number of force vectors stored for filtering.
TW	Time window for filtering.

Remarks:

1. Nodes connected by a spot weld cannot be members of another constraint set that constrain the same degrees-of-freedom, a tied interface, or a rigid body, i.e., nodes cannot be subjected to multiple, independent, and possibly conflicting constraints. Also, care must be taken to ensure that single point constraints applied to nodes in a constraint set do not conflict with the constraint sets constrained degrees-of-freedom.

2. Failure of the spot welds occurs when:

$$\left(\frac{|f_n|}{S_n}\right)^n + \left(\frac{|f_s|}{S_s}\right)^m \geq 1$$

where f_n and f_s are the normal and shear interface force. Component f_n is nonzero for tensile values only.

3. When the failure time, TF, is reached the spot weld becomes inactive and the constrained nodes may move freely.

4. Spot weld failure due to plastic straining occurs when the effective nodal plastic strain exceeds the input value,$\varepsilon^p_{\mathrm{fail}}$. This option can model the tearing out of a spotweld from the sheet metal since the plasticity is in the material that surrounds the spotweld, not the spotweld itself. A least squares algorithm is used to generate the nodal values of plastic strains at the nodes from the element integration point values. The plastic strain is integrated through the element and the average value is projected to the nodes via a least square fit. This option should only be used for the material models related to metallic plasticity and can result is slightly increased run times. Failures can include both the plastic and brittle failures.

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$
$
$$$$  *CONSTRAINED_SPOTWELD
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$
$
$  Spotweld two nodes (34574 and 34383) with the approximate strength
$  of a 3/8" SAE Grade No 3 bolt.
$
*CONSTRAINED_SPOTWELD
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>...
.8
$       n1        n2        sn        sf         n         m        tf
ps
     34574     34383      36.0      18.0       2.0       2.0       10.
1.0
$
$
$       sn = 36.0  normal failure force is 36 kN
$       sf = 18.0  shear failure force is 18 kN
$        n = 2.0   normal failure criteria is raised to the power of 2
$        m = 2.0   shear failure criteria is raised to the power of 2
$       tf = 10.0  failure occurs at time 10 unless strain failure occurs
$       ps = 2.0   plastic strain at failure
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$
```

5. Thermal: The 2 nodes identified by this keyword will be constrained to the same temperature in a thermal problem or in a couple thermal-mechanical problem.

*CONSTRAINED_SPR2

Purpose: Define a self-piercing rivet with failure. This model for a self-piercing rivet (SPR2) includes a plastic-like damage model that reduces the force and moment resultants to zero as the rivet fails. The domain of influence is specified by a diameter, which should be approximately equal to the rivet's diameter.

The location of the rivet is defined by a single node at the center of two riveted sheets. The algorithm does a normal projection from the master and slave sheets to the rivet node and locates all nodes within the user-defined diameter of influence. The numerical implementation of this rivet model was developed by L. Olovsson of Impetus Afea, based on research work on SPR point connector models originally carried out by SIMLab (NTNU) and SINTEF, see references by Porcaro, Hanssen, and *et.al*. [2006, 2006, 2007].

Originally only two sheets (master and slave) could be connected with one SPR2 node. But since release R9, up to 6 sheets can be connected with one SPR2 by defining additional parts on optional card 4. The following stacking sequence should be used: MID – XPID1 – XPID2 – XPID3 – XPID4 – SID. Omitted parts can be left blank, e.g. for a 3-sheet connection the extra part lies in between master and slave, and for a regular 2-sheet connection card 4 can be dropped completely.

Card 1	1	2	3	4	5	6	7	8
Variable	MID	SID	NSID	THICK	D	FN	FT	DN
Type	I	I	I	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

Card 2	1	2	3	4	5	6	7	8
Variable	DT	XIN	XIT	ALPHA1	ALPHA2	ALPHA3	DENS	INTP
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	0.0

Card 3 is optional.

Card 3	1	2	3	4	5	6	7	8
Variable	EXPN	EXPT						
Type	F	F						
Default	8.0	8.0						

Card 4 is optional.

Card 4	1	2	3	4	5	6	7	8
Variable	XPID1	XPID2	XPID3	XPID4				
Type	I	I	I	I				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
MID	Master sheet Part ID
SID	Slave sheet Part ID
NSID	Node set ID of rivet location nodes.
THICK	Total thickness of master and slave sheet.
D	Rivet diameter.
FN	Rivet strength in tension (pull-out).
FT	Rivet strength in pure shear.
DN	Failure displacement in normal direction.
DT	Failure displacement in tangential direction.
XIN	Fraction of failure displacement at maximum normal force.
XIT	Fraction of failure displacement at maximum tangential force.

VARIABLE	DESCRIPTION
ALPHA1	Dimensionless parameter scaling the effective displacement.
ALPHA2	Dimensionless parameter scaling the effective displacement.
ALPHA3	Dimensionless parameter scaling the effective displacement. The sign of ALPHA3 can be used to choose the normal update procedure: GT.0: incremental update (default), LT.0: total update (recommended).
DENS	Rivet density (necessary for time step calculation).
INTP	Flag for interpolation. EQ.0: linear (default), EQ.1: uniform, EQ.2: inverse distance weighting.
EXPN	Exponent value for load function in normal direction.
EXPT	Exponent value for load function in tangential direction.
XPID1	Extra part id 1 for multi-sheet connection.
XPID2	Extra part id 2 for multi-sheet connection.
XPID3	Extra part id 3 for multi-sheet connection.
XPID4	Extra part id 4 for multi-sheet connection.

Self-piercing rivets are a type of fastener that is sometimes used in place of spot welds to join sheet metal of similar or dissimilar materials. The rivet penetrates the first sheet, expands to interlock with the lower sheet without penetration. The strength and fatigue characteristics of self-piercing rivets can meet or even exceed that of spot welds; consequently, their practical applications are expanding.

In the local description of the underlying model, all considerations are done in the plane-of-maximum opening defined by

$$\hat{\mathbf{n}}_o = \hat{\mathbf{n}}_s \times \hat{\mathbf{n}}_m .$$

The unit normal vectors of the slave and master sheets are $\hat{\mathbf{n}}_s$ and $\hat{\mathbf{n}}_m$ respectively (see Figure 10-39), and tangential unit normal vector of the rivet is

$$\hat{\mathbf{n}}_t = \hat{\mathbf{n}}_o \times \hat{\mathbf{n}}_m .$$

A single-sheet rivet system is assumed, i.e. the rivet translation and rotation follow the motion of the master sheet. The opening appears at the slave sheet.

The local deformation is defined by normal stretch vector $\boldsymbol{\delta}_n$, tangential stretch $\boldsymbol{\delta}_t$ and total stretch $\boldsymbol{\delta} = \boldsymbol{\delta}_n + \boldsymbol{\delta}_t$ (see Figure 10-41). At any given time the total stretch is computed from the position vectors: $\boldsymbol{\delta} = \mathbf{x}_s^r - \mathbf{x}_s^s$ so that the scalar measures of normal stretch and tangential stretch are $\delta_{\boldsymbol{n}} = \boldsymbol{\delta} \cdot \hat{\mathbf{n}}_n$ and $\delta_t = \boldsymbol{\delta} \cdot \hat{\mathbf{n}}_t$. The normal and tangential forces f_n and f_t are then determined by the material model, which will be explained next.

The moments on the rivet always satisfy,

$$M_m + M_s = (h_1 + h_2) f_t / 2.$$

The motion, the forces and moments are then distributed to the nodes within the radius of influence by a weighting function, which is, by default, linear (see parameter INTP).

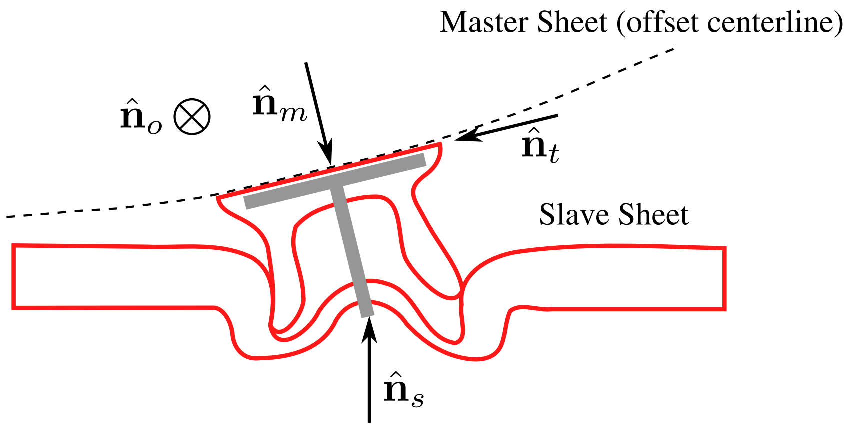


Figure 10-39. Plane of maximim opening.

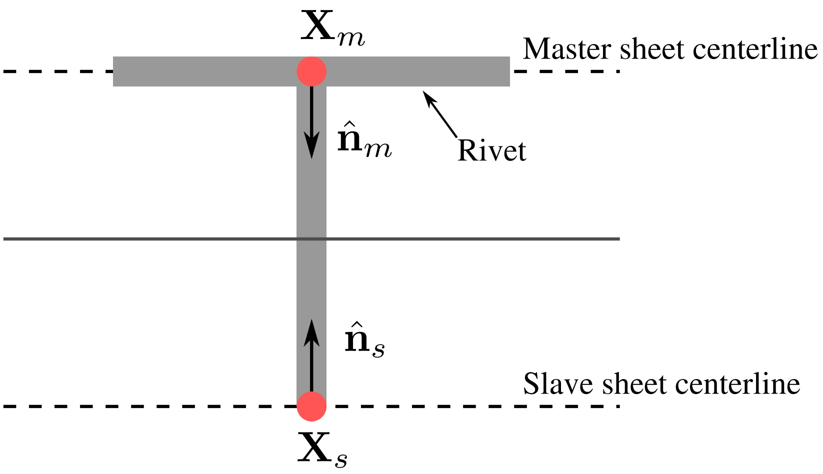


Figure 10-40. Single-sheet rivet system.

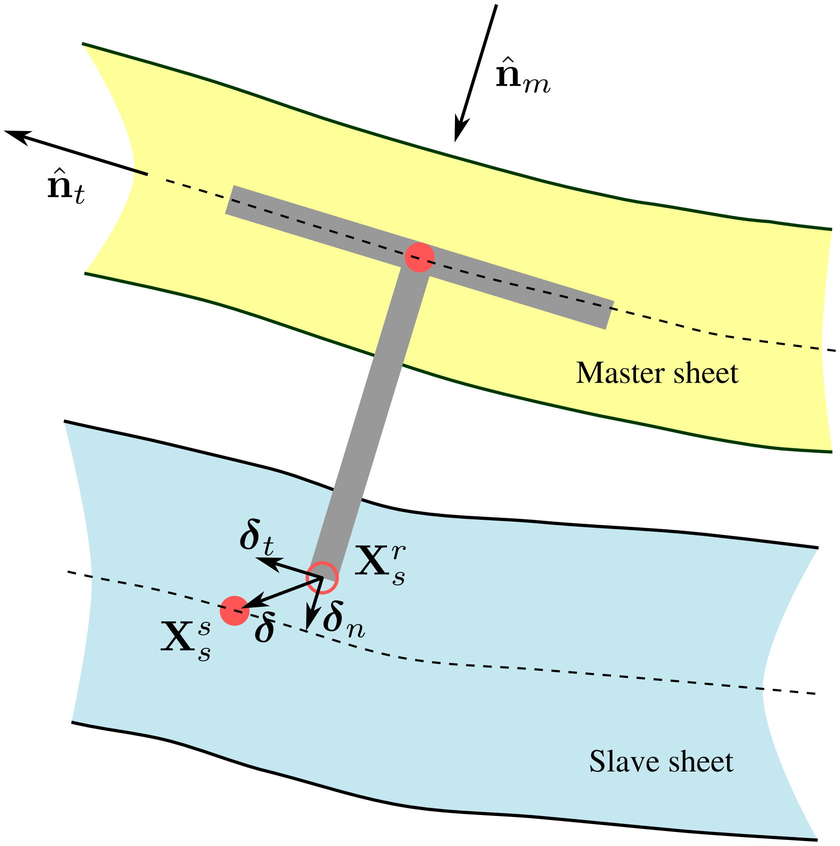


Figure 10-41. Local kinematics.

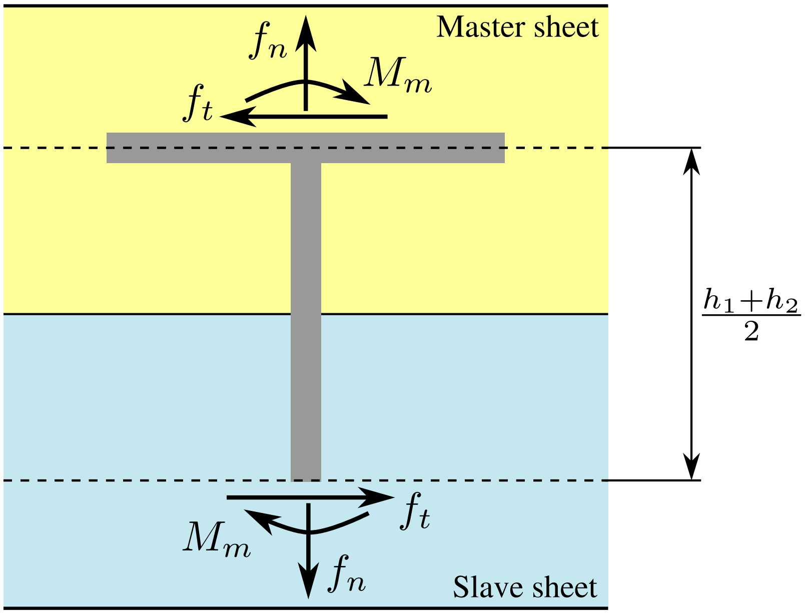


Figure 10-42. Local forces/moments.

The force-deformation relationship is defined by a non-linear damage model for arbitrary mixed-mode loading conditions (combination of tension and shear). For pure tensile and pure shear loading, the behavior is given by,

$$f_n = \frac{f_n^{\max}\delta_n}{\eta_{\max}\delta_n^{\text{fail}}}\hat{f}_n(\eta_{\max}), \qquad f_t = \frac{f_t^{\max}\delta_t}{\eta_{\max}\delta_t^{\text{fail}}}\hat{f}_t(\eta_{\max}) \tag{1}$$

Respectively where,

$$\hat{f}_n(\eta_{\max}) = \begin{cases} 1 - \left(\frac{\xi_n - \eta_{\max}}{\xi_n}\right)^{\text{EXPN}} & \eta_{\max} \le \xi_n \\ 1 - \frac{\eta_{\max} - \xi_n}{1 - \xi_n} & \eta_{max} > \xi_n \end{cases}$$
$$\hat{f}_t(\eta_{\max}) = \begin{cases} 1 - \left(\frac{\xi_t - \eta_{\max}}{\xi_t}\right)^{\text{EXPT}} & \eta_{\max} \le \xi_t \\ 1 - \frac{\eta_{\max} - \xi_t}{1 - \xi_t} & \eta_{max} > \xi_t \end{cases} \tag{2}$$

In pure tension and pure shear the damage measure, $\eta_{\max}(t)$, defined in (3), simplifies to coincide with strain as indicated in figure 10-43.

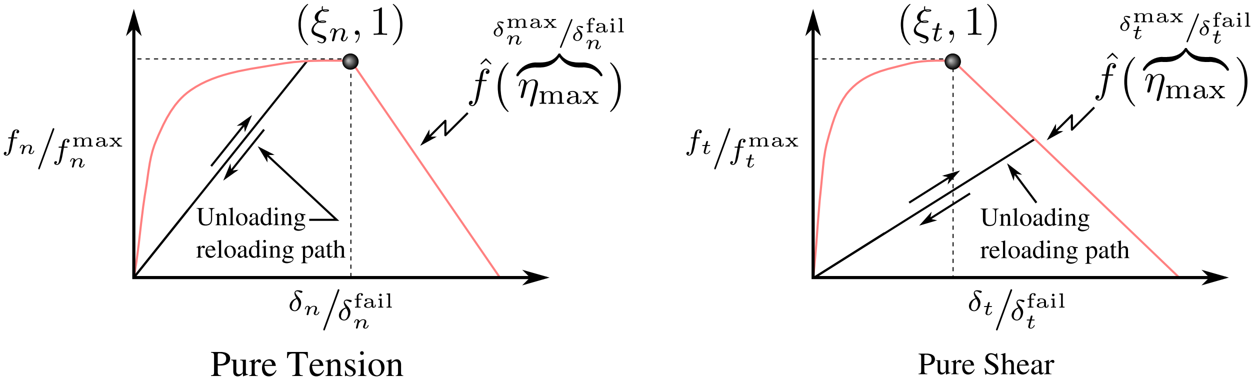


Figure 10-43. Force response of self penetrating rivet.

Usually, the material parameters $f_n^{\max}$, $f_t^{\max}$, δ_n^{fail}, and δ_t^{fail} can be determined directly from experiments, whereas material parameters ξ_n, and ξ_t can be found by reverse engineering. For mixed-mode behavior, an effective displacement measure, $\eta(\theta)$, is given by

$$\eta(\theta, \eta_{max}, t\ ) = \left[\xi(\theta) + \frac{1 - \ \xi(\theta)}{\alpha(\eta_{\max})}\right] \sqrt{\left[\frac{\delta_n(t)}{\delta_n^{\text{fail}}}\right]^2 + \left[\frac{\delta_t(t)}{\delta_t^{\text{fail}}}\right]^2}, \tag{3}$$

where,

$$\theta = \arctan\left(\frac{\delta_n}{\delta_t}\right)$$

$$\eta_{\max}(t) = \max[\eta(t)].$$

The parameter $\xi(\theta)$ which ranges from 0 to 1 scales the effective displacement as a function of the direction of the displacement vector in the δ_n-δ_t-plane according to,

$$\xi(\theta) = 1 - \frac{27}{4}\left(\frac{2\theta}{\pi}\right)^2 + \frac{27}{4}\left(\frac{2\theta}{\pi}\right)^3. \tag{4}$$

The directional scaling of the effective displacement is allowed to change as damage develops, which is characterized by the shape coefficient $\alpha(\eta_{\max})$ defined as

$$\alpha(\eta_{\max}) = \begin{cases} \dfrac{\xi_t - \eta_{\max}}{\xi_t}\alpha_1 + \dfrac{\eta_{\max}}{\xi_t}\alpha_2 & \eta_{\max} < \xi_t \\ \dfrac{1 - \eta_{\max}}{1 - \xi_t}\alpha_2 + \dfrac{\eta_{\max} - \xi_t}{1 - \xi_t}\alpha_3 & \eta_{\max} \geq \xi_t \end{cases}, \tag{5}$$

where α_1, α_2, and α_3 are material parameters.

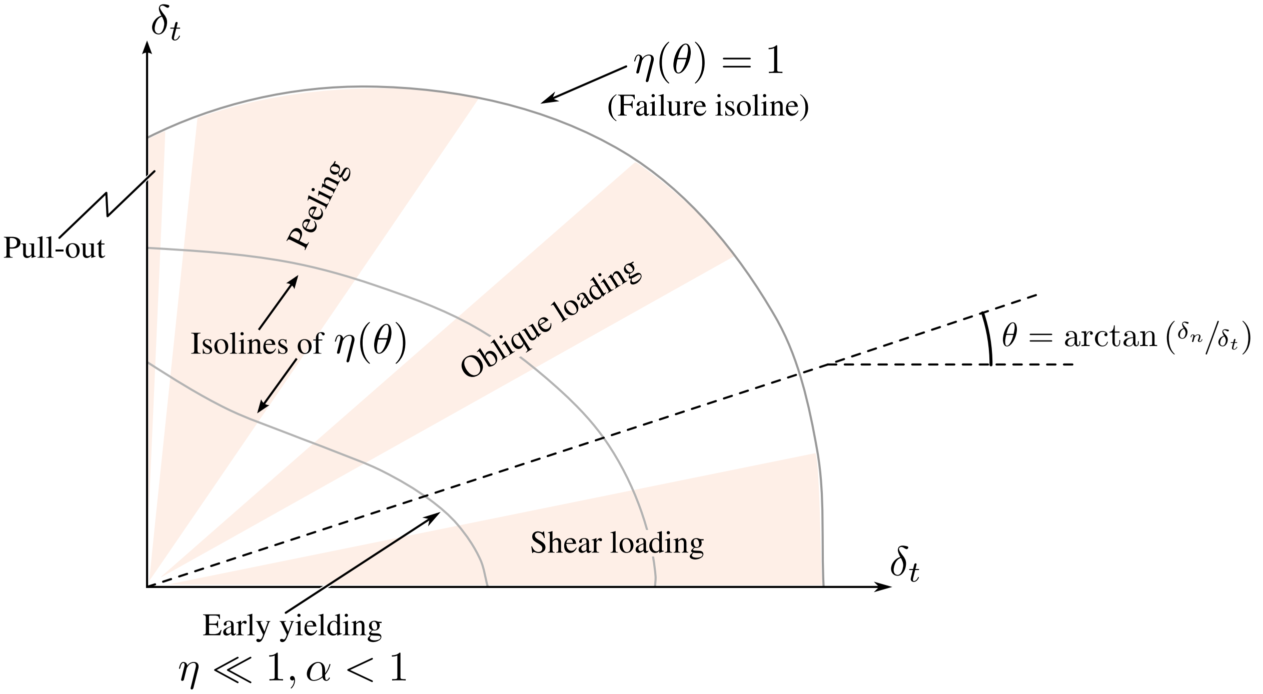


Figure 10-44. Isosurfaces of $\eta(\theta)$

The directional dependency of the effective displacement is necessary for an accurate force-displacement response in different loading directions. The coefficients $\alpha_{1,}$ and α_2 decrease the forces in the peeling and oblique loading cases to the correct levels. Both parameters are usually less than 1; whereas α_3 is typically larger than 1 as its main purpose is to moderate the failure displacement in oblique loading directions. Several qualitative features captured by this model are illustrated in Figure 10-44.

For the moment distribution, the difference between master sheet (stronger side where the rivet is entered) and slave sheet (weaker side) is accounted for by a gradual transfer from the slave to the master side as damage grows:

$$M_m = \frac{h_1 + h_2}{4}\left(1 + \frac{\eta_{\max} - \xi_1}{1 - \xi_1}\right) f_1, \qquad M_s = \frac{h_1 + h_2}{4}\left(1 - \frac{\eta_{\max} - \xi_1}{1 - \xi}\right) f_1 \tag{6}$$

Eventually the connection to the slave sheet becomes a moment free hinge.

It is recommended to use the drilling rotation constraint method for the connected components in explicit analysis, i.e. parameter DRCPSID of *CONTROL_SHELL should refer to all shell parts involved in SPR2 connections.

*CONSTRAINED_TIE-BREAK

Purpose: Define a tied shell edge to shell edge interface that can release locally as a function of plastic strain of the shells surrounding the interface nodes. A rather ductile failure is achieved.

Card 1	1	2	3	4	5	6	7	8
Variable	SNSID	MNSID	EPPF					
Type	I	I	F					
Default	none	none	0.					
Remarks		1, 2	3, 4					

VARIABLE	DESCRIPTION
SNSID	Slave node set ID, see *SET_NODE_*OPTION*.
MNSID	Master node set ID, see *SET_NODE_*OPTION*.
EPPF	Plastic strain at failure

Remarks:

1. Nodes in the master node set must be given in the order they appear as one moves along the edge of the surface.

2. Tie-breaks may not cross.

3. Tie-breaks may be used to tie shell edges together with a failure criterion on the joint. If the average volume-weighted effective plastic strain in the shell elements adjacent to a node exceeds the specified plastic strain at failure, the node is released. The default plastic strain at failure is defined for the entire tie-break but can be overridden in the slave node set to define a unique failure plastic strain for each node.

4. Tie-breaks may be used to simulate the effect of failure along a predetermined line, such as a seam or structural joint. When the failure criterion is reached in the adjoining elements, nodes along the slideline will begin to separate. As this

effect propagates, the tie-breaks will appear to “unzip,” thus simulating failure of the connection.

*CONSTRAINED_TIED_NODES_FAILURE

Purpose: Define a tied node set with failure based on plastic strain. The nodes must be coincident.

Card 1	1	2	3	4	5	6	7	8
Variable	NSID	EPPF	ETYPE					
Type	I	F	I					
Default	none	0.	0					
Remarks	1, 2, 3, 4							

VARIABLE	DESCRIPTION
NSID	Nodal set ID, see *SET_NODE_*OPTION*.
EPPF	Plastic strain, volumetric strain, or damage (MAT_224) at failure.
ETYPE	Element type for nodal group: EQ.0: shell, EQ.1: solid element

Remarks:

1. This feature applies to solid and shell elements using plasticity material models, and to solid elements using the honeycomb material *MAT_HONEYCOMB (EPPF = plastic volume strain). The failure variable is the volume strain for materials 26, 126, and 201. The failure variable is the damage for material 224, and the equivalent plastic strain is used for all other plasticity models. The specified nodes are tied together until the average volume weighted value of the failure variable exceeds the specified value. Entire regions of individual shell elements may be tied together unlike the tie-breaking shell slidelines. The tied nodes are coincident until failure. When the volume weighted average of the failure value is reached for a group of constrained nodes, the nodes of the elements that exceed the failure value are released to simulate the formation of a crack.

2. To use this feature to simulate failure, each shell element in the failure region should be generated with unique node numbers that are coincident in space with those of adjacent elements. Rather than merging these coincident nodes, the *CONSTRAINED_TIED_NODES_FAILURE option ties the nodal points together. As plastic strain develops and exceeds the failure strain, cracks will form and propagate through the mesh.

3. Entire regions of individual shell elements may be tied together, unlike the *CONSTRAINED_TIE-BREAK option. This latter option is recommended when the location of failure is known, e.g., as in the plastic covers which hide airbags in automotive structures.

4. When using surfaces of shell elements defined using the *CONSTRAINED_-TIED_NODES_FAILURE option in contact, it is best to defined each node in the surface as a slave node with the NODE_TO_SURFACE contact options. If this is not possible, the automatic contact algorithms beginning with *CONTACT_-AUTOMATIC_... all of which include thickness offsets are recommended.

Example:

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$  *CONSTRAINED_TIED_NODES_FAILURE
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Tie shell elements together at the nodes specified in nodal set 101.  The
$  constraint will be broken when the plastic strain at the nodes exceeds 0.085.
$
$  In this example, four shell elements come together at a common point.
$  The four corners of the shells are tied together with failure as opposed
$  to the more common method of merging the nodes in the pre-processing stage.
$
*CONSTRAINED_TIED_NODES_FAILURE
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$     nsid      eppf
       101     0.085
$
$
*SET_NODE_LIST
$      sid
       101
$     nid1      nid2      nid3      nid4      nid5      nid6      nid7      nid8
       775       778       896       897
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

*CONTACT

The keyword *CONTACT provides a way of treating interaction between disjoint parts. Different types of contact may be defined:

CONTACT_*OPTION1_{OPTION2}_{OPTION3}_{OPTION4}_{OPTION5}

*CONTACT_ADD_WEAR

*CONTACT_AUTO_MOVE

*CONTACT_COUPLING

*CONTACT_ENTITY

CONTACT_GEBOD_*OPTION

*CONTACT_GUIDED_CABLE

*CONTACT_INTERIOR

*CONTACT_RIGID_SURFACE

*CONTACT_1D

CONTACT_2D_*OPTION1_{OPTION2}_{OPTION3}

The first, *CONTACT_..., is the general 3D contact algorithms. The second, *CONTACT_COUPLING, provides a means of coupling to deformable surfaces to MADYMO. The third, *CONTACT_ENTITY, treats contact using mathematical functions to describe the surface geometry for the master surface. The fourth, *CONTACT_GEBOD is a specialized form of the contact entity for use with the rigid body dummies (see *COMPONENT_GEBOD). The fifth, *CONTACT_INTERIOR, is under development and is used with soft foams where element inversion is sometimes a problem. Contact between layers of brick elements is treated to eliminate negative volumes. The sixth, *CONTACT_RIGID_SURFACE is for modeling road surfaces for durability and NVH calculations. The seventh, *CONTACT_1D, remains in LS-DYNA for historical reasons, and is sometimes still used to model rebars which run along edges of brick elements. The last, *CONTACT_2D, is the general 2D contact algorithm based on those used previously in LS-DYNA2D.

CONTACT_*OPTION1_{OPTION2}_{OPTION3}_{OPTION4}_{OPTION5}_{OPTION6}

Purpose: Define a contact interface in a 3D model. For contact in 2D models, see *CONTACT_2D_*OPTION*.

OPTIONS FOR *CONTACT KEYWORD

OPTION	REQUIRED	DESCRIPTION
OPTION1	Yes	Specifies contact type
OPTION2	No	Flag for thermal
OPTION3	No	Flag indicating ID cards follow
OPTION4	No	Offset options
OPTION5	No	Flag for MPP
OPTION6	No	Flag for orthotropic friction

Allowed values for *OPTION1*

All contact types are available for explicit and implicit calculations.

AIRBAG_SINGLE_SURFACE

AUTOMATIC_BEAMS_TO_SURFACE

AUTOMATIC_GENERAL

AUTOMATIC_GENERAL_EDGEONLY

AUTOMATIC_GENERAL_INTERIOR

AUTOMATIC_NODES_TO_SURFACE

AUTOMATIC_NODES_TO_SURFACE_SMOOTH

AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE

AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE_TIEBREAK

AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE_SMOOTH

AUTOMATIC_SINGLE_SURFACE

AUTOMATIC_SINGLE_SURFACE_MORTAR

AUTOMATIC_SINGLE_SURFACE_SMOOTH

AUTOMATIC_SINGLE_SURFACE_TIED

AUTOMATIC_SURFACE_TO_SURFACE

AUTOMATIC_SURFACE_TO_SURFACE_MORTAR

AUTOMATIC_SURFACE_TO_SURFACE_MORTAR_TIED

AUTOMATIC_SURFACE_TO_SURFACE_TIED_WELD

AUTOMATIC_SURFACE_TO_SURFACE_TIEBREAK

AUTOMATIC_SURFACE_TO_SURFACE_TIEBREAK_MORTAR

AUTOMATIC_SURFACE_TO_SURFACE_SMOOTH

CONSTRAINT_NODES_TO_SURFACE

CONSTRAINT_SURFACE_TO_SURFACE

DRAWBEAD

ERODING_NODES_TO_SURFACE

ERODING_SINGLE_SURFACE

ERODING_SURFACE_TO_SURFACE

FORCE_TRANSDUCER_CONSTRAINT

FORCE_TRANSDUCER_PENALTY

FORMING_NODES_TO_SURFACE

FORMING_NODES_TO_SURFACE_SMOOTH

FORMING_ONE_WAY_SURFACE_TO_SURFACE

FORMING_SURFACE_TO_SURFACE_MORTAR

FORMING_ONE_WAY_SURFACE_TO_SURFACE_SMOOTH

FORMING_SURFACE_TO_SURFACE

FORMING_SURFACE_TO_SURFACE_SMOOTH

NODES_TO_SURFACE

NODES_TO_SURFACE_INTERFERENCE

NODES_TO_SURFACE_SMOOTH

ONE_WAY_SURFACE_TO_SURFACE

ONE_WAY_SURFACE_TO_SURFACE_INTERFERENCE

ONE_WAY_SURFACE_TO_SURFACE_SMOOTH

RIGID_NODES_TO_RIGID_BODY

RIGID_BODY_ONE_WAY_TO_RIGID_BODY

RIGID_BODY_TWO_WAY_TO_RIGID_BODY

SINGLE_EDGE

SINGLE_SURFACE

SLIDING_ONLY

SLIDING_ONLY_PENALTY

SPOTWELD

SPOTWELD_WITH_TORSION

SPOTWELD_WITH_TORSION_PENALTY

SURFACE_TO_SURFACE

SURFACE_TO_SURFACE_INTERFERENCE

SURFACE_TO_SURFACE_SMOOTH

SURFACE_TO_SURFACE_CONTRACTION_JOINT

TIEBREAK_NODES_TO_SURFACE

TIEBREAK_NODES_ONLY

TIEBREAK_SURFACE_TO_SURFACE

TIED_NODES_TO_SURFACE

TIED_SHELL_EDGE_TO_SURFACE

TIED_SHELL_EDGE_TO_SOLID

TIED_SURFACE_TO_SURFACE

TIED_SURFACE_TO_SURFACE_FAILURE

Allowed values for *OPTION2*:

THERMAL

THERMAL_FRICTION

> **NOTE:** *THERMAL* and *THERMAL_FRICTION* options are restricted to contact types having *"SURFACE_TO_-SURFACE"* in *OPTION1*.

Allowed value for *OPTION3*:

ID

Allowed values for *OPTION4*:

OPTION4 specifies that offsets may be used with the tied contact types. If one of these three offset options is set, then offsets are permitted for these contact types, and, if not, the nodes are projected back to the contact surface during the initialization phase and a constraint formulation is used. Note that in a constraint formulation, the nodes of rigid bodies are not permitted in the definition.

OFFSET

The OFFSET option switches the formulation from a constraint type formulation to one that is penalty based where the force and moment (if applicable) resultants are transferred by discrete spring elements between the slave nodes and master segments.

OFFSET is available when *OPTION1* is:

TIED_NODES_TO_SURFACE

TIED_SHELL_EDGE_TO_SURFACE

TIED_SURFACE_TO_SURFACE

With this option, there is no coupling between the transmitted forces and moments and thus equilibrium is not enforced. In the TIED_SHELL_EDGE_TO_-SURFACE contact, the BEAM_OFFSET option may be preferred since corresponding moments accompany transmitted forces. Rigid bodies can be used with this option.

BEAM_OFFSET

The BEAM_OFFSET option switches the formulation from a constraint type formulation to one that is penalty based. Beam-like springs are used to transfer force and moment resultants between the slave nodes and the master segments. Rigid bodies can be used with this option.

BEAM_OFFSET is available when *OPTION1* is:

TIED_SHELL_EDGE_TO_SURFACE

SPOTWELD

CONSTRAINED_OFFSET

The CONSTRAINED_OFFSET option is a constraint type formulation.

CONSTRAINED_OFFSET is available when *OPTION1* is:

TIED_NODES_TO_SURFACE

TIED_SHELL_EDGE_TO_SURFACE

TIED_SURFACE_TO_SURFACE

SPOTWELD

Allowed value *OPTION5*:

MPP

Allowed value for *OPTION6*:

ORTHO_FRICTION

Remarks

1. **Smooth Contact.** For SMOOTH contact, a smooth curve-fitted surface is used to represent the master segment, so that it can provide a more accurate representation of the actual surface, reduce the contact noise, and produce smoother results with coarser meshes. All contact options that include SMOOTH are available for MPP. Only the FORMING contacts, wherein the master side is rigid, can be used with SMOOTH in the case of SMP.

For SURFACE_TO_SURFACE and SINGLE_SURFACE contacts with SMOOTH in MPP, both the slave and master sides are smoothed every cycle, thereby slowing the contact treatment considerably.

The SMOOTH option does not apply to segment based (SOFT = 2) contacts.

2. **Automatic General Contact.** *CONTACT_AUTOMATIC_GENERAL is a single surface contact similar to *CONTACT_AUTOMATIC_SINGLE_SURFACE but which includes treatment of beam-to-beam contact and in doing so, checks along the entire length of the beams for penetration. *CONTACT_AUTOMATIC_GENERAL essentially adds null beams to the exterior edges of shell parts so that edge-to-edge treatment of the shell parts is handled by virtue of contact of the automatically-generated null beams. By adding the word INTERIOR to *CONTACT_AUTOMATIC_GENERAL, the contact algorithm goes a step further by adding null beams to all the shell meshlines, both along the exterior, unshared edges and the interior, shared shell edges. The EDGEONLY option skips the node-to-surface contact and does only the edge-to-edge and beam-to-beam contact.

3. **Recommendations for TIED Contact Types**. For tying solids-to-solids, that is, for situations where none of the nodes have rotational degrees-of-freedom, use TIED_NODES_TO_SURFACE and TIED_SURFACE_TO_SURFACE type contacts. These contact types may include the OFFSET or CONSTRAINED_-OFFSET option.

   For tying shells-to-shells, beams-to-shells, that is, for situations where all the nodes have rotational degrees-of-freedom, use TIED_SHELL_EDGE_TO_SURFACE type contacts. This contact type may include the OFFSET, CONSTRAINED_OFFSET, or BEAM_OFFSET option.

   TIED_SHELL_EDGE_TO_SOLID is intended for tying shell edges to solids or beam ends to solids, that is, situations where only the slave side nodes have rotational degrees-of-freedom.

4. **Tied Contact Types and the Implicit Solver.** Non-physical results have been observed when the implicit time integrator is used for models that combine tied contact formulations with automatic single point constraints on solid element rotational degrees of freedom (AUTOSPC on *CONTROL_IMPLICIT_-SOLVER). The following subset of tied interfaces support a *strongly objective* mode (see next paragraph) and verified to behave correctly with the implicit time integrator:

   1) TIED_NODES_TO_SURFACE_CONSTRAINED_OFFSET

   2) TIED_NODES_TO_SURFACE_OFFSET

3) TIED_SHELL_EDGE_TO_SURFACE_CONSTRAINED_OFFSET

4) TIED_SHELL_EDGE_TO_SURFACE_BEAM_OFFSET

The first two of these ignore rotational degrees of freedom, while the third and fourth constrain rotations. The first and third are constraint based; while the second and fourth are penalty based. These four contact types are intended to cover most use scenarios.

Setting IACC = 1 on *CONTROL_ACCURACY activates the *strongly objective formulation* for the above mentioned contacts (as well as the non-offset options *CONTACT_TIED_NODES_TO_SURFACE and *CONTACT_TIED_SHELL_-EDGE_TO_SURFACE as a side effect). When active, forces and moments transform correctly under superposed rigid body motions within a single implicit step. Additionally, this formulation applies rotational constraints consistently *when, and only when, necessary*. In particular, strong objectivity is implemented so that slave nodes without rotational degrees of freedom are not rotationally constrained, while slave nodes with bending and torsional rotations are rotationally constrained. Additionally, strong objectivity ensures that the constraint is physically correct.

For a master node belonging to a *shell*, the slave node's bending rotations (rotations in the plane of the master segment) are constrained to match the master segment's rotational degrees of freedom; for master nodes *not belonging to a shell*, the slave's bending rotations are constrained to the master segment rotation as determined from its individual nodal translations. The slave node's torsional rotations (rotations with respect to the normal of the master segment) are *always* constrained based on the master segment's torsional rotation as determined from its individual nodal translations, thus avoiding the relatively weak drilling mode of shells. This tied contact formulation properly treats bending *and* torsional rotations. Since slave node rotational degrees of freedom typically come from shell or beam elements the most frequently used options are:

TIED_SHELL_EDGE_TO_SURFACE_CONSTRAINED_OFFSET

TIED_SHELL_EDGE_TO_SURFACE_BEAM_OFFSET

The other two "non-rotational" formulations:

TIED_NODES_TO_SURFACE_CONSTRAINED_OFFSET

TIED_NODES_TO_SURFACE_OFFSET

are included for situations in which rotations do not need to be constrained at all. See the LS-DYNA Theory Manual for further details.

5. **Additional Remarks.** Additional notes on contact types and a few examples are provided at the end of this section in "General Remarks: *CONTACT". A theoretical discussion is provided in the LS-DYNA Theory Manual.

ADDITIONAL CARDS FOR *CONTACT KEYWORD

Cards *must* appear in the *exact* order listed below.

CARD	DESCRIPTION
ID	Card required when *OPTION3* set to ID option; otherwise this card is omitted.
MPP	Card required when *OPTION5* set to MPP.
Card 1	Always required.
Card 2	Always required.
Card 3	Always required.
Card 4	Required for the following permutations of *CONTACT. **NOTE:** The format of Card 4 is **different** for each option listed below. *CONTACT_AUTOMATIC_SINGLE_SURFACE_TIED *CONTACT_CONSTRAINT_*type* *CONTACT_DRAWBEAD *CONTACT_ERODING_*type* *CONTACT_…_INTERFERENCE *CONTACT_RIGID_*type* *CONTACT_TIEBREAK_*type* *CONTACT_…_CONTRACTION_JOINT_*type*
THERMAL	Required if *OPTION2* is set. Otherwise omit.
THERMAL_FRICTION	Required if *OPTION2* is set to THERMAL_FRICTION. Otherwise omit.
ORTHO_FRICTION	Required if *OPTION6* is set. Otherwise omit. Contains friction coefficients

CARD	DESCRIPTION
Optional Card A	Optional parameters. **NOTE:** Default values are highly optimized. **NOTE:** *Required* if Optional Card B is included. If Optional Card A is a blank line, then values are set to their defaults, and Optional Card B may follow.
Optional Card B	Optional parameters. *Required* if Optional Card C is included. (See Optional Card A note; similar logic applies)
Optional Card C	Optional parameters. *Required* if Optional Card D is included. (See Optional Card A note; similar logic applies.)
Optional Card D	Optional parameters. *Required* if Optional Card E is included. (See Optional Card A note; similar logic applies.)
Optional Card E	Optional parameters.

ID Card.

Additional keyword for ID keyword option.

ID	1	2	3	4	5	6	7	8
Variable	CID	HEADING						
Type	I	A70						

The contact ID is needed during full deck restarts for contact initialization. If the contact ID is undefined, the default ID is determined by the sequence of the contact definitions, i.e., the first contact definition has an ID of 1, the second, 2, and so forth. In a full deck restart without contact IDs, for a successful run no contact interfaces can be deleted and those which are added must be placed after the last definition in the previous run. The ID and heading is picked up by some of the peripheral LS-DYNA codes to aid in post-processing.

VARIABLE	DESCRIPTION
CID	Contact interface ID. This must be a unique number.
HEADING	Interface descriptor. It is suggested that unique descriptions be used.

MPP Cards. Variables set with these cards are only active when using MPP LS-DYNA.

MPP Card 1. Additional card for the MPP option. This card is ignored, but still read in, when SOFT = 2 on optional card A.

MPP 1	1	2	3	4	5	6	7	8
Variable	IGNORE	BCKT	LCBCKT	NS2TRK	INITITR	PARMAX		CPARM8
Type	I	I	I	I	I	F		I
Default	0	200	none	3	2	See below		0

MPP Card 2. The keyword reader will interpret the card following MPP Card 1 as MPP Card 2 if the first column of the card is occupied by an ampersand. Otherwise, it is interpreted as Card 1. This card is ignored, but still read in, when SOFT = 2 on optional card A.

MPP 2	1	2	3	4	5	6	7	8
Variable	&	CHKSEGS	PENSF	GRPABLE				
Type		I	F	I				
Default		0	1.0	0				

VARIABLE	DESCRIPTION
IGNORE	This is the same as the "ignore initial penetrations" option on the *CONTROL_CONTACT Optional Card C entry 2 and can also be specified in the normal contact control cards. It predates both of those, and is not really needed anymore since both are honored by the MPP code. That is, if any of the three are on, initial penetrations are tracked.
BCKT	Bucket sort frequency, this parameter does not apply when SOFT = 2 on optional card A or to the Mortar contact (option MORTAR on the CONTACT card). For the two exceptions, the BSORT option on Optional Card A applies instead.
LCBCKT	Load curve for bucket sort frequency, this parameter does not apply when SOFT = 2 on optional card A or to the Mortar contact (option MORTAR on the CONTACT card). For the two exceptions, the negative BSORT option on Optional Card A applies instead.
NS2TRK	Number of potential contacts to track for each slave node. The normal input for this (DEPTH on Optional Card A) is ignored.
INITITR	Number of iterations to perform when trying to eliminate initial penetrations. Note: an input of 0 means 0, not the default value (which is 2). Leaving this field blank will set INITITR to 2.
PARMAX	The parametric extension distance for contact segments. The MAXPAR parameter on Optional Card A is not used for MPP. For non-TIED contacts, the default is 1.0005. For TIED contacts the default is 1.035 and, the actual extension used is computed as follows:

$$\mathrm{PARMAX_{computed}} = \begin{cases} 1.0 + \mathrm{PARMAX} & 0.0 < \mathrm{PARMAX} < 0.5 \\ \mathrm{PARMAX} & 1.0 \le \mathrm{PARMAX} \le 1.0004 \\ \max(\mathrm{PARMAX}, 1.035) & \text{otherwise} \end{cases}$$

VARIABLE DESCRIPTION

CPARM8

Flag for CONTACT_AUTOMATIC_GENERAL behavior. CPARM8's value is interpreted as two separate flags: OPT1 and OPT2 according to the rule,

$$\text{CPARM8} = \text{OPT1} + \text{OPT2}.$$

When OPT1 and OPT2 are *both* set, *both* options are active.

OPT1: Flag to exclude beam-to-beam contact from the same PID.

EQ.0: Flag is not set (default).

EQ.1: Flag is set.

EQ.2: Flag is set. CPARM8 = 2 has the additional effect of permitting contact treatment of spot weld (type 9) beams in AUTOMATIC_GENERAL contacts; spot weld beams are otherwise disregarded entirely by AUTOMATIC_GENERAL contacts.

OPT2: Flag to shift generated beam affecting only shell-edge-to-shell-edge treatment. See also SRNDE in Optional Card E.

EQ.10: Beam generated on exterior shell edge will be shifted into the shell by half the shell thickness. Therefore, the shell-edge-to-shell-edge contact starts right at the shell edge and not at an extension of the shell edge.

CHKSEGS

If this value is non-zero, then the node to surface and surface to surface contacts will perform a special check at time 0 for elements that are inverted (or nearly so), and remove them from contact. These poorly formed elements have been known to occur on the tooling in metalforming problems, which allows these problems to run. It should not normally be needed for reasonable meshes.

PENSF

This option is used together with IGNORE for 3D forging problems. If non-zero, the IGNORED penetration distance is multiplied by this value each cycle, effectively pushing the slave node back out to the surface. This is useful for nodes that might get generated below the master surface during 3D remeshing. Care should be exercised, as energy may be generated and stability may be effected for values lower than 0.95. A value in the range of 0.98 to 0.99 or higher (but < 1.0) is recommended.

VARIABLE	DESCRIPTION
GRPABLE	Set to 1 to invoke an alternate MPP communication algorithm for SINGLE_SURFACE, NODE_TO_SURFACE, and SURFACE_TO_SURFACE contacts. The new algorithm does not support all contact options, including SOFT = 2, as of yet, and is still under development. It can be significantly faster and scale better than the normal algorithm when there are more than two or three applicable contact types defined in the model. Its intent is to speed up the contact processing but not to change the behavior of the contact. See also *CONTROL_MPP_CONTACT_GROUPABLE.

Remarks:

1. The MPP cards are ignored by the segment based contact options that are made active by setting SOFT = 2 on optional card A. When SOFT = 2. The BSORT parameter on optional card A can be used to override the default bucket sort frequency.

Card 1.

Card 1	1	2	3	4	5	6	7	8
Variable	SSID	MSID	SSTYP	MSTYP	SBOXID	MBOXID	SPR	MPR
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none			0	0
Remarks	1	2			optional	optional	0 = off	0 = off

VARIABLE	DESCRIPTION
SSID	Slave segment, node set ID, part set ID, part ID, or shell element set ID, see *SET_SEGMENT, *SET_NODE_*OPTION*, *PART, *SET_PART or *SET_SHELL_*OPTION*. For ERODING_SINGLE_-SURFACE and ERODING_SURFACE_TO_SURFACE contact types, use either a part ID or a part set ID. For ERODING_-NODES_TO_SURFACE contact, use a node set which includes all nodes that may be exposed to contact as element erosion occurs. EQ.0: all part IDs are included for single surface contact, automatic single surface, and eroding single surface.
MSID	Master segment set ID, part set ID, part ID, or shell element set ID, see *SET_SEGMENT, *SET_NODE_*OPTION*, *PART, *SET_-PART, or *SET_SHELL_*OPTION*: EQ.0: for single surface contact, automatic single surface, and eroding single surface.

VARIABLE **DESCRIPTION**

SSTYP

ID type of SSID:

EQ.0: segment set ID for surface-to-surface contact,

EQ.1: shell element set ID for surface-to-surface contact,

EQ.2: part set ID,

EQ.3: part ID,

EQ.4: node set ID for node to surface contact,

EQ.5: include all (SSID is ignored),

EQ.6: part set ID for exempted parts. All non-exempted parts are included in the contact.

For *AUTOMATIC_BEAMS_TO_SURFACE contact either a part set ID or a part ID can be specified.

MSTYP

ID type of MSID:

EQ.0: segment set ID,

EQ.1: shell element set ID,

EQ.2: part set ID,

EQ.3: part ID.

EQ.4: node set ID (for eroding force transducer only. See remark 3),

EQ.5: include all (MSID is ignored).

SBOXID

Include in contact definition only those slave nodes/segments within box SBOXID (corresponding to BOXID in *DEFINE_BOX), or if SBOXID is negative, only those slave nodes/segments within contact volume |SBOXID| (corresponding to CVID in *DEFINE_CONTACT_VOLUME). SBOXID can be used only if SSTYP is set to 2 or 3, i.e., SSID is a part ID or part set ID. SBOXID is not available for_ERODING contact options.

MBOXID

Include in contact definition only those master segments within box MBOXID (corresponding to BOXID in *DEFINE_BOX), or if MBOXID is negative, only those master segments within contact volume |MBOXID| (corresponding to CVID in *DEFINE_CONTACT_VOLUME). MBOXID can be used only if MSTYP is set to 2 or 3, i.e., MSID is a part ID or part set ID. MBOXID is not available for_ERODING contact options

VARIABLE **DESCRIPTION**

SPR

Include the slave side in the *DATABASE_NCFORC and the *DATABASE_BINARY_INTFOR interface force files, and optionally in the dynain file for wear:

EQ.1: slave side forces included.

EQ.2: same as EQ.1, but also allows for slave nodes to be written as *INITIAL_CONTACT_WEAR to dynain, see NCYC on *INTERFACE_SPRINGBACK_LSDYNA.

MPR

Include the master side in the *DATABASE_NCFORC and the *DATABASE_BINARY_INTFOR interface force files, and optionally in the dynain file for wear:

EQ.1: master side forces included.

EQ.2: same as EQ.1, but also allows for master nodes to be written as *INITIAL_CONTACT_WEAR to dynain, see NCYC on *INTERFACE_SPRINGBACK_LSDYNA.

Remarks:

1. Giving a slave set ID equal to zero is valid only for the single surface contact algorithms, i.e., the options:

   SINGLE_SURFACE

   AUTOMATIC_…

   AIRBAG_…

   ERODING_SINGLE_SURFACE

2. A master set ID is not defined for the single surface contact algorithms (including AUTOMATIC_GENERAL). A master set ID is optional for FORCE_TRANSDUCERS. If a master set is defined for the FORCE_TRANSDUCER option, only those force that develop between and master and slave surfaces are considered. If a transducer is used for extracting forces from Mortar contacts, the slave and master sides *must* be defined through parts or part sets, segment or node sets will not gather the correct data.

NOTE: The master surface option of FORCE_TRANSDUCER is only implemented for the PENALTY option and works only in conjunction with the AUTOMATIC_SINGLE_SURFACE contact types, except as noted in the next remark.

3. A master node set can only be used with the TRANSDUCER_PENALTY option, and requires that the slave side also be defined via a node set. This allows the transducer to give correct results for eroding materials. The node sets should include all nodes that may be exposed as erosion occurs. This option does not apply to Mortar contacts.

Card 2.

Card 2	1	2	3	4	5	6	7	8
Variable	FS	FD	DC	VC	VDC	PENCHK	BT	DT
Type	F	F	F	F	F	I	F	F
Default	0.	0.	0.	0.	0.	0	0.	1.0E20
Remarks								

VARIABLE **DESCRIPTION**

If *OPTION1* is TIED_SURFACE_TO_SURFACE_FAILURE, then

FS Normal tensile stress at failure. failure occurs if

$$\left[\frac{\max(0.0, \sigma_{\text{normal}})}{FS}\right]^2 + \left[\frac{\sigma_{\text{shear}}}{FD}\right]^2 > 1$$

where σ_{normal} and σ_{shear} are the interface normal and shear stresses.

FD Shear stress at failure. See FS.

Else

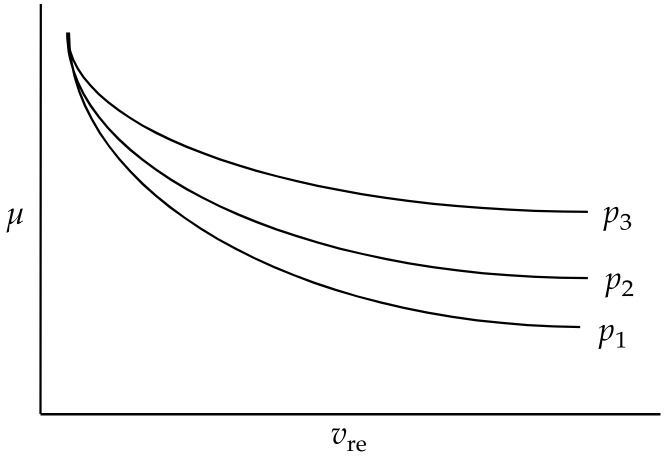


Figure 11-1. Friction coefficient, μ, can be a function of relative velocity and pressure. See Remarks for FS = 2.0.

VARIABLE **DESCRIPTION**

FS

Static coefficient of friction. If FS is > 0 and not equal to 2. The frictional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact according to,

$$\mu_c = \mathrm{FD} + (\mathrm{FS} - \mathrm{FD})e^{-\mathrm{DC}|v_{\mathrm{rel}}|}.$$

For mortar contact μ_c = FS, i.e., dynamic effects are ignored. The two other possibilities are:

EQ.-2: If only the one friction table is defined using *DEFINE_FRICTION, it will be used and there is no need to define parameter FD. If more than one friction table is defined then the Table ID is defined by the FD Parameter below.

EQ.-1: If the frictional coefficients defined in the *PART section are to be used, set FS to the negative number, -1.0.

WARNING: Please note that the FS = -1.0 and FS = -2.0 options apply only to contact types:

SINGLE_SURFACE,

AUTOMATIC_GENERAL,

AUTOMATIC_SINGLE_SURFACE,

AUTOMATIC_SINGLE_SURFACE_MORTAR,

AUTOMATIC_NODES_TO_SURFACE,

VARIABLE	DESCRIPTION
	AUTOMATIC_SURFACE_TO_SURFACE, AUTOMATIC_SURFACE_TO_SURFACE_MORTAR, AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE, ERODING_SINGLE_SURFACE.
	EQ.2: For a subset of SURFACE_TO_SURFACE type contacts (see Remarks for Card 2 below), the variable FD serves as a table ID (see *DEFINE_TABLE). That table specifies two or more values of contact pressure, with each pressure value in the table corresponding to a curve of friction coefficient vs. relative velocity. Thus the friction coefficient becomes a function of pressure and relative velocity. See Figure 11-1.
FD	Dynamic coefficient of friction. The frictional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact according to, $\mu_c = \mathrm{FD} + (\mathrm{FS} - \mathrm{FD})e^{-DC\|v_{\mathrm{rel}}\|}$ For mortar contact $\mu_c = \mathrm{FS}$, i.e., dynamic effects are ignored. When FS = -2: If FS = -2 and more than one friction table is defined, FD is used to specify friction table to be used.
End If	
DC	Exponential decay coefficient. The frictional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact $\mu_c = \mathrm{FD} + (\mathrm{FS} - \mathrm{FD})e^{-\mathrm{DC}\|\mathrm{v_{rel}}\|}$. For mortar contact $\mu_c = \mathrm{FS}$, i.e., dynamic effects are ignored.
VC	Coefficient for viscous friction. This is necessary to limit the friction force to a maximum. A limiting force is computed $F_{\mathrm{lim}} = \mathrm{VC} \times A_{\mathrm{cont}}$. A_{cont} being the area of the segment contacted by the node in contact. The suggested value for VC is the yield stress in shear $VC = \frac{\sigma_o}{\sqrt{3}}$ where σ_0 is the yield stress of the contacted material.

VARIABLE	DESCRIPTION
VDC	Viscous damping coefficient in percent of critical or the coefficient of restitution expressed as percentage. In order to avoid undesirable oscillation in contact, e.g., for sheet forming simulation, a contact damping perpendicular to the contacting surfaces is applied. When ICOR, the 6th column of the optional E card, is not defined or 0, the applied damping coefficient is given by $$\xi = \frac{\mathrm{VDC}}{100}\xi_{\mathrm{crit}},$$ where VDC is an integer (in units of percent) between 0 and 100. The formula for critical damping is $$\xi_{\mathrm{crit}} = 2m\omega,$$ where m is determined by nodal masses as $$m = \min(m_{\mathrm{slave}}, m_{\mathrm{master}}),$$ and ω is determined from k, the interface stiffness, according to $$\omega = \sqrt{k\frac{m_{\mathrm{slave}} + m_{\mathrm{master}}}{m_{\mathrm{master}} m_{\mathrm{slave}}}}.$$
PENCHK	Small penetration in contact search option. If the slave node penetrates more than the segment thickness times the factor XPENE, see *CONTROL_CONTACT, the penetration is ignored and the slave node is set free. The thickness is taken as the shell thickness if the segment belongs to a shell element or it is taken as 1/20 of its shortest diagonal if the segment belongs to a solid element. This option applies to the surface-to-surface contact algorithms: See Table 11-17 for contact types and more details.
BT	Birth time (contact surface becomes active at this time). LT.0: Birth time is set to |BT|. When negative, birth time is followed during the dynamic relaxation phase of the calculation. After dynamic relaxation has completed, contact is activated regardless the value of BT. EQ.0: Birth time is inactive, i.e., contact is always active GT.0: If DT = -9999, BT is interpreted as the curve or table ID defining multiple pairs of birth-time/death-time, see remarks below. Otherwise, if DT > 0, birth time applies both during and after dynamic relaxation.
DT	Death time (contact surface is deactivated at this time).

VARIABLE **DESCRIPTION**

LT.0: If DT = -9999, BT is interpreted as the curve or table ID defining multiple pairs of birth-time/death-time. Otherwise, negative DT indicates that contact is inactive during dynamic relaxation. After dynamic relaxation the birth and death times are followed and set to |BT| and |DT| respectively.

EQ.0: DT defaults to 1.E+20.

GT.0: DT, the death time, sets the time at which the contact is deactivated.

Remarks:

The FS = 2 method of specifying the friction coefficient as a function of pressure and relative velocity is implemented in all contacts for which SOFT = 2. It is recommended that when FS = 2 and SOFT = 2 are used together, that FNLSCL be set in the range of 0.5 to 1.0 and DNLSCL be set to 0 (refer to Remark 5 under the description of Optional Card D for *CONTACT). If sliding is prevalent, DPRFAC = 0.01 on Optional Card C is also recommended.

When FS = 2 and SOFT = 0 or 1, the following ONE_WAY contacts are recommended. If sliding is prevalent, DPRFAC = 0.01 is also recommended.

ONE_WAY_SURFACE_TO_SURFACE	(SMP and MPP)
AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE	(MPP only)
FORMING_ONE_WAY_SURFACE_SURFACE_TO_SURFACE	(MPP only)

For SOFT = 0 or 1, FS = 2 is implemented but not advised for the following contacts:

SURFACE_TO_SURFACE	(SMP and MPP)
AUTOMATIC_SURFACE_TO_SURFACE	(MPP only)
FORMING_SURFACE_TO_SURFACE	(MPP only)

A caveat pertaining to the MPP contacts listed above is that the “groupable” option must not be invoked. See *CONTROL_MPP_CONTACT_GROUPABLE.

For SOFT = 0 or 1, FS = 2 is not implemented in SMP for AUTOMATIC and FORMING contact types. The static friction coefficient will literally be taken as 2.0 if FS is set to 2 for these SMP contacts.

If DT = -9999, BT is taken to be the ID of an activation curve defining multiple birth-times and death-times as ordered (x, y) pairs. A data point in the activation curve defines a time slot during which the contact is active. For example, an activation curve with two data points of $(20, 30)$ and $(50, 70)$ activates the contact when 20 ≤ time ≤ 30 and when 50 ≤ time ≤ 70. To define separate activation curves for dynamic relaxation and the subsequent dynamics, BT can be defined as a table containing two activation curves, one with VALUE = 0 for transient analysis and the other one with VALUE = 1 for dynamic relaxation, see *DEFINE_TABLE.

Card 3.

Card 3	1	2	3	4	5	6	7	8
Variable	SFS	SFM	SST	MST	SFST	SFMT	FSF	VSF
Type	F	F	F	F	F	F	F	F
Default	1.	1.	element thickness	element thickness	1.	1.	1.	1.

VARIABLE	DESCRIPTION
SFS	Scale factor on default slave penalty stiffness when SOFT = 0 or SOFT = 2, see also *CONTROL_CONTACT.
SFM	Scale factor on default master penalty stiffness when SOFT = 0 or SOFT = 2, see also *CONTROL_CONTACT.
SST	Optional contact thickness for slave surface (overrides default contact thickness). This option applies to contact with shell and beam elements. SST has no bearing on the actual thickness of the elements; it only affects the location of the contact surface. For the *CONTACT_TIED_… options, SST and MST below can be defined as negative values, which will cause the determination of whether or not a node is tied to depend only on the separation distance relative to the absolute value of these thicknesses. More information is given under General Remarks on *CONTACT following Optional Card E.
MST	Optional contact thickness for master surface (overrides default contact thickness). This option applies only to contact with shell elements. For the TIED options, see SST above.
SFST	Scale factor applied to contact thickness of slave surface. This option applies to contact with shell and beam elements. SFST has no bearing on the actual thickness of the elements; it only affects the location of the contact surface. SFST is ignored if SST is nonzero except in the case of MORTAR contact (see Contact General Remark 10).

VARIABLE	DESCRIPTION
SFMT	Scale factor applied to contact thickness of master surface. This option applies only to contact with shell elements. SFMT has no bearing on the actual thickness of the elements; it only affects the location of the contact surface. SFMT is ignored if MST is nonzero except in the case of MORTAR contact (see Contact General Remark 10).
FSF	Coulomb friction scale factor. The Coulomb friction value is scaled as $\mu_{sc} = \text{FSF} \times \mu_c$, see above.
VSF	Viscous friction scale factor. If this factor is defined then the limiting force becomes: $F_{\text{lim}} = \text{VSF} \times \text{VC} \times A_{\text{cont}}$, see above.

Remarks:

The variables FSF and VSF above can be overridden segment by segment on the *SET_-SEGMENT or *SET_SHELL_*OPTION* cards for the **slave surface only** as A3 and A4, and for the **master surface only** as A1 and A2. See *SET_SEGMENT and *SET_SHELL_ *OPTION*.

Card 4: AUTOMATIC_SURFACE_TIEBREAK

This card 4 is mandatory for:

CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_TIEBREAK_*{OPTION}

CONTACT_AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE_TIEBREAK_*{OPTION}

If the response parameter OPTION below is set to 9 or 11, three damping constants can be defined for the various failure modes. To do this, set the keyword option to

DAMPING

For OPTION = 7 and OPTION = 9 but for the automatic surface to surface contact only, the mortar treatment of the tiebreak contact may be activated. This is primarily intended for implicit analysis and no damping can be used with this option, see also remarks on mortar contacts. The keyword option for this is

MORTAR

The mortar treatment of tiebreak contact is available only for OPTION = 7 and OPTION = 9, and only with surface to surface contact, i.e., neither the ONE_WAY nor the DAMPING option is compatible with the MORTAR option.

Card 4a	1	2	3	4	5	6	7	8
Variable	OPTION	NFLS	SFLS	PARAM	ERATEN	ERATES	CT2CN	CN
Type	I	F	F	F	F	F	F	F
Default	required	required	required	0.0	0.0	0.0	1.0	0.0

Damping Card. Additional card for the case of OPTION = 9 with the DAMPING keyword option active.

Card 4b	1	2	3	4	5	6	7	8
Variable	DMP_1	DMP_2	DMP_3					
Type	F	F	F					
Default	0.0	0.0	0.0					

VARIABLE	DESCRIPTION
OPTION	Response: EQ.-3: see 3, moments are transferred. SMP only. EQ.-2: see 2, moments are transferred. SMP only. EQ.-1: see 1, moments are transferred. SMP only. EQ.1: slave nodes in contact and which come into contact will permanently stick. Tangential motion is inhibited. EQ.2: tiebreak is active for nodes which are initially in contact. Until failure, tangential motion is inhibited. If PARAM is set to unity, (1.0) shell thickness offsets are ignored, and the orientation of the shell surfaces is required such that the outward normals point to the opposing contact surface. EQ.3: as 1 above but with failure after sticking. EQ.4: tiebreak is active for nodes which are initially in contact but tangential motion with frictional sliding is permitted. EQ.5: tiebreak is active for nodes which are initially in contact. Stress is limited by the yield condition described in Remark 5 below. Damage behavior is modeled by a curve which defines normal stress vs. gap (crack opening). This option can be used to represent deformable glue bonds. EQ.6: This option is for use with solids and thick shells only. Tiebreak is active for nodes which are initially in contact. Failure stress must be defined for tiebreak to occur. After the failure stress tiebreak criterion is met, damage is a linear function of the distance C between

VARIABLE **DESCRIPTION**

points initially in contact. When the distance is equal to PARAM, damage is fully developed and interface failure occurs. After failure, this option behaves as a surface-to-surface contact.

EQ.7: Dycoss Discrete Crack Model. "…_ONE_WAY_SURFACE_TO_SURFACE_TIEBREAK" definition is recommended for this option. See Remarks.

EQ.8: This is similar to OPTION = 6, but it works with offset shell elements. "…_ONE_WAY_SURFACE_TO_SURFACE_TIEBREAK" definition is recommended for this option.

EQ.9: Discrete Crack Model with power law and B-K damage models. "…_ONE_WAY_SURFACE_TO_SURFACE_-TIEBREAK" definition is recommended for this option. See Remarks.

EQ.10: This is similar to OPTION = 7, but it works with offset shell elements. "…_ONE_WAY_SURFACE_TO_SURFACE_TIEBREAK" definition is recommended for this option.

EQ.11: This is similar to OPTION = 9, but it works with offset shell elements. "…_ONE_WAY_SURFACE_TO_SURFACE_TIEBREAK" definition is recommended for this option.

NFLS Normal failure stress for OPTION = 2, 3, 4, 6, 7, 8, 9, 10 or 11. For OPTION = 5 NFLS becomes the plastic yield stress as defined in Remark 5. For OPTION = 9 or 11 and NFLS < 0, a load curve ID = -NFLS is referenced defining normal failure stress as a function of element size. See remarks.

SFLS Shear failure stress for OPTION = 2, 3, 6, 7, 8, 9, 10 or 11. For OPTION = 4, SFLS is a frictional stress limit if PARAM = 1. This frictional stress limit is independent of the normal force at the tie. For OPTION = 5 SFLS becomes the curve ID which defines normal stress vs. gap. For OPTION = 9 or 11 and SFLS < 0, a load curve ID = -SFLS is referenced defining shear failure stress as a function of element size. See remarks.

PARAM For OPTION = 2, setting PARAM = 1 causes the shell thickness offsets to be ignored. For OPTION = 4, setting PARAM = 1 causes SFLS to be a frictional stress limit. For OPTION = 6 or 8,

VARIABLE	DESCRIPTION
	PARAM is the critical distance, CCRIT, at which the interface failure is complete. For OPTION = 7 or 10 PARAM is the friction angle in degrees. For OPTION = 9 or 11, it is the exponent in the damage model. A positive value invokes the power law, while a negative one, the B-K model. See MAT_138 for additional details.
ERATEN	For OPTION = 7, 9, 10, 11 only. Normal energy release rate (stress × length) used in damage calculation, see Lemmen and Meijer [2001].
ERATES	For OPTION = 7, 9, 10, 11 only. Shear energy release rate (stress × length) used in damage calculation, see Lemmen and Meijer [2001].
CT2CN	The ratio of the tangential stiffness to the normal stiffness for OPTION = 9, 11. The default is 1.0.
CN	Normal stiffness (stress/length) for OPTION = 9, 11, and OPTION = 7 for the MORTAR option only. If CN is not given explicitly, penalty stiffness divided by segment area is used (default). This optional stiffness should be used with care, since contact stability can get affected. A warning message with a recommended time step is given initially.
DMP_1	Mode I damping force per unit velocity per unit area.
DMP_2	Mode II damping force per unit velocity per unit area.
DMP_3	Mode III damping force per unit velocity per unit area.

Remarks:

1. After failure, this contact option behaves as a surface-to-surface contact with thickness offsets. After failure, no interface tension is possible.

2. The soft constraint option with SOFT = 2 is not implemented for the tiebreak option.

3. For OPTION = 2, 3, and 6 the tiebreak failure criterion has normal and shear components:

$$\left(\frac{|\sigma_n|}{\text{NFLS}}\right)^2 + \left(\frac{|\sigma_s|}{\text{SFLS}}\right)^2 \geq 1.$$

4. For OPTION = 4, the tiebreak failure criterion has only a normal stress component:

$$\frac{|\sigma_n|}{\text{NFLS}} \geq 1.$$

5. For OPTION = 5, the stress is limited by a perfectly plastic yield condition. For ties in tension, the yield condition is

$$\frac{\sqrt{\sigma_n^2 + 3|\sigma_s|^2}}{\text{NLFS}} \leq 1.$$

For ties in compression, the yield condition is

$$\frac{\sqrt{3|\sigma_s|^2}}{\text{NLFS}} \leq 1.$$

The stress is also scaled by the damage function which is obtained from the load curve. For ties in tension, both normal and shear stress are scaled. For ties in compression, only shear stress is scaled.

6. For OPTION = 6 or 8, damage initiates when the stress meets the failure criterion. The stress is then scaled by the damage function. Assuming no load reversals, the energy released due to the failure of the interface is approximately 0.5 × S × CCRIT, where

$$S = \sqrt{\max(\sigma_n, 0)^2 + |\sigma_s|^2}$$

at the initiation of damage. This interface may be used for simulating crack propagation. For the energy release to be correct, the contact penalty stiffness must be much larger than

$$\frac{\min(\text{NFLF, SFLS})}{\text{CCRIT}}.$$

7. OPTION = 7 or 10 is the Dycoss Discrete Crack Model as described in Lemmen and Meijer [2001]. The relation for the crack initiation is given as

$$\left[\frac{\max(\sigma_n, 0)}{\text{NFLS}}\right]^2 + \left[\frac{\sigma_s}{\text{SFLS} - \sin(\text{PARAM})\min(0, \sigma_n)}\right]^2 = 1.$$

8. OPTION = 9 or 11 is based on the fracture model in the cohesive material model *MAT_COHESIVE_MIXED_MODE, where the model is described in detail. Failure stresses/peak tractions NFLS and/or SFLS can be defined as functions of characteristic element length (square root of master segment area) via load curve. This option is useful to get nearly the same global responses (e.g. load-displacement curve) with coarse meshes compared to a fine mesh solution. In

general, lower peak tractions are needed for coarser meshes. See also *MAT_-138.

9. For OPTIONs 6 thru 11 of *CONTACT_AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE_TIEBREAK, one can determine the condition of the tiebreak surface via the component labeled "contact gap" in the intfor database (*DATABASE_BINARY_INTFOR). The "contact gap" actually represents a damage value ranging from 0 (tied, no damage) to 1 (released, full damage).

10. Tying in the AUTOMATIC_..._TIEBREAK contacts occurs if the slave node is within a small tolerance of the master surface *after* taking into account contact thicknesses. For MPP, the tolerance is given by

$$\text{tol} = 0.01\sqrt{2 \times \text{master segment area}}$$

for SMP, the tolerance is 0.4(slave contact thickness + master contact thickness).

11. It is recommended that the slave and master sides of tiebreak contact be defined using segment sets rather than part IDs or part set IDs. In this way, the user can be more selective in choosing which segments are to be tied and ensure that contact stresses calculated from nodal contact forces are not diluted by segments that are not actually on the actual contact surface. The user also has direct control over the contact segment normal vectors when segment sets are used. Segment normal vectors should point toward the opposing contact surface so that tension is properly distinguished from compression.

Card 4: AUTOMATIC_SURFACE_TO_SURFACE_COMPOSITE

This card 4 is mandatory for:

*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_COMPOSITE

Card 4a	1	2	3	4	5	6	7	8
Variable	TFAIL	MODEL	CIDMU	CIDETA	D			
Type	F	I	I	I	F			
Default	required	required	required	required	0.0			

VARIABLE	DESCRIPTION
TFAIL	Tensile traction σ_f required for failure.
MODEL	Model for shear response. See the equations in the Remarks for details. EQ.1: limiting shear stress depends on CIDMU in both tension and compression. See remark 3. EQ.2: limiting shear stress depends on CIDETA in tension and CIDMU in compression. See remark 4. EQ.3: limiting shear stress depends on CIDETA in both tension and compression. See remark 5.
CIDMU	Curve ID for the coefficient of friction $\mu(H)$ as a function of the Hershey number H.
CIDETA	Curve ID for the viscosity $\eta(T)$ as a function of temperature T.
D	Composite film thickness.

Remarks:

1. This contact model is designed for simulating the processing of laminated composite materials. Surfaces in contact may support shear up to the limit defined by MODEL and be in compression or in tension up to the tensile limit σ_f defined by TFAIL. After TFAIL is reached, the contact fails in both tension and shear. If the surfaces come back into contact, the bonding heals, and the contacting surfaces may support shear and tension.

2. The viscosity $\eta(T)$ is defined as a function of temperature by CIDETA. The value of the viscosity is not extrapolated if the temperature falls outside of the temperature range defined by the curve.

3. The coefficient of friction μ for MODEL = 1 is defined in terms of the Hershey number $H = \eta(T)V/(p + \sigma_f)$ where p is the contact pressure (positive in compression, and negative in tension) and V the relative velocity between the surfaces.

$$\tau \le \mu(H)(p + \sigma_f)$$

4. The coefficient of friction μ for MODEL = 2 is defined in terms of the Hershey number $H = \eta(T)V/p$. Note the definition of the Hershey number for this model differs from MODEL = 1. In compression the shear stress is limited by

$$\tau \le \mu(H)p$$

and in tension, the shear stress is limited according to

$$\tau \leq \eta(T)V/d$$

5. The shear stress for MODEL = 3 in tension and compression is limited according to

$$\tau \leq \eta(T)V/d.$$

Card 4: SINGLE_SURFACE_TIED

This card 4 is mandatory for:

*CONTACT_AUTOMATIC_SINGLE_SURFACE_TIED

Card 4	1	2	3	4	5	6	7	8
Variable	CLOSE							
Type	F							
Default	0.0							

VARIABLE	DESCRIPTION
CLOSE	Surfaces closer than CLOSE are tied. If CLOSE is left as 0.0, it is defaulted to one percent of the mesh characteristic length scale. Nodes that are above or below the surface will be tied if they are close enough to the surface.

Remarks:

This special feature is implemented to allow for the calculation of eigenvalues and eigenvectors on geometries that are connected by a contact interface using the AUTOMATIC_SINGLE_SURFACE options.

If there is significant separation between the tied surfaces, the rigid body modes will be opposed by the contact stiffness, and the calculated eigenvalues for rigid body rotations will not be zero.

Card 4: AUTOMATIC_SURFACE_TO_SURFACE_TIED_WELD

This card 4 is mandatory for:

*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_TIED_WELD

Card 4	1	2	3	4	5	6	7	8
Variable	TEMP	CLOSE						
Type	F	F						
Default	None	0.0						

VARIABLE	DESCRIPTION
TEMP	Minimum temperature required on both surfaces for tying. Once the surfaces are tied, they remain tied even if the temperature drops.
CLOSE	Surfaces closer than CLOSE are tied. If CLOSE is left as 0.0, it is defaulted to one percent of the mesh characteristic length scale. Nodes that are above or below the surface will be tied if they are close enough to the surface.

Remarks:

This special feature is implemented to allow for the simulation of welding. As regions of the surfaces are heated to the welding temperature and come into contact, the nodes are tied.

If there is significant separation between the tied surfaces, the rigid body modes will not be opposed by the contact stiffness. In other words, the offset between the surfaces is handled like the contact with OFFSET.

If the surfaces are below the welding temperature, the surfaces interact with the standard AUTOMATIC_SURFACE_TO_SURFACE options.

Card 4: CONSTRAINT_…_TO_SURFACE

This card 4 is mandatory for:

*CONTACT_CONSTRAINT_NODES_TO_SURFACE

*CONTACT_CONSTRAINT_SURFACE_TO_SURFACE

Card 4	1	2	3	4	5	6	7	8
Variable	KPF							
Type	F							
Default	0.0							

VARIABLE	DESCRIPTION
KPF	Kinematic partition factor for constraint: EQ.0.0: fully symmetric treatment. EQ.1.0: one way treatment with slave nodes constrained to master surface. Only the slave nodes are checked against contact. EQ.-1.0: one way treatment with master nodes constrained to slave surface. Only the master nodes are checked against contact.

Card 4: DRAWBEAD

This card 4 is mandatory for:

*CONTACT_DRAWBEAD

*CONTACT_DRAWBEAD_INITIALIZE

Note variables related to automatic multiple draw bead feature, including NBEAD, POINT1, POINT2, WIDTH, and EFFHGT are not applicable to *CONTACT_DRAWBEAD_INITIALIZE.

Card 4a	1	2	3	4	5	6	7	8
Variable	LCIDRF	LCIDNF	DBDTH	DFSCL	NUMINT	DBPID	ELOFF	NBEAD
Type	I	I	F	F	I	I	I	I
Default	required	none	0.0	1.0	0	0	0	none

Additional card to be included if NBEAD is defined.

Card 4b	1	2	3	4	5	6	7	8
Variable	POINT1	POINT2	WIDTH	EFFHGT				
Type	I	I	F	F				
Default	none	none	none	none				

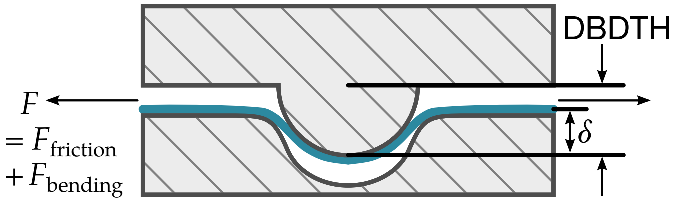


Figure 11-2. The draw bead contact model.

Initialization Card. Additional card for INITIALIZE keyword option. Card to initialize the plastic strain and thickness of elements that pass under the draw bead.

Card 4c	1	2	3	4	5	6	7	8
Variable	LCEPS	TSCALE	LCEPS2	OFFSET				
Type	I	F	I	F				
Default	required	1.0	optional	optional				

VARIABLE	DESCRIPTION
LCIDRF	If LCIDRF is positive then it defines the load curve ID giving the bending component of the restraining force, F_{bending}, per unit draw bead length as a function of displacement, δ, see Figure 11-2. This force is due to the bending and unbending of the blank as it moves through the draw bead. The total restraining force is the sum of the bending and friction components. If LCIDRF is negative, then the absolute value gives the load curve ID defining max bead force versus normalized draw bead length. The abscissa values are between zero and 1 and are the normalized draw bead length. The ordinate gives the maximum allowed draw bead, retaining force when the bead is in the fully closed position. If the draw bead is not fully closed, linear interpolation is used to compute the draw bead force.
LCIDNF	Load curve ID giving the normal force per unit draw bead length as a function of displacement, δ, see Figure 11-2. This force originates from bending the blank into the draw bead as the binder closes on the die. The normal force begins to develop when the distance between the die and binder is less than the draw bead depth. As the binder and die close on the blank this force should diminish or reach a plateau.

VARIABLE	DESCRIPTION
DBDTH	Draw bead depth, see Figure 11-2. Necessary to determine correct δ displacement from contact displacements.
DFSCL	Scale factor for load curve. Default = 1.0. This factor scales load curve ID, LCIDRF above.
NUMINT	Number of equally spaced integration points along the draw bead: EQ.0: Internally calculated based on element size of elements that interact with draw bead. This is necessary for the correct calculation of the restraining forces. More integration points may increase the accuracy since the force is applied more evenly along the bead.
DBPID	Optional part ID for the automatically generated truss elements for the draw bead display in the post-processor. If undefined LS-DYNA assigns a unique part ID.
ELOFF	Option to specify and element ID offset for the truss elements that are automatically generated for the draw bead display. If undefined LS-DYNA chooses a unique offset.
NBEAD	Number of line beads in odd integer.
POINT1	Node ID of the first node on a binder.
POINT2	Node ID of a *matching* node on the opposing binder.
WIDTH	Total bead width defining distance between inner and outer most bead walls.
EFFHGT	Effective bead height. Draw bead restraining force starts to take effect when binder gap is less than EFFHGT.
LCEPS	Load curve ID defining the plastic strain versus the parametric coordinate through the shell thickness. The parametric coordinate should be defined in the interval between -1 and 1 inclusive. The value of plastic strain at the integration point is interpolated from this load curve. If the plastic strain at an integration point exceeds the value of the load curve at the time initialization occurs, the plastic strain at the point will remain unchanged.

VARIABLE **DESCRIPTION**

TSCALE Scale factor that multiplies the shell thickness as the shell element moves under the draw bead.

LCEPS2 Optional load curve ID defining the plastic strain versus the parametric coordinate through the shell thickness, which is used after an element has traveled a distance equal to OFFSET. The parametric coordinate should be defined in the interval between -1 and 1 inclusive. The value of plastic strain at the integration point is interpolated from this load curve. If the plastic strain at an integration point exceeds the value of the load curve at the time initialization occurs, the plastic strain at the point will remain unchanged. Input parameters LCEPS2 and OFFSET provides a way to model the case where a material moves under two draw beads. In this latter case the curve would be the sum of the plastic strains generate by moving under two consecutive beads.

OFFSET If the center of an element has moved a distance equal to OFFSET, the load curve ID, LCEPS2 is used to reinitialize the plastic strain. The TSCALE scale factor is also applied.

Overview:

In the framework of this draw bead model the blank is the master part, and the male part of the draw bead is the slave. The male part of the draw bead, which moves with the punch, is input as a curve defined using a list of nodes or a part consisting of beams, as discussed below. Associated with this curve is a *region of influence* that is characterized by the DBDTH field of card 4a.

As the punch comes down and the region of influence intersects the elements on the blank, forces are applied to the blank at the points of closest approach. These forces depend on the separation distance, δ, which is geometrically defined in Figure 11-2. The draw bead force model consists of two terms:

6. There is a *resisting force*, which is a function of δ, and is defined through the load curve specified in the LCIDRF field. This force is applied in a direction opposite to velocity.

7. There is also a *normal force* pushing the male part of the draw bead away from the blank, which is specified by LCIDNF. This normal force, in turn, is used to model friction, which depends on the product of the friction coefficient and the normal force.

The curve representing the male part of the draw bead can be defined in three ways:

1. A *consecutive* list of *slave* nodes that lie along the bead.

2. A part ID of a beam that lies along the draw bead.

3. A part set ID of beams that lie along the draw bead.

For straight draw beads, only two nodes or a single beam needs to be defined, i.e., one at each end. For curved beads, many nodes or beams may be required to define the curvature of the bead geometry.

When beams are used to define the bead, with the exception of the first and last node, each node must connect with two beam elements. This requirement means that the number of slave nodes equals the number of beam elements plus one.

It is at the integration points where the contact algorithm checks for penetration. Integration points are equally spaced along the draw bead and do not depend on the nodal spacing used in the definition of the draw bead. By using the capability of tying extra nodes to rigid bodies (see *CONSTRAINED_EXTRA_NODES or *CONSTRAINED_RIGID_BODIES) the draw bead nodal points do not need to belong to the element connectivities of the die and binder. The blank makes up the master surface.

NOTE: It is highly recommended to define a BOXID around the draw bead to limit the size of the master surface considered for the draw bead. This will substantially reduce cost and memory requirements.

LS-PrePost:

While defining a contact draw bead may involve several keywords, the processed is streamlined by the "draw bead" definition feature of LS-PrePost4.0's eZ-Setup for metal forming application. See,

http://ftp.lstc.com/anonymous/outgoing/lsprepost/4.0/metalforming/

Multiple draw beads model:

Developed in conjunction with the *Ford Motor Company Research & Advanced Engineering Laboratory*, the multiple draw bead features provides a simple way to model (1) the neglected effects of the draw bead width, and (2) to attenuate the bead forces when the distance between upper and lower binders is more than the draw bead height.

1. **Draw Bead Width Correction.** As shown in Figure 11-3, it often happens that a sheet blank edge does not cross the draw bead's curve of definition but *does*

fall within its width. When the bead is modelled as a 1-dimensional (no width) curve, it is possible that a major portion of the blank would have no forces applied, while, in reality, there are still two bending radii at the inner bead wall providing about 50% of the total bead forces. The neglect of width effects leads to excessive blank edge draw-ins resulting in either loose metal in the part, or wrinkles on the draw wall or product surface.

The multiple beads feature ameliorates this particular shortcoming by replacing the single 1-dimensional bead with an equivalent set of beads distributed over the width of the physical bead. The bead force is distributed uniformly over the NBEAD sub-beads, such that the resultant force is equal to that of the original 1-dimensionsal bead. Note that NBEAD *must* be an odd integer.

Figure 11-4 schematically represents the NBEAD = 3 case for which two additional line beads are automatically generated. The forces specified by the load curve, LCIDRF, will be evenly distributed over the 3 beads. In Figure 11-5, bead forces are recovered from the ASCII rcforc files for both cases of NBEAD = 1 and 3, indicating the total force applied (shown on the left) on one single bead is distributed evenly among the three automatically generated beads for the case of NBEAD = 3.

The stress distribution is also more realistic with the multiple beads. In a channel draw (half model) as shown in Figure 11-9, no significant changes in mean stress values are found between NBEAD = 3 and one single line bead. In fact, the compressive stresses are more realistically and evenly distributed around the bead region, with stresses in NBEAD = 3 about 1/3 of those in one line bead.

2. **Lower Binder Gap Correction.** As originally implemented, the draw bead contact model applies the draw bead forces, as specified in the load curve, when the upper binder reaches the blank, regardless of the lower binder’s position. If the lower binder is not in contact with the blank, LS-DYNA still applies draw bead forces, even though it is unphysical to do so. The EFFHGT, POINT1 and POINT2 fields together provide a simple model to avoid these unphysical forces. The POINT1 and POINT2 fields are taken as nodes on the opposing binders. The draw bead contact is disabled when the Euclidean distance between POINT1 and POINT2 is greater than EFFHGT; consequently, the two nodes *must* be chosen so they converge to a single point as the draw bead closes.

   As shown in Figure 11-6, a simple model was built to verify the effectiveness of the variable EFFHGT. The upper binder is pushed down to close with the lower binder while a strip of sheet blank is being pulled in the direction indicated. The distance between the binders is 12mm initially, as shown in Figure 11-7, and the closing gap and pulling force in *x* were recovered throughout the simulation. With the EFFHGT set at 8mm, the pulling force history indicates the bead forces starting to take effect after the upper binder has traveled for 4mm, Figure 11-8, as expected.

Revision information:

The NBEAD feature is available in LS-DYNA R6 Revision 69556 and later releases, with important updates in Revision 79270.

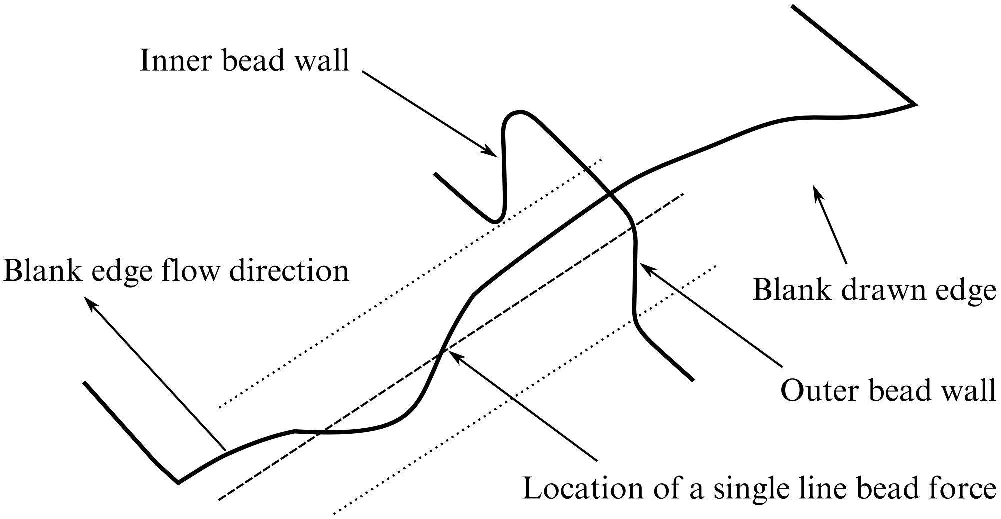

Inner bead wall
Blank edge flow direction
Blank drawn edge
Outer bead wall
Location of a single line bead force

Figure 11-3. A possible scenario of sheet blank edge draw-in condition.

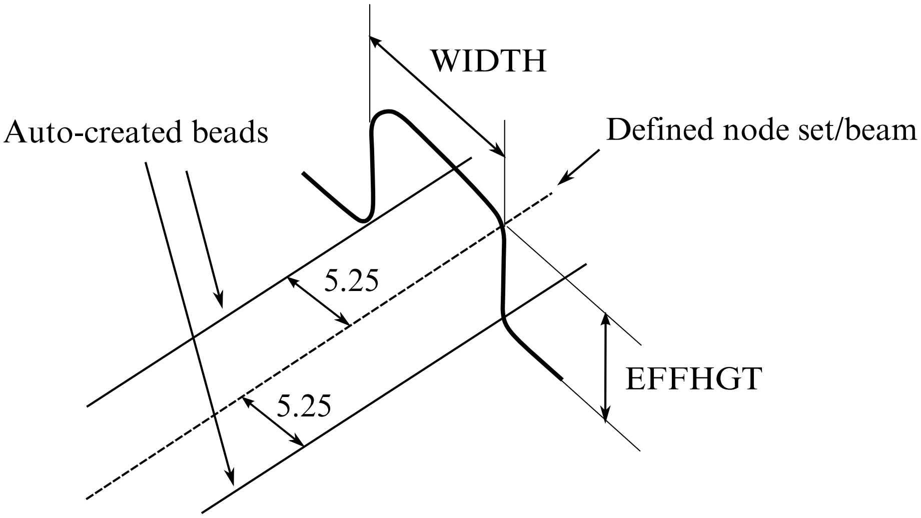


Figure 11-4. Definition of multiple draw beads.

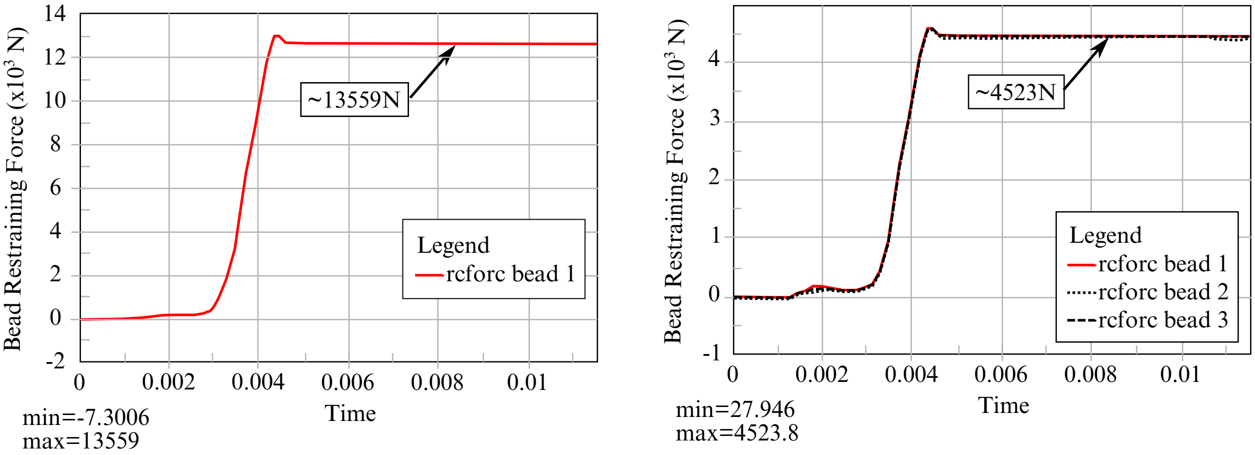


Figure 11-5. Bead force verification between NBEAD = 1 (left) and 3.

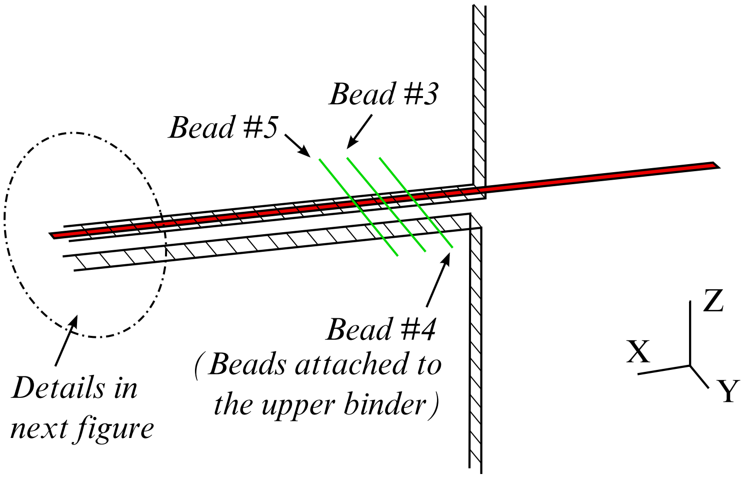


Figure 11-6. A verification model for the variable EFFHGT.

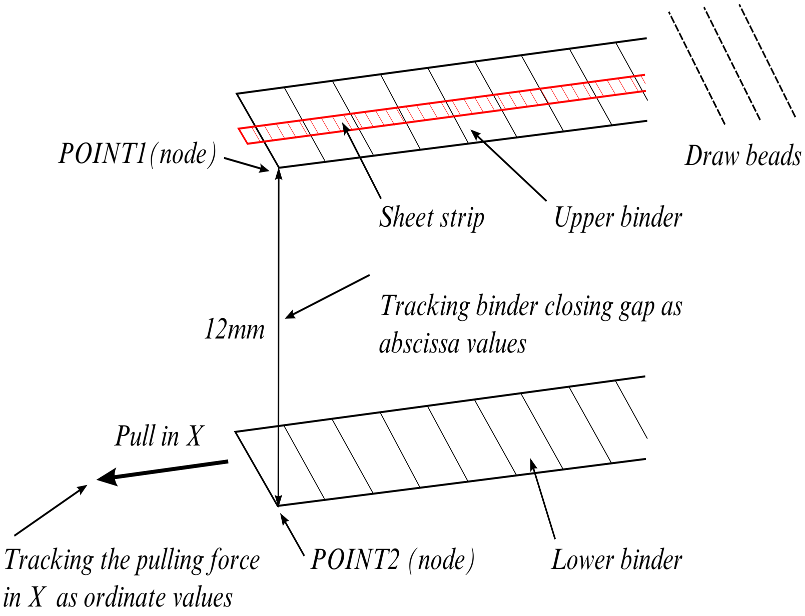

POINT1(node)
Draw beads
Sheet strip
Upper binder
12mm
Tracking binder closing gap as abscissa values
Pull in X
Tracking the pulling force in X as ordinate values
POINT2 (node)
Lower binder

Figure 11-7. Tracking the closing gap and pulling distance.

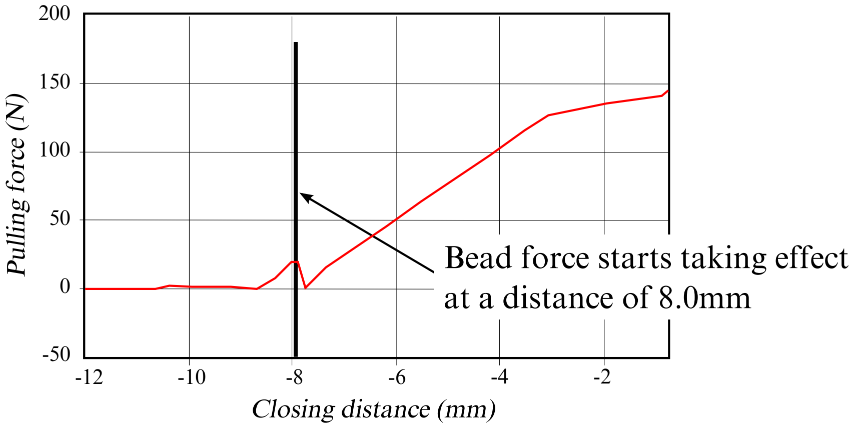
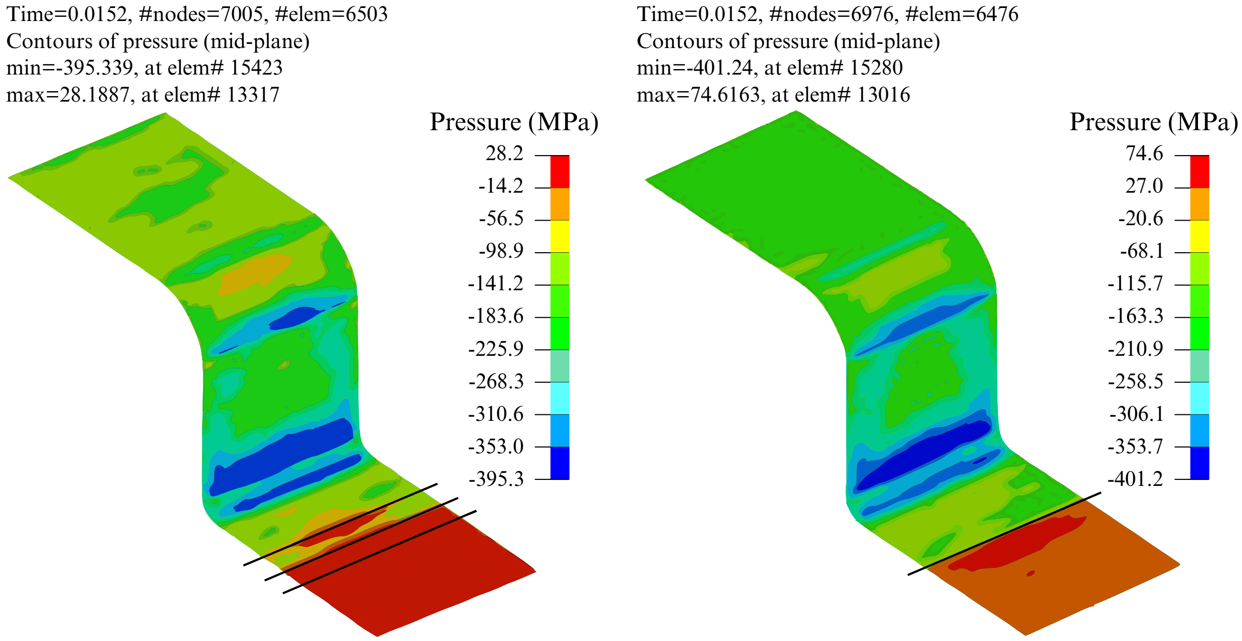


Figure 11-8. Pulling force (NODFOR) vs. closure distance.



Mean stresses of a channel draw
(NBEAD=3)

Mean stresses of a channel draw
on one line bead

Figure 11-9. Mean stress comparison between NBEAD = 3 and 1.

Card 4: ERODING_..._SURFACE

This card 4 is mandatory for:

*CONTACT_ERODING_NODES_TO_SURFACE

*CONTACT_ERODING_SINGLE_SURFACE

*CONTACT_ERODING_SURFACE_TO_SURFACE

Card 4	1	2	3	4	5	6	7	8
Variable	ISYM	EROSOP	IADJ					
Type	I	I	I					
Default	0	0	0					

VARIABLE	DESCRIPTION
ISYM	Symmetry plane option: EQ.0: off, EQ.1: do not include faces with normal boundary constraints (e.g., segments of brick elements on a symmetry plane). This option is important to retain the correct boundary conditions in the model with symmetry.
EROSOP	Erosion/Interior node option: (reset to 1 internally) EQ.0: only exterior boundary information is saved, EQ.1: storage is allocated so that eroding contact can occur. Otherwise, no contact is assumed after erosion of the corresponding element.
IADJ	Adjacent material treatment for solid elements: (reset to 1 internally) EQ.0: solid element faces are included only for free boundaries, EQ.1: solid element faces are included if they are on the boundary of the material subset. This option also allows the erosion within a body and the subsequent treatment of contact.

Remarks:

Eroding contact may control the timestep (see ECDT in *CONTROL_CONTACT). For ERODING_NODES_TO_SURFACE, define the slave side using a node set, not a part ID or part set ID.

Use of an ERODING contact automatically invokes a negative volume failure criterion for all solid elements in the model, except as overridden by PSFAIL in *CONTROL_SOLID. Use of PSFAIL will limit the negative volume failure criterion to a set of solid parts. A negative volume failure criterion circumvents an error termination due to negative volume by deleting solid elements that develop negative volume.

Contact friction is not considered by SMP LS-DYNA for *CONTACT_ERODING_-NODES_TO_SURFACE and *CONTACT_ERODING_SURFACE_TO_SURFACE unless SOFT is set to 2 on Optional Card A. MPP LS-DYNA has no such exclusion for contact friction.

Values of EROSOP = 0 and IADJ = 0 are not supported, and both are reset to 1 internally.

Card 4: SURFACE_INTERFERENCE

This card 4 is mandatory for:

*CONTACT_NODES_TO_SURFACE_INTERFERENCE

*CONTACT_ONE_WAY_SURFACE_TO_SURFACE_INTERFERENCE

*CONTACT_SURFACE_TO_SURFACE_INTERFERENCE

Purpose: This contact option provides a means of modeling parts which are shrink fitted together and are, therefore, prestressed in the initial configuration. This option turns off the nodal interpenetration checks (which changes the geometry by moving the nodes to eliminate the interpenetration) at the start of the simulation and allows the contact forces to develop to remove the interpenetrations. The load curves defined in this section scale the interface stiffness constants such that the stiffness can increase slowly from zero to a final value with effect that the interface forces also increase gradually to remove the overlaps.

Card 4	1	2	3	4	5	6	7	8
Variable	LCID1	LCID2						
Type	I	I						
Default	0	0						

VARIABLE	DESCRIPTION
LCID1	Load curve ID which scales the interface stiffness during dynamic relaxation. This curve must originate at (0, 0) at time = 0 and gradually increase.
LCID2	Load curve ID which scales the interface stiffness during the transient calculation. This curve generally has a constant value of unity for the duration of the calculation if LCID1 is defined. If LCID1 = 0, this curve must originate at (0, 0) at time = 0 and gradually increase to a constant value.

Remarks:

1. Shell thickness offsets are taken into account for deformable shell elements.

2. The check to fix initial penetrations is skipped.

3. Automatic orientation of shell elements is skipped.

4. Furthermore, segment orientation for shell elements and interpenetration checks *are skipped*.

Therefore, it is necessary in the problem setup to ensure that all contact segments which belong to shell elements are properly oriented, i.e., the outward normal vector of the segment based on the right hand rule relative to the segment numbering, must point to the opposing contact surface; consequently, automatic contact generation should be avoided for parts composed of shell elements unless automatic generation is used on the slave side of a nodes to surface interface.

Card 4: RIGID_TO_RIGID

This card 4 is mandatory for:

*CONTACT_RIGID_NODES_TO_RIGID_BODY

*CONTACT_RIGID_BODY_ONE_WAY_TO_RIGID_BODY

*CONTACT_RIGID_BODY_TWO_WAY_TO_RIGID_BODY

Card 4	1	2	3	4	5	6	7	8
Variable	LCID	FCM	US		LCDC	DSF	UNLCID	
Type	I	I	F		I	F	I	
Default	required	required	LCID		optional	0.0	optional	

VARIABLE	DESCRIPTION
LCID	Load curve ID giving force versus penetration behavior for RIGID_contact. See also the definition of FCM below.

FCM — Force calculation method for RIGID_contact:

EQ.1: Load curve gives total normal force on surface versus maximum penetration of any node (RIGID_BODY_-ONE_WAY only).

EQ.2: Load curve gives normal force on each node versus penetration of node through the surface (all RIGID_contact types).

EQ.3: Load curve gives normal pressure versus penetration of node through the surface (RIGID_BODY_TWO_WAY and RIGID_BODY_ONE_WAY only).

EQ.4: Load curve gives total normal force versus maximum soft penetration. In this case the force will be followed based on the original penetration point. (RIGID_-BODY_ONE_WAY only).

US — Unloading stiffness for RIGID_contact. The default is to unload along the loading curve. This should be equal to or greater than the maximum slope used in the loading curve.

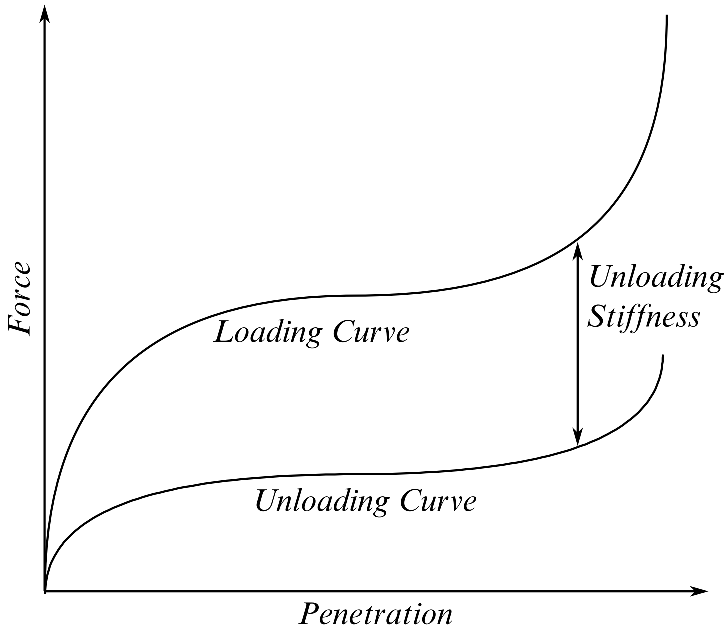


Figure 11-10. Behavior if an unloading curve is defined

VARIABLE	DESCRIPTION
LCDC	Load curve ID giving damping coefficient (DC) versus penetration velocity. The damping force FD is then: FD = DSF × DC × velocity.
DSF	Damping scaling factor.
UNLCID	Optional load curve ID giving force versus penetration behavior for RIGID_BODY_ONE_WAY contact. This option requires the definition of the unloading stiffness, US. See Figure 11-10.

Card 4: TIEBREAK_NODES

This card 4 is mandatory for:

*CONTACT_TIEBREAK_NODES_TO_SURFACE and

*CONTACT_TIEBREAK_NODES_ONLY

Card 4	1	2	3	4	5	6	7	8
Variable	NFLF	SFLF	NEN	MES				
Type	F	F	F	F				
Default	required	required	2.	2.				

VARIABLE	DESCRIPTION
NFLF	Normal failure force. Only tensile failure, i.e., tensile normal forces, will be considered in the failure criterion.
SFLF	Shear failure force
NEN	Exponent for normal force
MES	Exponent for shear force. Failure criterion: $$\left(\frac{\lvert f_n \rvert}{\text{NFLF}}\right)^{\text{NEN}} + \left(\frac{\lvert f_s \rvert}{\text{SFLF}}\right)^{\text{MES}} \geq 1.$$ Failure is assumed if the left side is larger than 1. f_n and f_s are the normal and shear interface force.

Remarks:

These attributes can be overridden node by node on the *SET_NODE_option cards.

Both NFLF and SFLF must be defined. If failure in only tension or shear is required then set the other failure force to a large value (1E+10).

After failure, contact_tiebreak_nodes_to_surface behaves as a nodes-to-surface contact with no thickness offsets (no interface tension possible) whereas the contact_tiebreak_nodes_only stops acting altogether. Prior to failure, the two contact types behave identically.

Card 4: TIEBREAK_SURFACE

This card 4 is mandatory for:

*CONTACT_TIEBREAK_SURFACE_TO_SURFACE and

*CONTACT_TIEBREAK_SURFACE_TO_SURFACE_ONLY

Card 4	1	2	3	4	5	6	7	8
Variable	NFLS	SFLS	TBLCID	THKOFF				
Type	F	F	I	I				
Default	required	required	0	0				

VARIABLE	DESCRIPTION
NFLS	Tensile failure stress. See remark below.
SFLS	Shear failure stress. Failure criterion $$\left(\frac{\lvert\sigma_n\rvert}{\mathrm{NFLS}}\right)^2+\left(\frac{\lvert\sigma_s\rvert}{\mathrm{SFLS}}\right)^2\geq 1.$$
TBLCID	Optional load curve number defining the resisting tensile stress versus gap opening in the normal direction for the post failure response. This option applies only to SMP and can be used to model adhesives.
THKOFF	Thickness offsets are considered if THKOFF = 1. If shell offsets are included in the meshed geometry, this option is highly recommended since segment orientation can be arbitrary and the contact surfaces can be disjoint. This option is *not* available in the MPP version of LS-DYNA. It works by substituting *CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_TIEBREAK (OPTION = 2 if TBLCID is not specified; OPTION = 5 if TBLCID is specified).

Remarks:

The failure attributes can be overridden segment by segment on the *SET_SEGMENT or *SET_SHELL_option cards for the slave surface as A1 and A2. These variables do not apply to the master surface. Both NFLS and SFLS must be defined. If failure in only tension or shear is required then set the other failure stress to a large value (1E+10). When used with shells, contact segment normals are used to establish the tension direction (as opposed to compression). Compressive stress does not contribute to the failure equation.

After failure, *CONTACT_TIEBREAK_SURFACE_TO_SURFACE behaves as a surface-to-surface contact with no thickness offsets.

After failure, *CONTACT_TIEBREAK_SURFACE_TO_SURFACE_ONLY stops acting altogether. Until failure, it ties the slave nodes to the master nodes.

Card 4: CONTRACTION_JOINT

This card 4 is mandatory for:

*CONTACT_SURFACE_TO_SURFACE_CONTRACTION_JOINT

Purpose: This contact option turns on the contraction joint model designed to simulate the effects of sinusoidal joint surfaces (shear keys) in the contraction joints of arch dams and other concrete structures. The sinusoidal functions for the shear keys are defined according to the following three methods [Solberg and Noble 2002]:

Method 1:

$$\hat{g} = g - A\{1 - \cos[B(s_2 - s_1)]\}$$

Method 2:

$$\hat{g} = g - 2A\left|\sin\left[\frac{B(s_2 - s_1)}{2}\right]\right|$$

Method 3: (default)

$$\hat{g} = g - A\cos(Bs_2) + A\cos(Bs_1)$$

Where g is a gap function for contact surface, $\hat{g}$ is gap function for the joint surface. A is key amplitude parameter, and B is key frequency parameter. s_1 and s_2 are referential surfaces:

$$s_1 = \mathbf{X}_{\mathrm{surface1}} \cdot \mathbf{T}_{\mathrm{key}}$$

$$s_2 = \mathbf{X}_{\mathrm{surface2}} \cdot \mathbf{T}_{\mathrm{key}}$$

$$\mathbf{T}_{\mathrm{key}} = \mathbf{T}_{\mathrm{slide}} \times \mathbf{n}$$

Where $\mathbf{T}_{\mathrm{slide}}$ is the free sliding direction of the keys, $\mathbf{n}$ is the surface normal in reference.

Card 4	1	2	3	4	5	6	7	8
Variable	MTCJ	ALPHA	BETA	TSVX	TSVY	TSVZ		
Type	I	F	F	F	F	F		
Default	0	0.0	0.0	0.0	0.0	0.0		

VARIABLE	DESCRIPTION
MTCJ	The method option for the gap function, $\hat{g}$
ALPHA	Key amplitude parameter A
BETA	Key frequency parameter B
TSVX	x component of the free sliding direction $\mathbf{T}_{\text{slide}}$
TSVY	y component of the free sliding direction $\mathbf{T}_{\text{slide}}$
TSVZ	z component of the free sliding direction $\mathbf{T}_{\text{slide}}$

THERMAL:

This card is mandatory for the *THERMAL* option, i.e.:

*CONTACT_…_THERMAL_…

Reminder: If Card 4 is required, then it must go before this thermal card. (Card 4 is required for certain contact types - see earlier in this section for the list, later in this section for details of Card 4.)

Thermal Card I.

Card 1	1	2	3	4	5	6	7	8
Variable	K	FRAD	H0	LMIN	LMAX	FTOSLV	BC_FLG	ALGO
Type	F	F	F	F	F	F	I	I
Default	none	none	none	none	none	0.5	0	0

VARIABLE	DESCRIPTION

K Thermal conductivity of fluid between the contact surfaces. If a gap with a thickness l_{gap} exists between the contact surfaces, then the conductance due to thermal conductivity between the contact surfaces is

$$h_{\text{cond}} = \frac{K}{l_{\text{gap}}}$$

Note that LS- DYNA calculates l_{gap} based on deformation

FRAD Radiation factor between the contact surfaces.

$$f_{\text{rad}} = \frac{\sigma}{\frac{1}{\varepsilon_1} + \frac{1}{\varepsilon_2} - 1}$$

Where,

σ = Stefan-Boltman constant

ε_1 = emissivity of master surface

ε_2 = emissivity of slave surface

LS-DYNA calculates a radiant heat transfer conductance

$$h_{\text{rad}} = f_{\text{rad}}(T_m + T_s)(T_m^2 + T_s^2)$$

VARIABLE **DESCRIPTION**

H0

Heat transfer conductance for closed gaps. Use this heat transfer conductance for gaps in the range

$$0 \le l_{\mathrm{gap}} \le l_{\mathrm{min}}$$

LMIN

Minimum gap, l_{min}, use the heat transfer conductance defined (H0) for gap thicknesses less than this value.

If $l_{\mathrm{min}} < 0$, then $-l_{\mathrm{min}}$ is a load curve number defining l_{min} as a function time.

LMAX

No thermal contact if gap is greater than this value (l_{max}).

FTOSLV

Fraction, f, of sliding friction energy partitioned to the slave surface. Energy partitioned to the master surface is $(1-f)$.

EQ.0: Default set to 0.5: The sliding friction energy is partitioned 50% - 50% to the slave and master surfaces in contact.

$$f = \frac{1}{1 + \frac{\sqrt{(\rho C_p k)_{\text{master side material}}}}{\sqrt{(\rho C_p k)_{\text{slave side material}}}}}.$$

BC_FLAG

Thermal boundary condition flag

EQ.0: thermal boundary conditions are on when parts are in contact

EQ.1: thermal boundary conditions are off when parts are in contact

ALGO

Contact algorithm type.

EQ.0: two way contact, both surfaces change temperature due to contact

EQ.1: one way contact, master surface does not change temperature due to contact. Slave surface does change temperature.

Remarks:

Note that LS- DYNA calculates l_{gap} based on deformation

In summary:

$$h = \begin{cases} h_0 & 0 \leq l_{\text{gap}} \leq l_{\text{min}} \\ h_{\text{cond}} + h_{\text{rad}} & l_{\text{min}} < l_{\text{gap}} \leq l_{\text{max}} \\ 0 & l_{\text{gap}} > l_{\text{max}} \end{cases}$$

THERMAL FRICTION:

This card is required if the FRICTION suffix is added to THERMAL.

*CONTACT_…_THERMAL_FRICTION_…

The blank (or work piece) must be defined as the slave surface in a metal forming model.

Purpose:

1. Used to define the mechanical static and dynamic friction coefficients as a function of temperature.

2. Used to define the thermal contact conductance as a function of temperature and pressure.

Card 1	1	2	3	4	5	6	7	8
Variable	LCFST	LCFDT	FORMULA	A	B	C	D	LCH
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0.	0	0	0

User Subroutine Cards. Additional cards for when FORMULA is a negative number. Use as many cards as necessary to set |FORMULA| number of parameters.

Card 2	1	2	3	4	5	6	7	8
Variable	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8
Type	F	F	F	F	F	F	F	F
Default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

VARIABLE	DESCRIPTION
LCFST	Load curve number for static coefficient of friction as a function of temperature. The load curve value multiplies the coefficient value FS.

VARIABLE	DESCRIPTION
LCFDT	Load curve number for dynamic coefficient of friction as a function of temperature. The load curve value multiplies the coefficient value FD.

FORMULA

Formula that defines the contact heat conductance as a function of temperature and pressure.

EQ.1: $h(P)$ is defined by load curve A, which contains data for contact conductance as a function of pressure.

EQ.2: $h(P)$ is given by the following where A, B, C and D although defined by load curves are typically constants for use in this formula. The load curves are to given as functions of temperature.

$$h(P) = a + bP + cP^2 + dP^3$$

EQ.3: $h(P)$ is given by the following formula from [Shvets and Dyban 1964].

$$h(P) = \frac{\pi k_{\text{gas}}}{4\lambda}\left[1. + 85\left(\frac{P}{\sigma}\right)^{0.8}\right] = \frac{a}{b}\left[1. + 85\left(\frac{P}{c}\right)^{0.8}\right]$$

where,

a: is evaluated from the load curve, A, for the thermal conductivity, k_{gas}, of the gas in the gap as a function of temperature.

b: is evaluated from the load curve, B, for the parameter grouping $\pi/4\lambda$. Therefore, this load curve should be set to a constant value. λ is the surface roughness.

c: is evaluated from the load curve, C , which specifies a stress metric for deformation (e.g., yield) as a function of temperature.

EQ.4: $h(P)$ is given by the following formula from [Li and Sellars 1996].

$$h(P) = a\left[1 - \exp\left(-b\frac{P}{c}\right)\right]^d$$

where,

a: is evaluated from the load curve, A, which defines a load curve as a function of temperature.

b: is evaluated from the load curve, B, which defines a load curve as a function of temperature.

VARIABLE	DESCRIPTION
	c: is evaluated from the load curve, C, which defines a stress metric for deformation (e.g., yield) as a function of temperature. d: is evaluated from the load curve D, which is a function of temperature. EQ.5: $h(\text{gap})$ is defined by load curve A, which contains data for contact conductance as a function of interface gap. LT.0: This is equivalent to defining the keyword *USER_INTERFACE_CONDUCTIVITY and the user subroutine usrhcon will be called for this contact interface for defining the contact heat transfer coefficient.
A	Load curve number for the a coefficient used in the formula.
B	Load curve number for the b coefficient used in the formula.
C	Load curve number for the c coefficient used in the formula.
D	Load curve number for the d coefficient used in the formula.
LCH	Load curve number for h. This parameter can refer to a curve ID (see *DEFINE_CURVE) or a function ID (see *DEFINE_FUNCTION). When LCH is a curve ID (and a function ID) it is interpreted as follows: GT.0: the heat transfer coefficient is defined as a function of time, t, by a curve consisting of $(t, h(t))$ data pairs. LT.0: the heat transfer coefficient is defined as a function of temperature, T, by a curve consisting of $(T, h(T))$ data pairs. When the reference is to a function it is prototyped as follows $h = h(t, T_{\text{avg}}, T_{\text{slv}}, T_{\text{msr}}, P, g)$ where: t = solution time T_{avg} = average interface temperature T_{slv} = slave segment temperature T_{msr} = master segment temperature P = interface pressure g = gap distance between master and slave segment

ORTHO FRICTION:

Additional cards for the ORTHO_FRICTION keyword option:

*CONTACT_…_ORTHO_FRICTION_…

Card 1	1	2	3	4	5	6	7	8
Variable	FS1_S	FD1_S	DC1_S	VC1_S	LC1_S	OACS_S	LCFS	LCPS
Type	F	F	F	F	I	I	I	I
Default	0.	0.	0.	0.	0	0	0	0

Card 2	1	2	3	4	5	6	7	8
Variable	FS2_S	FD2_S	DC2_S	VC2_S	LC2_S			
Type	F	F	F	F	I			
Default	0.	0.	0.	0.	0			

Card 3	1	2	3	4	5	6	7	8
Variable	FS1_M	FD1_M	DC1_M	VC1_M	LC1_M	OACS_M	LCFM	LCPM
Type	F	F	F	F	I	I	I	I
Default	0.	0.	0.	0.	0	0	0	0

Card 4	1	2	3	4	5	6	7	8
Variable	FS2_M	FD2_M	DC2_M	VC2_M	LC2_M			
Type	F	F	F	F	I			
Default	0.	0.	0.	0.	0			

VARIABLE **DESCRIPTION**

FS*n*_S or M

Static coefficient of friction in the local *n* orthotropic direction for the slave (S) or master (M) surface. The frictional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact,

$$\mu_c = \mathrm{FD} + (\mathrm{FS} - \mathrm{FD})e^{-\mathrm{DC}|v_{\mathrm{rel}}|}$$

where the direction and surface are left off for clarity. The ORTHO_FRICTION option applies to contact types:

AUTOMATIC_SURFACE_TO_SUFACE,

AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE,

when they are defined by segment sets. (See *DEFINE_FRICTION_ORIENTATION for remarks regarding use of *CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ORTHO_FRICTION.) Each segment in the set requires the specification of an offset angle in degrees from the 1-2 side which locates the 1 direction. The offset angle is input as the first attribute of the segment in *SET_SEGMENT. The transverse direction, 2, is in the plane of the segment and is perpendicular to the 1 direction.

FDI_S or M

Dynamic coefficient of friction in the local *n* orthotropic direction.

DC*n*_S or M

Exponential decay coefficient for the local *n* direction.

VC*n*_S or M

Coefficient for viscous friction in the local *n* direction. See the description for VC for mandatory Card 2 above.

VARIABLE **DESCRIPTION**

LC*n*_S or M

The table ID of a two dimensional table, see *DEFINE_TABLE or *DEFINE_TABLE_2D, giving the friction coefficient in the local n direction as a function of the relative velocity and interface pressure. In this case, each curve in the table definition defines the coefficient of friction versus the interface pressure corresponding to a particular value of the relative velocity.

OACS_S or M

If the default value, 0, is active, the frictional forces acting on a node sliding on a segment are based on the local directions of the segment. If OACS is set to unity, 1, the frictional forces acting on a node sliding on a segment are based on the local directions of the sliding node. No matter what the setting for OACS, the_S coefficients are always used for slave nodes and the_M coefficients for master nodes.

LCFS or M

Optional load curve that gives the coefficient of friction as a function of the direction of relative motion, as measured in degrees from the first orthotropic direction. If this load curve is specified, the other parameters (FS, FD, DC, VC, LC) are ignored. This is currently only supported in the MPP version.

LCPS or M

Optional load curve that gives a scale factor for the friction coefficient as a function of interface pressure. This is only used if LCFS (or M) is defined.

Optional Card A:

> **Reminder:** If Card 4 is required, then it must go before this optional card. (Card 4 is required for certain contact types - see earlier in this section for the list.)

Optional Card A.

Optional	1	2	3	4	5	6	7	8
Variable	SOFT	SOFSCL	LCIDAB	MAXPAR	SBOPT	DEPTH	BSORT	FRCFRQ
Type	I	F	I	F	F	I	I	I
Default	0	.1	0	1.025	0.	2	10-100	1
Remarks			type a13					

VARIABLE	DESCRIPTION

SOFT

Soft constraint option:

EQ.0: penalty formulation,

EQ.1: soft constraint formulation,

EQ.2: segment-based contact.

EQ.4: constraint approach for FORMING contact option.

EQ.6: special contact algorithm to handle sheet blank edge (deformable) to gage pin (rigid shell) contact during implicit gravity loading, applies to *CONTACT_FORMING_NODES_TO_SURFACE only. See more details in About SOFT = 6.

The soft constraint may be necessary if the material constants of the elements which make up the surfaces in contact have a wide variation in the elastic bulk moduli. In the soft constraint option, the interface stiffness is based on the nodal mass and the global time step size. This method of computing the interface stiffness will typically give much higher stiffness value than would be obtained by using the bulk modulus; therefore, this method the preferred approach when soft foam materials interact with metals. See the remark below for the segment-based penalty formulation.

VARIABLE **DESCRIPTION**

SOFSCL

Scale factor for constraint forces of soft constraint option (default=.10). Values greater than .5 for single surface contact and 1.0 for a one-way treatment are inadmissible.

LCIDAB

Load curve ID defining airbag thickness as a function of time for type a13 contact (*CONTACT_AIRBAG_SINGLE_SURFACE).

MAXPAR

Maximum parametric coordinate in segment search (values between 1.025 and 1.20 are recommended). This variable applies only to SMP; for MPP, see PARMAX. Larger values can increase cost. If zero, the default is set to 1.025 for most contact options. Other defaults are:

EQ.1.006: SPOTWELD,

EQ.1.006: TIED_SHELL_…_CONSTRAINED_OFFSET,

EQ.1.006: TIED_SHELL_…_OFFSET,

EQ.1.006: TIED_SHELL_…_:BEAM_OFFSET,

EQ.1.100: AUTOMATIC_GENERAL

This factor allows an increase in the size of the segments which may be useful at sharp corners. For the SPOTWELD and …_ OFFSET options larger values can sometimes lead to numerical instabilities; however, a larger value is sometimes necessary to ensure that all nodes of interest are tied.

SBOPT

Segment-based contact options (SOFT = 2).

EQ.0: defaults to 2.

EQ.1: pinball edge-edge contact (not recommended)

EQ.2: assume planer segments (default)

EQ.3: warped segment checking

EQ.4: sliding option

EQ.5: do options 3 and 4

VARIABLE **DESCRIPTION**

DEPTH

Search depth in automatic contact, check for nodal penetration through the closest contact segments. Value of 1 (one segment) is sufficiently accurate for most crash applications and is much less expensive. LS-DYNA for improved accuracy sets this value to 2 (two segments), which is default when set to zero, default search depth for *CONTACT_AUTOMATIC_GENERAL is 3.

LT.0: |DEPTH| is the load curve ID defining searching depth versus time. (not available when SOFT = 2)

See remarks below for segment-based contact (SOFT = 2)

options controlled by DEPTH.

BSORT

Number of cycles between bucket sorts. Values of 25 and 100 are recommended for contact types 4 and 13 (SINGLE_SURFACE), respectively. Values of 10-15 are okay for the surface to surface and node to surface contact. If zero, LS-DYNA determines the interval. BSORT applies only to SMP (see BCKT on MPP Card 1 for MPP) except in the case of SOFT = 2 or for Mortar contact (option MORTAR on the CONTACT card), in which case BSORT applies to both SMP and MPP. For Mortar contact the default is the value associated with NSBCS on *CONTROL_CONTACT.

LT.0: |BSORT| load curve ID defining bucket sorting frequency versus time.

FRCFRQ

Number of cycles between contact force updates for penalty contact formulations. This option can provide a significant speed-up of the contact treatment. If used, values exceeding 3 or 4 are dangerous. Considerable care must be exercised when using this option, as this option assumes that contact does not change FRCFRG cycles.

EQ.0: FRCFRG is set to 1 and force calculations are performed each cycle-strongly recommended.

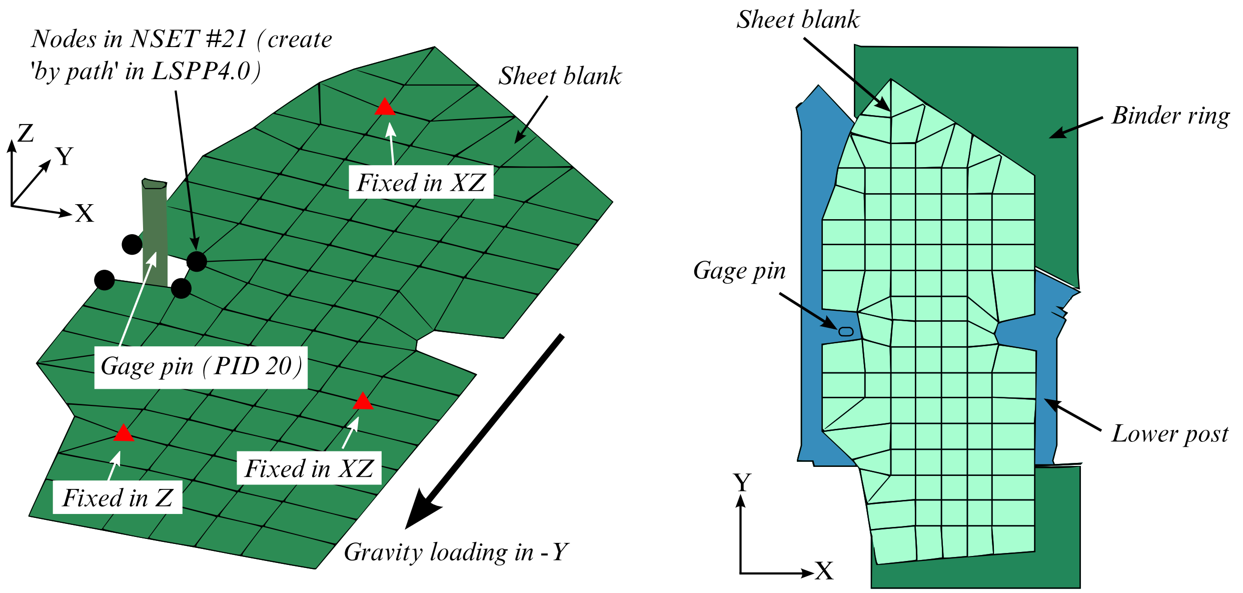


Figure 11-11. Illustrative/test model for SOFT = 6 (left) and initial blank position.

General remarks:

Setting SOFT = 1 or 2 on optional contact card A will cause the contact stiffness to be determined based on stability considerations, taking into account the time step and nodal masses. This approach is generally more effective for contact between materials of dissimilar stiffness or dissimilar mesh densities.

About SOFT = 2:

SOFT = 2 is for general shell and solid element contact. This option is available for SURFACE_TO_SURFACE, ONE_WAY_SURFACE_TO_SURFACE, and SINGLE_SURFACE options including AUTOMATIC, ERODING, and AIRBAG contact. When the AUTOMATIC option is used, orientation of shell segment normals is automatic. When the AUTOMATIC option is not used, the segment or element orientations are used as input. The segment-based penalty formulation contact algorithm checks for segments vs. segment penetration rather than node vs. segment. After penetrating segments are found, an automatic judgment is made as to which is the master segment, and penalty forces are applied normal to that segment. The user may override this automatic judgment by using the ONE_WAY options in which case the master segment normals are used as input by the user. All parameters on the first three cards are active except for VC, and VSF. On optional card A, some parameters have different meanings than they do for the default contact.

For SOFT = 2, the SBOPT parameter on optional card A controls several options. Setting DEPTH = 1 for pinball edge-to-edge checking is not recommended and is included only for back compatibility. For edge-to-edge checking setting DEPTH = 5 is recommended instead (see below). The warped segment option more accurately checks for penetration of warped surfaces. The sliding option uses neighbor segment information to improve sliding behavior. It is primarily useful for preventing segments from incorrectly catching nodes on a sliding surface.

For SOFT = 2, the DEPTH parameter controls several additional options for segment based contact.

1. **DEPTH = 2 (former default; not recommended).** surface penetrations measured at nodes are checked.

2. **DEPTH = 3 (current default).** Surface penetration is also be measured at the edge. This option is more accurate than DEPTH=2, and is good for a wide variety of simulations, but does not check for edge-to-edge penetration.

3. **DEPTH = 5.** Both surface penetrations and edge-to-edge penetration is checked.

4. **DEPTH = 13.** The penetration checking is the same as for DEPTH=3, but the code has been tuned to better conserve energy.

5. **DEPTH = 23 or 33.** The penetration checking is similar to DEPTH=3, but new methods are used to try to improve robustness.

6. **DEPTH = 25 or 35.** The penetration checking is similar to DEPTH=5 but use new methods to try to improve robustness.

7. **DEPTH = 45.** The splitting pinball method [Belytschko and Yeh, 1993] is used. This method is more accurate at the cost of more CPU time, and is recommended when modeling complex contacts between parts comprised of shells. It does not apply to solid or thick shell parts but such parts can be coated with null shells as a means of making DEPTH=45 available.

8. **DEPTH = 1 or 4.** The airbag contact has two additional options, DEPTH=1 and 4. DEPTH=4 activates additional airbag logic that uses neighbor segment information when judging if contact is between interior or exterior airbag surfaces. This option is not recommended and is maintained only for backward compatibility. Setting DEPTH=1 suppresses all airbag logic.

For SOFT = 2 contact, only the ISYM, I2D3D, SLDTHK, and SLDSTF parameters are active on optional card B. Also, the negative MAXPAR option is now incorporated into the DTSTIF option on optional card C. Data that uses the negative MAXPAR option will continue to run correctly.

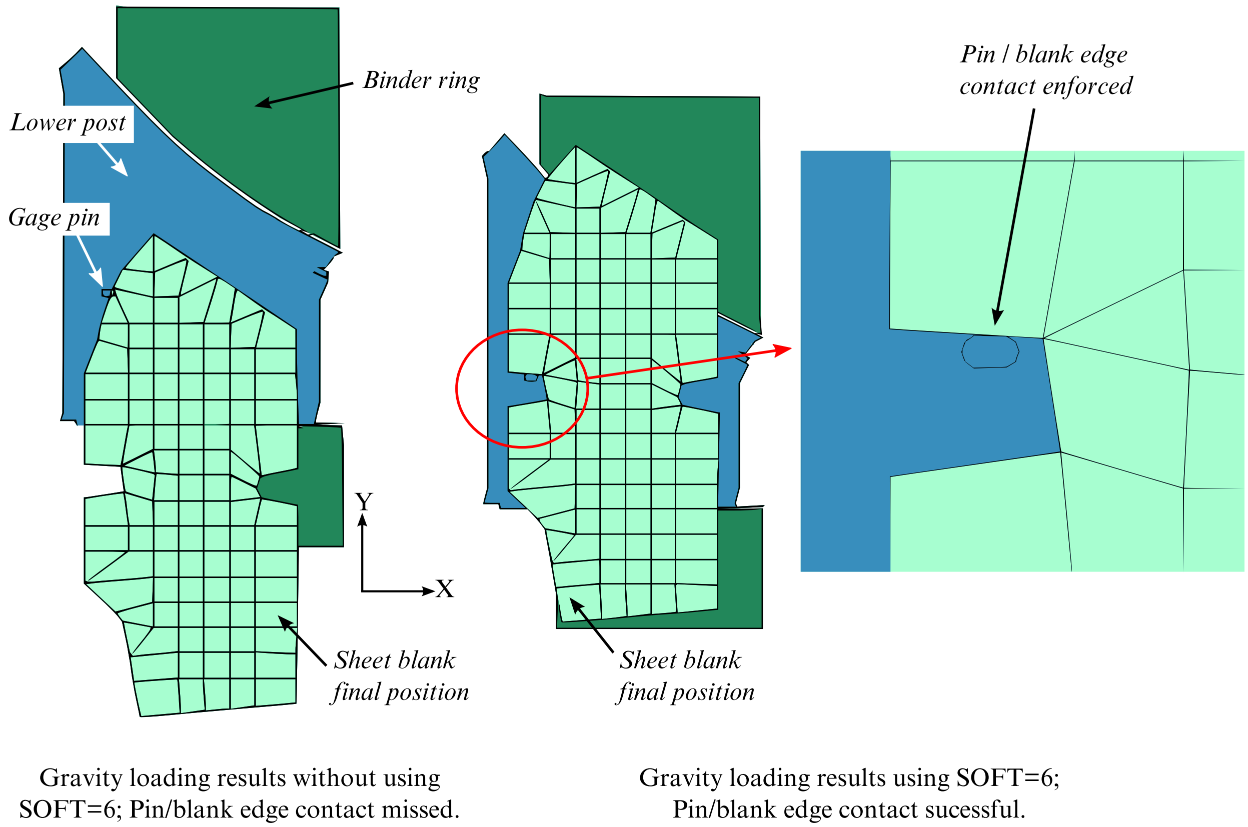

Binder ring
Lower post
Gage pin
Y
X
Sheet blank
final position
Sheet blank
final position
Pin / blank edge
contact enforced
Gravity loading results without using
SOFT=6; Pin/blank edge contact missed.
Gravity loading results using SOFT=6;
Pin/blank edge contact sucessful.

Figure 11-12. Final blank position without (left) and with (right) SOFT = 6.

About SOFT = 6:

SOFT = 6 contact addresses contact issues in situation where blank gage pins are narrow or small and blank mesh are coarse (Figure 11-11 left), leading to missing contact in some cases. This feature applies only to gravity loading of sheet blank with non-adaptive mesh, and **for use with *CONTACT_FORMING_NODES_TO_SURFACE only**.

Nodes along the entire or a portion of the blank edge to be contacted with gage pins must be included in a node set for the variable SSID in * CONTACT_FORMING_NODES_TO_SURFACE (Figure 11-11 left). The nodes in the node set must be listed in a consecutive order, as defined "*by path*" in *LSPP4.0*, under *Model → CreEnt → Cre → Set Data → *SET_NODE*. No thickness exists for either blank edge or gage pins. In addition, the variable ORIENT in *CONTROL_CONTACT must be set to "4". Currently this feature is available in double precision, SMP only, starting in Revision 81297 and later releases. *Starting in Revision 110072, SSID in *CONTACT_FORMING_NODES_TO_SURFACE can be input as part ID of the blank, making it much easier to use SOFT = 6.*

In a partial keyword example below, node set ID 21 (SSTYP = 4) is in contact with gage pin of part set ID 20. As shown in Figure 11-11 (left), with the boundary condition applied, a blank with a very coarse mesh is loaded with a body force. The left notch is anticipated to be in contact with the gage pins. The initial position (top view) of the test model is shown in Figure 11-11 (right) and the final gravity loaded blank positions are shown in Figure 11-12 (left) without SOFT = 6, and in Figure 11-12 (middle and right) with SOFT = 6, respectively. It is shown that without the SOFT = 6 the contact between the blank edge and the pin missed completely.

```
*CONTROL_TERMINATION
1.0
*CONTROL_IMPLICIT_FORMING
1
*CONTROL_IMPLICIT_GENERAL
1,0.2
*CONTROL_IMPLICIT_NONLINEAR
$  NSLOLVR    ILIMIT    MAXREF     DCTOL     ECTOL     RCTOL     LSTOL
         2         1      1200     0.000      0.00         0
$    dnorm   divflag   inistif
         0         2         0        1                  1
*SET_NODE_LIST
$ blank edge node set around the gage pin
21
1341,1342,1343,1344
*SET_PART_LIST
$ gage pin
20
20
*SET_PART_LIST
$ blank
13
13
*CONTACT_FORMING_NODES_TO_SURFACE
$     SSID      MSID     SSTYP     MSTYP    SBOXID    MBOXID       SPR       MPR
        21        20         4         2
$       FS        FD        DC         V       VDC    PENCHK        BT        DT
     0.125                                     20.         4
$      SFS       SFM       SST       MST      SFST      SFMT       FSF       VSF

$     SOFT
         6
```

Beginning in Revision109342, an SSTYP of 3 (a part PID, **not a part set ID**) can be used for the SSID, simplifying the definition of contact interfaces for SOFT = 6. In Soft 6 contact definition below, a blank PID of 13 is defined for the SSID using SSTYP = 3:

```
*CONTACT_FORMING_NODES_TO_SURFACE
$     SSID      MSID     SSTYP     MSTYP    SBOXID    MBOXID       SPR       MPR
        13        20         3         2
$       FS        FD        DC         V       VDC    PENCHK        BT        DT
     0.125                                     20.         4
$      SFS       SFM       SST       MST      SFST      SFMT       FSF       VSF

$     SOFT
         6
```

Optional Card B:

Reminder: If Optional Card B is used, then Optional Card A must be defined. (Optional Card A may be a blank line).

Optional Card B.

Optional	1	2	3	4	5	6	7	8
Variable	PENMAX	THKOPT	SHLTHK	SNLOG	ISYM	I2D3D	SLDTHK	SLDSTF
Type	F	I	I	I	I	I	F	F
Default	0	0	0	0	0	0	0	0
Remarks		Old types 3, 5, 10	Old types 3, 5, 10					

VARIABLE	DESCRIPTION

PENMAX

Maximum penetration distance for old type 3, 5, 8, 9, 10 and Mortar contact or the segment thickness multiplied by PENMAX defines the maximum penetration allowed (as a multiple of the segment thickness) for contact types a 3, a 5, a10, 13, 15, and 26. (see discussion at end of section, including Table 11-17):

EQ.0.0: for old type contacts 3, 5, and 10: Use small penetration search and value calculated from thickness and XPENE, see *CONTROL_CONTACT.

EQ.0.0: for contact types a 3, a 5, a10, 13, and 15: Default is 0.4, or 40 percent of the segment thickness

EQ.0.0: for contact type26 the default value is the segment thickness multiplied by 10

EQ.0.0: for Mortar contact the default is a characteristic size of the element, see Theory manual

VARIABLE **DESCRIPTION**

THKOPT

Thickness option for contact types 3, 5, and 10:

EQ.0: default is taken from control card, *CONTROL_CONTACT,

EQ.1: thickness offsets are included,

EQ.2: thickness offsets are not included (old way).

SHLTHK

Define if and only if THKOPT above equals 1. Shell thickness considered in type surface to surface and node to surface type contact options, where options 1 and 2 below activate the new contact algorithms. The thickness offsets are always included in single surface and constraint method contact types:

EQ.0: thickness is not considered,

EQ.1: thickness is considered but rigid bodies are excluded,

EQ.2: thickness is considered including rigid bodies.

SNLOG

Disable shooting node logic in thickness offset contact. With the shooting node logic enabled, the first cycle that a slave node penetrates a master segment, that node is moved back to the master surface without applying any contact force.

EQ.0: logic is enabled (default),

EQ.1: logic is skipped (sometimes recommended for metalforming calculations or for contact involving foam materials).

ISYM

Symmetry plane option:

EQ.0: off,

EQ.1: do not include faces with normal boundary constraints (e.g., segments of brick elements on a symmetry plane).

This option is important to retain the correct boundary conditions in the model with symmetry. For the ERODING contacts this option may also be defined on card 4.

I2D3D

Segment searching option:

EQ.0: search 2D elements (shells) before 3D elements (solids, thick shells) when locating segments.

EQ.1: search 3D (solids, thick shells) elements before 2D elements (shells) when locating segments.

VARIABLE	DESCRIPTION
SLDTHK	Optional solid element thickness. A nonzero positive value will activate the contact thickness offsets in the contact algorithms where offsets apply. The contact treatment will then be equivalent to the case where null shell elements are used to cover the brick elements. The contact stiffness parameter below, SLDSTF, may also be used to override the default value. This parameter applies also to Mortar contacts, but SLDSTF is then ignored.
SLDSTF	Optional solid element stiffness. A nonzero positive value overrides the bulk modulus taken from the material model referenced by the solid element. For segment based contact (SOFT = 2), SLDSTF replaces the stiffness used in the penalty equation. This parameter does not apply to Mortar contacts.

Optional Card C:

Reminder: If Optional Card C is used, then Optional Cards A and B must be defined. (Optional Cards A and B may be blank lines).

Optional Card C.

Optional	1	2	3	4	5	6	7	8
Variable	IGAP	IGNORE	DPRFAC / MPAR1	DTSTIF / MPAR2			FLANGL	CID_RCF
Type	I	I	F	F			F	I
Default	1	0	0	0			0	0
Remarks		3	1	2				

VARIABLE	DESCRIPTION

IGAP

For mortar contact IGAP is used to progressively increase contact stiffness for large penetrations, see remarks on mortar contact below.

For other contacts it is a flag to improve implicit convergence behavior at the expense of (1) creating some sticking if parts attempt to separate and (2) possibly underreporting the contact force magnitude in the output files rcforc and ncforc. (IMPLICIT ONLY.).

LT.0: Set IGAP = 1 and set the distance for turning on the stiffness to (IGAP/10) times the original distance.

EQ.1: Apply method to improve convergence (DEFAULT)

EQ.2: Do not apply method

GT.2: Set IGAP = 1 for first IGAP − 2 converged equilibrium states, then set IGAP = 2

VARIABLE **DESCRIPTION**

IGNORE

Ignore initial penetrations in the *CONTACT_AUTOMATIC options.

LT.0: Applies only to the Mortar contact. When less than zero, the behavior is the same as for |IGNORE|, but contact between segments belonging to the same part is ignored.

The main purpose of this option is to avoid spurious contact detections that otherwise could result for complicated geometries in a single surface contact, typically, when eliminating initial penetrations by interference. See IGNORE.EQ.3 and IGNORE.EQ.4.

EQ.0: Take the default value from the fourth card of the CONTROL_CONTACT input.

EQ.1: Allow initial penetrations to exist by tracking the initial penetrations.

EQ.2: Allow initial penetrations to exist by tracking the initial penetrations. However, penetration warning messages are printed with the original coordinates and the recommended coordinates of each slave node given.

EQ.3: Applies only to the Mortar contact. With this option initial penetrations are eliminated between time zero and the time specified by MPAR1. Intended for small initial penetrations. See remarks on Mortar contact.

EQ.4: Applies only to the Mortar contact. With this option initial penetrations are eliminated between time zero and the time specified by MPAR1. In addition a maximum penetration distance can be given as MPAR2, intended for large initial penetrations. See remarks on Mortar contact.

VARIABLE	DESCRIPTION
DPRFAC/ MPAR1	Applies to the SOFT = 2 and Mortar contacts. Depth of penetration reduction factor (DPRFAC) for SOFT = 2 contact. EQ.0.0: Initial penetrations are always ignored. GT.0.0: Initial penetrations are penalized over time. LE.-1.0: |DPRFAC| is the load curve ID defining DPRFAC versus time. For the mortar contact MPAR1 corresponds to initial contact pressure in interfaces with initial penetrations if IGNORE = 2, for IGNORE = 3,4 it corresponds to the time of closure of initial penetrations. See remarks below.
DTSTIF/ MPAR2	Applies to the SOFT = 1 and SOFT = 2 and Mortar contacts. Time step used in stiffness calculation for SOFT = 1 and SOFT = 2 contact. EQ.0.0: Use the initial value that is used for time integration. GT.0.0: Use the value specified. $\in (-1.0, -0.01)$: use a moving average of the solution time step. (SOFT = 2 only) LE.-1.0: |DTSTIF| is the ID of a curve that defines DTSTIF vs. time. For the mortar contact and IGNORE = 4, MPAR2 corresponds a penetration depth that must be at least the penetration occurring in the contact interface. See remarks below.
FLANGL	Angle tolerance in radians for feature lines option in smooth contact. EQ.0.0: No feature line is considered for surface fitting in smooth contact. GT.0.0: Any edge with angle between two contact segments bigger than this angle will be treated as feature line during surface fitting in smooth contact.
CID_RCF	Coordinate system ID to output rcforc force resultants and ncforc data in a local system.

Remarks:

1. DPRFAC/MPAR1 is used only by segment based contact (SOFT = 2) and Mortar Contact (see general remarks below for the latter option). By default, SOFT = 2 contact measures the initial penetration between segment pairs that are found to be in contact and subtracts the measured value from the total penetration for as long as a pair of segments remains in contact. The penalty force is proportional to this modified value. This approach prevents shooting nodes, but may allow unacceptable penetration. DPRFAC can be used to decrease the measured value over time until the full penetration is penalized. Setting DPRFAC = 0.01 will cause ~1% reduction in the measured value each cycle. The maximum allowable value for DPRFAC is 0.1. A small value such as 0.001 is recommended. DPRFAC does not apply to initial penetrations at the start of the calculation, only those that are measured at later times. This prevents non-physical movement and energy growth at the start of the calculation.

2. The anticipated use for the load curve option is to allow the initial penetrations to be reduced at the end of a calculation if the final geometry is to be used for a subsequent analysis. To achieve this, load curve should have a y-value of zero until a time near the end of the analysis and then ramp up to a positive value such as 0.01 near the end of the analysis.

3. DTSTIF/MPAR2 is used only by the SOFT = 1 and SOFT = 2 contact options and the Mortar contact (for the latter, see remarks on Mortar contact). By default when the SOFT option is active, the contact uses the initial solution time step to scale the contact stiffness. If the user sets DTSTIF to a nonzero value, the inputted value will be used. Because the square of the time step appears in the denominator of the stiffness calculation, a DTSTIF value larger than the initial solution time step reduces the contact stiffness and a smaller value increases the stiffness. This option could be used when one component of a larger model has been analyzed independently and validated. When the component is inserted into the larger model, the larger model may run at a smaller time step due to higher mesh frequencies. In the full model analysis, setting DTSTIF equal to the component analysis time step for the contact interface that treats the component will cause consistent contact stiffness between the analyses.

   The load curve option allows contact stiffness to be a function of time. This should be done with care as energy will not be conserved. A special case of the load curve option is when |DTSTIF| = LCTM on *CONTROL_CONTACT. LCTM sets an upper bound on the solution time step. For |DTSTIF| = LCTM, the contact stiffness time step value will track LCTM whenever the LCTM value is less than the initial solution time step. If the LCTM value is greater, the initial solution time step is used. This option could be used to stiffen the contact at the end of an analysis. To achieve this, the LCTM curve should be defined such that it is larger than the solution time step until near the end of the analysis.

Then the LCTM curve should ramp down below the solution time step causing it to decrease and the contact to stiffen. A load curve value of 0.1 of the calculated solution time step will cause penetrations to reduce by about 99%. To prevent shooting nodes, the rate at which the contact stiffness increases is automatically limited. Therefore, to achieve 99% reduction, the solution should be run for perhaps 1000 cycles with a small time step.

For segment based contact (SOFT = 2), setting DTSTIF less than or equal to -0.01 and greater than -1.0, causes the contact stiffness to be updated based on the current solution time step. Varying the contact stiffness during a simulation can cause energy growth so this option should be used with care when extra stiffness is needed to prevent penetration and the solution time step has dropped from the initial. Because quick changes in contact stiffness can cause shooting nodes, using a moving average of the solution time step can prevent this. The value of DTSTIF determines the number of terms in the moving average where n = 100 × (-DTSTIF) such that n = 1 for DTSTIF = -0.01 and n = 100 for DTSTIF = -0.999. Setting DTSTIF = -1.0 triggers the load curve option described in the previous paragraph, so DTSTIF cannot be smaller than -0.999 for this option.

4. When SOFT = 2 on Optional Card A of *CONTACT, treatment of initial penetrations is always like IGNORE = 1 in that initial penetrations are ignored when calculating penalty forces. If SOFT = 2 and IGNORE = 2, then a report of initial penetrations will be written to the messag file(s) in the first cycle.

Optional Card D:

Reminder: If Optional Card D is used, then Optional Cards A, B and C must be defined. (Optional Cards A, B and C may be blank lines).

Optional Card D.

Optional	1	2	3	4	5	6	7	8
Variable	Q2TRI	DTPCHK	SFNBR	FNLSCL	DNLSCL	TCSO	TIEDID	SHLEDG
Type	I	F	F	F	F	I	I	I
Default	0	0	0	0	0	0	0	
Remarks	1	2	3	5	5		4	

VARIABLE	DESCRIPTION
Q2TRI	Option to split quadrilateral contact segments into two triangles (only available when SOFT = 2). EQ.0: Off (default). EQ.1: On for all slave shell segments. EQ.2: On for all master shell segments. EQ.3: On for all shell segments. EQ.4: On for all shell segments of material type 34.
DTPCHK	Time interval between shell penetration reports (only available for segment based contact) EQ.0.0: Off (default). GT.0.0: Check and report segment penetrations at time intervals equal to DTPCHK

VARIABLE **DESCRIPTION**

SFNBR

Scale factor for neighbor segment contact (only available for segment based contact)

EQ.0.0: Off (default).

GT.0.0: Check neighbor segments for contact

LT.0.0: Neighbor segment checking with improved energy balance when |SFNBR| < 1000. |SFNBR|≥1000 activates a split-pinball based neighbor contact with a penalty force scale factor of |SFNBR+1000|. For example, force scale factor used is 2 when SFNBR = -1002.

FNLSCL

Scale factor for nonlinear force scaling.

TCSO

Option to consider only contact segments (not all attached elements) when computing the contact thickness for a node or segment (for SEGMENT_TO_SEGMENT contact and shell elements only)

EQ.0: Off (default).

EQ.1: Only consider segments in the contact definition

DNLSCL

Distance for nonlinear force scaling.

TIEDID

Incremental displacement update for tied contacts. (See remark 4 below.)

EQ.0: Off (default).

EQ.1: On

SHLEDG

Flag for assuming edge shape for shells when measuring penetration. This is available for segment based contact (see SOFT on optional card A)

EQ.0: default to SHLEDG on *CONTROL_CONTACT

EQ.1: Shell edges are assumed square and are flush with the nodes

EQ.2: Shell edges are assumed round with radius equal to ½ shell thickness

Remarks:

1. **Q2TRI.** Setting Q2TRI to a nonzero value causes quadrilateral shell segments to be spilt into two triangles. The contact segments only are split. The elements are not changed. This option is only available for segment based contact which is activated by setting SOFT = 2.

2. **DTPCHK (Penetration Check).** Setting DTPCHK to a positive value causes a penetration check to be done periodically with the interval equal to DTPCHK. The check looks for shell segments that are penetrating the mid-plane of another shell segment. It does not report on penetration of thickness offsets. The penetrating pairs are reported to the messag file or files for MPP. If at least one penetration is found, the total number of pairs is reported to the screen output. This option is only available for segment based contact which is activated by setting SOFT = 2.

3. **SFNBR.** SFNBR is a scale factor for optional neighbor segment contact checking. This is available only in segment based (SOFT=2) contact. This is helpful option when a mesh folds as can happen with compression folding of an airbag. Only shell element segments are checked. Setting SFNBR to a negative value modifies the neighbor checking to improve energy balance. When used, a value between -0.5 and -1.0 is recommended.

4. **Round off in OFFSET and TIEBREAK.** There have been several issues with tied OFFSET contacts and AUTOMATIC_TIEBREAK contacts with offsets creating numerical round-off noise in stationary parts. By computing the interface displacements incrementally rather than using total displacements, the round-off errors that occur in single precision are eliminated. The incremental approach is available for the following contact types:

   TIED_SURFACE_TO_SURFACE_OFFSET

   TIED_NODES_TO_SURFACE_OFFSET

   TIED_NODES_TO_SURFACE_CONSTRAINED_OFFSET

   TIED_SHELL_EDGE_TO_SURFACE_OFFSET

   AUTOMATIC_…_TIEBREAK

5. **FNLSCL.** FNLSCL = f and DNLSCL = d invoke alternative contact stiffness scaling options.

   When FNLSCL > 0 and DNLSCL > 0, the first option scales the stiffness by the depth of penetration to provide smoother initial contact and a larger contact force as the depth of penetration exceeds DNLSCL. The stiffness k is scaled by the relation

$$k \rightarrow kf\sqrt{\frac{\delta}{d}}$$

where δ is the depth of penetration, making the penalty force proportional to the 3/2 power of the penetration depth. Adding a small amount of surface damping (e.g., VDC = 10) is advised with this option.

When SOFT = 2 and FNLSCL < 0, DNLSCL > 0, an alternative stiffness scaling scheme is used,

$$k \rightarrow k\left[\frac{0.01fA_o}{d(d-\delta)}\right]$$

where A_0 is the overlap area of segments in contact. For δ greater than $0.9d$, the stiffness is extrapolated to prevent it from going to infinity.

When SOFT = 2, FNLSCL > 0, and DNLSCL = 0, an option to scale the contact by the overlap area is invoked.

$$k \rightarrow kf\left(\frac{A_o}{A_m}\right)$$

where A_m is the mean area of all the contact segments in the contact interface. This third option can improve friction behavior, particularly when the FS = 2 option is used.

Optional Card E:

Reminder: If Optional Card E is used, then Optional Cards A, B, C and D must be defined. (Optional Cards A, B, C and D may be blank lines).

Optional Card E.

Optional	1	2	3	4	5	6	7	8
Variable	SHAREC	CPARM8	IPBACK	SRNDE	FRICSF	ICOR	FTORQ	REGION
Type	I	I	I	I	F	I	I	I
Default	0	0	0	0	1.	0	0	0
Remarks	1		2	3	5			4

VARIABLE	DESCRIPTION
SHAREC	Shared constraint flag (only available for segment based contact) EQ.0: Segments that share constraints not checked for contact. EQ.1: Segments that share constraints are checked for contact.
CPARM8	This variable is similar to CPARM8 in *CONTACT_…_MPP but applies to SMP and not to MPP. CPARM8 for SMP only controls treatment of spot weld beams in CONTACT_AUTOMATIC_-GENERAL. EQ.0: Spot weld (type 9) beams are not considered in the contact even if included on the slave side of the contact. EQ.2: Spot weld (type 9) beams are considered in the contact if included on the slave side of the contact.
IPBACK	If set to a nonzero value, creates a "backup" penalty tied contact for this interface. This option applies to constrained tied contacts only. See Remark 2.

VARIABLE **DESCRIPTION**

SRNDE

Flag for non-extended exterior shell edges. See Remark 3 below for further information and restrictions:

EQ.0: Exterior shell edges have their usual treatment where the contact surface extends beyond the shell edge.

EQ.1: The contact surface is rounded at exterior shell edges but does not extend beyond the shell edges.

EQ.2: The shell edges are square.

FRICSF

Scale factor for frictional stiffness (available for SOFT = 2 only).

ICOR

If set to a nonzero value, VDC is the coefficient of restitution expressed as a percentage. When SOFT = 0 or 1, this option applies to AUTOMATIC_NODES_TO_SURFACE, AUTOMATIC_SURFACE_TO_SURFACE and AUTOMATIC_SINGLE_SURFACE. When SOFT = 2, it applies to all available keywords.

FTORQ

If set to 1, a torsional force is computed in the beam to beam portion of contact type AUTOMATIC_GENERAL, which balances the torque produced due to friction. This is currently only available in the MPP version.

REGION

The ID of a *DEFINE_REGION which will delimit the volume of space where this contact is active. See Remark 4 below.

Remarks:

1. The SHAREC flag is a segment based contact option that allows contact checking of segment pairs that share a multi-point constraint or rigid body. Sharing a constraint is defined as having at least one node of each segment that belongs to the same constraint.

2. The IPBACK flag is only applicable to constraint based tied contacts (TIED with no options, or with CONSTRAINED_OFFSET). An identical penalty based contact is generated with type OFFSET, except in the case of SHELL_EDGE constrained contact which generates a BEAM_OFFSET type. The ID of the generated interface will be set to the ID of the original interface plus 1 if that ID is available, otherwise one more than the maximum used contact ID. For nodes successfully tied by the constraint interface, the extra penalty tying should not cause problems, but nodes dropped from the constraint interface due to rigid body or other conflicting constraints will be handled by the penalty contact. In MPP, nodes successfully tied by the constraint interface are skipped during the penalty contact phase.

3. The SRNDE option only applies to SOFT = 0 and SOFT = 1 contacts in the MPP version. Shell edges for these contacts are by default treated by adding cylindrical caps along the free edges, with the radius of the cylinder equal to half the thickness of the segment. This has the side effect of extending the segment at the free edges, which can cause problems. Setting SRNDE = 1 "rounds over" the (through the thickness) corners of the element instead of extending it. The edges of the segment are still rounded, but the overall size of the contact area is not increased. The effect is as if the free edge of the segment was moved in toward the segment by a distance equal to half the segment thickness, and then the old cylindrical treatment was performed. Setting SRNDE = 2 will treat the shell edges as square, with no extension. This variable has no effect on shell-edge-to-shell-edge interaction in AUTOMATIC_GENERAL; for that, see CPARM8 on the MPP Card.

   The SRNDE = 1 option is available for the AUTOMATIC_SINGLE and AUTOMATIC_GENERAL contacts. The NODE_TO_SURFACE and SURFACE_TO_SURFACE contacts also support SRNDE = 1 if the GROUPABLE option is used. The SRNDE = 2 option is available for all these contact types if the GROUPABLE option is enabled.

4. Setting a non-zero value for REGION does not limit or in any way alter the list of slave or master nodes or segments, and this option should not be used for that purpose. For efficiency, the smallest possible portion of the model should be defined as slave or master using the normal mechanisms for specifying the slave and master surfaces. Setting a non-zero value will, however, result in contact outside the REGION being ignored. As slave and master nodes and segments pass into the indicated REGION, contact for them will become active. As they pass out of the REGION, they will be skipped in the contact calculation. This option is currently only available for the MPP version, and only for contacts of type AUTOMATIC_SINGLE_SURFACE, and AUTOMATIC_*_TO_SURFACE.

5. The FRICSF factor is an optional factor to scale the frictional stiffness. FRICSF is available only when SOFT = 2 on optional card A. With penalty contact, the frictional force is a function of the stiffness, the sliding distance, and the Coulomb limit.

Optional Card F:

Reminder: If Optional Card F is used, then Optional Cards A, B, C, D and E must be defined. (Optional Cards A, B, C, D and E may be blank lines).

Optional Card F.

Optional	1	2	3	4	5	6	7	8
Variable	PSTIFF	IGNROFF						
Type	I	I						
Default	0	0						
Remarks	1							

VARIABLE	DESCRIPTION
PSTIFF	Flag to choose the method for calculating the penalty stiffness. This is available for segment based contact (see SOFT on optional card A) EQ.0: Use the default as defined by PSTIFF on *CONTROL_-CONTACT. EQ.1: Based on nodal masses EQ.2: Based on material density and segment dimensions.
IGNROFF	Flag to ignore the thickness offset for shells in the calculation of the shell contact penetration depth. This allows shells to be used for meshing rigid body dies without modifying the positions of the nodes to compensate for the shell thickness. EQ.1: Ignore the master side thickness. EQ.2: Ignore the slave side thickness. EQ.3: Ignore the thickness of both sides..

Remarks:

1. See Remark 6 on *CONTROL_CONTACT for an explanation of the PSTIFF option. Specifying PSTIFF here will override the default value as defined by PSTIFF on *CONTROL_CONTACT.

General Remarks: *CONTACT

1. Modeling airbag interactions with structures and occupants using the actual fabric thickness, which is approximate 0.30 mm, may result in a contact breakdown that leads to inconsistent occupant behavior between different machines. Based on our experience, using a two-way automatic type contact definition, i.e., AUTOMATIC_SURFACE_TO_SURFACE, between any airbag to structure/occupant interaction and setting the airbag fabric contact thickness to at least 10 times the actual fabric thickness has helped improved contact behavior and eliminates the machine inconsistencies. Due to a large stiffness difference between the airbag and the interacting materials, the soft constraint option (SOFT = 1) or the segment based option (SOFT = 2) is recommended. It must be noted that with the above contact definition, only the airbag materials should be included in any *AIRBAG_SINGLE_SURFACE definitions to avoid duplicate contact treatment that can lead to numerical instabilities.

2. The following contact definitions are based on constraint equations and will not work with rigid bodies:

   TIED_NODES_TO_SURFACE

   TIED_NODES_TO_SURFACE_CONSTRAINED_OFFSET

   TIED_SURFACE_TO_SURFACE

   TIED_SURFACE_TO_SURFACE_CONSTRAINED_OFFSET

   TIED_SHELL_EDGE_TO_SURFACE

   TIED_SHELL_EDGE_TO_SURFACE_CONSTRAINED_OFFSET

   TIED_SHELL_EDGE_TO_SOLID

   SPOTWELD

   SPOTWELD_WITH_TORSION

   However, SPOTWELD_WITH_TORSION_PENALTY does work with rigid bodies and tied interfaces with the offset option can be used with rigid bodies, i.e., TIED_NODES_TO_SURFACE_OFFSET

TIED_SHELL_EDGE_TO_SURFACE_OFFSET

TIED_SHELL_EDGE_TO_SURFACE_BEAM_OFFSET

TIED_SURFACE_TO_SURFACE_OFFSET

Also, it may sometimes be advantageous to use the CONSTRAINED_EXTRA_NODE_*OPTION* instead for tying deformable nodes to rigid bodies since in this latter case the tied nodes may be an arbitrary distance away from the rigid body.

Tying will only work if the surfaces are near each other. The criteria used to determine whether a slave node is tied down is that it must be “close”. For shell elements “close” is defined as distance, δ, less than:

$$\delta_1 = 0.60 \times (\text{thickness of slave node} + \text{thickness of master segment})$$

$$\delta_2 = 0.05 \times \min(\text{master segment diagonals})$$

$$\delta = \max(\delta_1, \delta_2)$$

If a node is further away it will not be tied and a warning message will be printed. For solid elements the slave node thickness is zero and the segment thickness is the element volume divided by the segment area; otherwise, the same procedure is used.

If there is a large difference in element areas between the master and slave side, the distance, δ_2, may be too large and may cause the unexpected projection of nodes that should not be tied. This can occur during calculation when adaptive remeshing is used. To avoid this difficulty the slave and master thickness can be specified as negative values on Card 3 in which case

$$\delta = \text{abs}(\delta_1)$$

3. The contact algorithm for tying spot welds with torsion, SPOTWELD_WITH_TORSION, must be used with care. Parts that are tied by this option should be subjected to stiffness proportional damping of approximately ten percent, i.e., input a coefficient of 0.10. This can be defined for each part on the *DAMPING_PART_STIFFNESS input. Stability problems may arise with this option if damping is not used. This comment applies also to the PENALTY keyword option.

4. These contact definitions must be used with care. The surface and the nodes which are constrained to a surface are not allowed to be used in any other CONSTRAINT_… contact definition:

CONSTRAINT_NODES_TO_SURFACE

CONSTRAINT_SURFACE_TO_SURFACE

If, however, contact has to be defined from both sides as in sheet metal forming, one of these contact definitions can be a CONSTRAINT type; the other one could be a standard penalty type such as SURFACE_TO_SURFACE or NODES_TO_SURFACE.

5. These contact definitions require thickness to be taken into account for rigid bodies modeled with shell elements. Therefore, care should be taken to ensure that realistic thicknesses are specified for the rigid body shells.

   AIRBAG_SINGLE_SURFACE

   AUTOMATIC_GENERAL

   AUTOMATIC_GENERAL_INTERIOR

   AUTOMATIC_NODES_TO_SURFACE

   AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE

   AUTOMATIC_SINGLE_SURFACE

   AUTOMATIC_SURFACE_TO_SURFACE

   SINGLE_SURFACE

   A thickness that is too small may result in loss of contact and an unrealistically large thickness may result in a degradation in speed during the bucket sorts as well as nonphysical behavior. The SHLTHK option on the *CONTROL_CONTACT card is ignored for these contact types.

6. Two methods are used in LS-DYNA for projecting the contact surface to account for shell thicknesses. The choice of methods can influence the accuracy and cost of the calculation. Segment based projection is used in contact types: (See Figure 11-13)

   AIRBAG_SINGLE_SURFACE

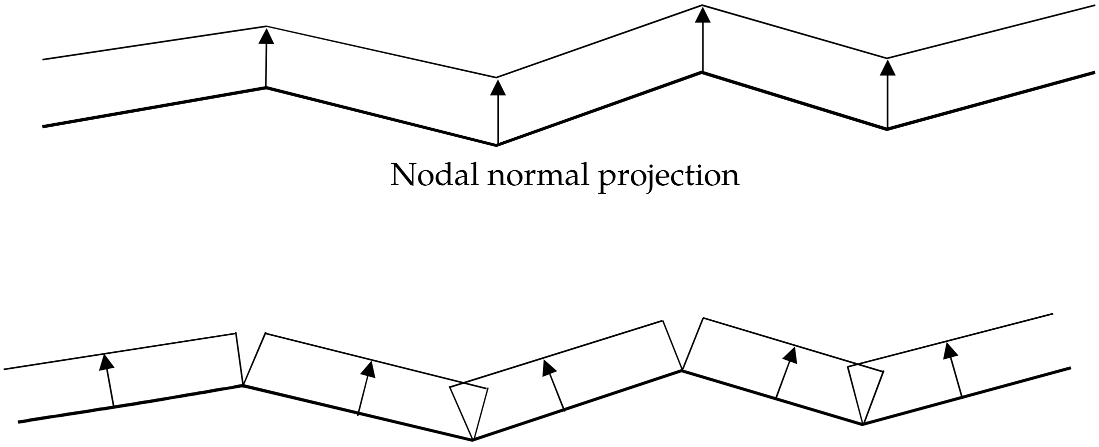
   AUTOMATIC_GENERAL

   AUTOMATIC_NODES_TO_SURFACE

   AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE

   AUTOMATIC_SINGLE_SURFACE

   AUTOMATIC_SURFACE_TO_SURFACE



Nodal normal projection

Segment based projection

Figure 11-13. Nodal normal and segment based projection is used in the contact options

FORMING_NODES_TO_SURFACE

FORMING_ONE_WAY_SURFACE_TO_SURFACE

FORMING_SURFACE_TO_SURFACE

The remaining contact types use nodal normal projections if projections are used. The main advantage of nodal projections is that a continuous contact surface is obtained which is much more accurate in applications such as metal forming. The disadvantages of nodal projections are the higher costs due to the nodal normal calculations, difficulties in treating T-intersections and other geometric complications, and the need for consistent orientation of contact surface segments. The contact type

SINGLE_SURFACE

uses nodal normal projections and consequently is slower than the alternatives.

7. These contact algorithms allow the total contact forces applied by all contacts to be picked up.

   FORCE_TRANSDUCER_PENALTY

   FORCE_TRANSDUCER_CONSTRAINT

8. This contact does not apply any force to the model and will have no effect on the solution. Only the slave set and slave set type need be defined for this contact type. Generally, only the first three cards are defined. The force transducer option, PENALTY, works with penalty type contact algorithms only, i.e., it does

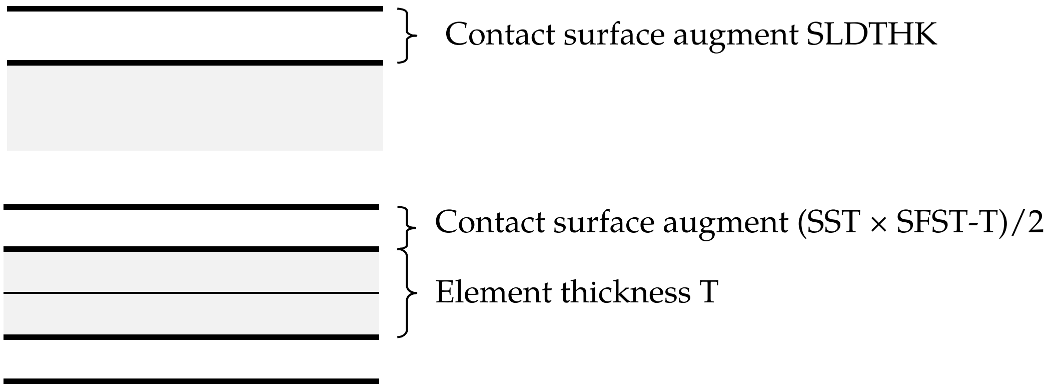

Contact surface augment SLDTHK
Contact surface augment (SST × SFST-T)/2
Element thickness T

Figure 11-14. Illustration of contact surface location for automatic Mortar contact, solids on top and shells below.

not work with the CONSTRAINT or TIED options. For these latter options, use the CONSTRAINT option. If a transducer is used for extracting forces from Mortar contacts, the slave and master sides *must* be defined through parts or part sets, segment or node sets will not gather the correct data.

> **NOTE:** If the interactions between two surfaces are needed, a master surface should be defined. In this case, only the contact forces applied between the slave and master surfaces are kept. The master surface option is only implemented for the PENALTY option and works only with the AUTOMATIC contact types.

9. FORMING_… These contacts are mainly used for metal forming applications. A connected mesh is not required for the master (tooling) side but the orientation of the mesh **must** be in the same direction. These contact types are based on the AUTOMATIC type contacts and consequently the performance is better than the original two surface contacts.

10. The mortar contact, invoked by appending the suffix MORTAR to either FORMING_SURFACE_TO_SURFACE, AUTOMATIC_SURFACE_TO_SURFACE or AUTOMATIC_SINGLE_SURFACE is a segment to segment penalty based contact. For two segments on each side of the contact interface that are overlapping and penetrating, a consistent nodal force assembly taking into account the individual shape functions of the segments is performed, see Figure 11-16 for an illustration. A TRANSDUCER_PENALTY can be used for extracting forces from Mortar contacts, *but the slave and master sides must then be defined through parts or part sets*.

    In this respect the results with this contact may be more accurate, especially when considering contact with elements of higher order. By appending the suffix TIED to the CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_MORTAR keyword or the suffix TIEBREAK_MORTAR (only OPTION = 7 and OP-

TION = 9 supported) to the CONTACT_AUTOMATIC_SURFACE_TO_SURFACE keyword, this is treated as a tied contact interface with optional failure in the latter case. This contact is intended for implicit analysis in particular but is nevertheless supported for explicit analysis as well. For explicit analysis, the bucket sort frequency is 100 if not specified.

The FORMING *mortar* contact, in contrast to other forming contacts, does *not* assume a rigid master side, but if this side consists of shell elements the normal should be oriented towards the slave side. Furthermore, no shell thickness is taken into account on the master side. The slave side is assumed to be a deformable shell part, and the orientation of the elements does not matter. However, each FORMING contact definition should be such that contact occurs with ONE deformable slave side only, which obviously leads to multiple contact definitions if two-sided contact is presumed. The AUTOMATIC contact is supported for solids, shells and beams, and here the thicknesses are taken into account both for rigid and deformable parts. Flat edge contact is supported for shell elements and contact with beams occurs on the lateral surface area as well as on the end tip. The contact assumes that the beam has a cylindrical shape with a cross sectional area coinciding with that of the underlying beam element.

For the AUTOMATIC contact, the contact surface can be augmented with the aid of parameters SST and SFST for shells and beams, while SLDTHK is used for solids and thick shells. For shells/beams SST corresponds to the contact thickness of the element (MST likewise for the master side), by default this is the same as the element thickness. This parameter can be scaled with aid of SFST (SFMT for the master side) to adjust the location of the contact surface, see Figure 11-14.

For solids PENMAX can be used to determine the maximum penetration and also determines the search depth for finding contact pairs, if set it should correspond to a characteristic thickness in the model. Also, the contact surface can be adjusted with the aid of SLDTHK if it is of importance to reduce the gap between parts, see Figure 11-14. This may be of interest if initial gaps result in free objects undergoing rigid body motion and thus preventing convergence in implicit.

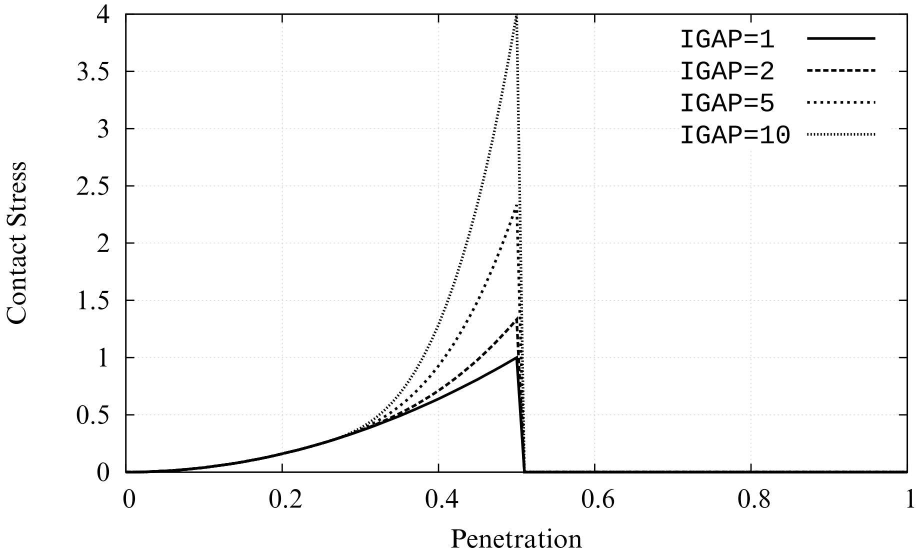


Figure 11-15. Mortar contact stress as function of penetration

For the TIED option, the criterion for tying two contact surfaces is by default that the distance should be less than 0.05 × T, i.e., by default it is within 5% of the element thickness (characteristic size for solids). In this case PENMAX can be used to set the tying distance, i.e., if PENMAX is positive then segments are tied if the distance is less than PENMAX.

If initial penetrations are detected (reported in the messag file) then by default these will yield an initial contact stress corresponding to this level of penetration. IGNORE > 0 can be used to prevent unwanted effects of this. IGNORE = 2 behaves differently than from other contacts, for this option the penetrations are not tracked but the contact surface is fixed at its initial location. In addition, for IGNORE = 2, an initial contact pressure can be imposed on the interface by setting the MPAR1 parameter to the desired contact pressure. All this allows to properly eliminate any rigid body motion due to initial contact gaps.

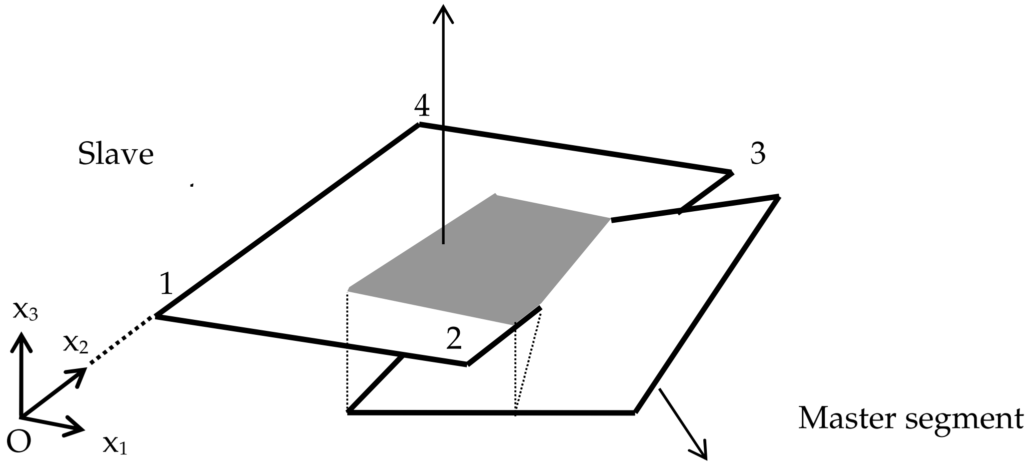


Figure 11-16. Illustration of Mortar segment to segment contact

A third option is IGNORE = 3, for which prestress can be applied. This allows initial penetrations to exists and they are closed during the time between zero and the value given by MPAR1, thus working pretty similar to the INTERFERENCE option with the exception that the closure is linear in time. A limitation with IGNORE = 3 in this context is that the initial penetrations must be small enough for the contact algorithm to detect them.

Thus, for large penetrations IGNORE = 4 is recommended (this can only be used if the slave side consists of solid elements). This does pretty much the same thing as IGNORE = 3, but the user may provide a penetration depth in MPAR2. This depth must be at least as large as (and preferably in the order of) the maximum initial penetration in the contact interface or otherwise an error termination will be the result. The need for such a parameter is for the contact algorithm to have a decent chance to locate the contact surface and thus estimate the initial penetration. With this option the contact surfaces are pushed back and placed in incident contact at places where initial penetrations are present, this can be done for (more or less) arbitrary initial penetration depths. As for IGNORE = 3, the contact surfaces will be restored linearly in the time given by MPAR1.

A problem with mortar contacts in implicit analysis could be that contact pressure is locally very high and leads to large enough penetrations to be released in subsequent steps. Penetration information can be requested on MINFO on *CONTROL_OUTPUT which issues a warning if there is a danger for this to happen. To prevent contact release the user may increase IGAP which penalizes large penetrations without affecting small penetration behavior and thereby overall implicit performance. Figure 11-15 shows the contact pressure as function of penetration for the mortar contact, including the effect of increasing IGAP. It also shows that for sufficiently large penetrations the contact is not detected in subsequent steps which is something to avoid.

INTERFACE TYPE ID	PENCHK	ELEMENT TYPE	FORMULA FOR RELEASE OF PENETRATING NODAL POINT
1, 2, 6, 7	—	—	—

3, 5, 8, 9, 10 (without thickness)	0	solid	d = PENMAX if PENMAX > 0
			d = 1.e+10 if PENMAX = 0

		shell	d = PENMAX if PENMAX > 0
			d = 1.e+10 if PENMAX = 0

	1	solid	d = XPENE × thickness of solid element
		shell	d = XPENE thickness of shell element
	2	solid	d = 0.05 × minimum diagonal length
		shell	d = 0.05 × minimum diagonal length
3, 5, 10 (thickness), 17 and 18	—	solid	d = XPENE × thickness of solid element
		shell	d = XPENE × thickness of shell element
a3, a5, a10, 13, 15	—	solid	d = PENMAX × thickness of solid element *[default: PENMAX = 0.5]*
		shell	d = PENMAX × (slave thickness + master thickness) *[default: PENMAX = 0.4]*
4	—	solid	d = 0.5 × thickness of solid element
		shell	d = 0.4 × (slave thickness + master thickness)
26	—	solid	d = PENMAX × thickness of solid element *[default: PENMAX = 10.0]*
		shell	d = PENMAX × (slave thickness + master thickness) *[default: PENMAX = 10.]*

Table 11-17. Criterion for node release for nodal points which have penetrated too far. This criterion does not apply to SOFT = 2 contact. Larger penalty stiffnesses are recommended for the contact interface which allows nodes to be released. For node-to-surface type contacts (5, 5a) the element thicknesses which contain the node determines the nodal thickness. The parameter is defined on the *CONTROL_CONTACT input.

Mapping of *CONTACT keyword option to "contact type" in d3hsp:

Structured Input Type ID	Keyword Name
a 13	AIRBAG_SINGLE_SURFACE
26	AUTOMATIC_GENERAL
i 26	AUTOMATIC_GENERAL_INTERIOR
a 5	AUTOMATIC_NODES_TO_SURFACE
a 5	AUTOMATIC_NODES_TO_SURFACE_TIEBREAK
a 10	AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE
13	AUTOMATIC_SINGLE_SURFACE
a 3	AUTOMATIC_SURFACE_TO_SURFACE
a 3	AUTOMATIC_SURFACE_TO_SURFACE_TIEBREAK
18	CONSTRAINT_NODES_TO_SURFACE
17	CONSTRAINT_SURFACE_TO_SURFACE
23	DRAWBEAD
16	ERODING_NODES_TO_SURFACE
14	ERODING_SURFACE_TO_SURFACE
15	ERODING_SINGLE_SURFACE
27	FORCE_TRANSDUCER_CONSTRAINT
25	FORCE_TRANSDUCER_PENALTY
m 5	FORMING_NODES_TO_SURFACE
m 10	FORMING_ONE_WAY_SURFACE_TO_SURFACE
m 3	FORMING_SURFACE_TO_SURFACE
5	NODES_TO_SURFACE
5	NODES_TO_SURFACE_INTERFERENCE
10	ONE_WAY_SURFACE_TO_SURFACE
20	RIGID_NODES_TO_RIGID_BODY

Structured Input Type ID	Keyword Name
21	RIGID_BODY_ONE_WAY_TO_RIGID_BODY
19	RIGID_BODY_TWO_WAY_TO_RIGID_BODY
22	SINGLE_EDGE
4	SINGLE_SURFACE
1	SLIDING_ONLY
p 1	SLIDING_ONLY_PENALTY
3	SURFACE_TO_SURFACE
3	SURFACE_TO_SURFACE_INTERFERENCE
8	TIEBREAK_NODES_TO_SURFACE
9	TIEBREAK_SURFACE_TO_SURFACE
6	TIED_NODES_TO_SURFACE
o 6	TIED_NODES_TO_SURFACE_OFFSET
c 6	TIED_NODES_TO_SURFACE_CONSTRAINED_OFFSET
7	TIED_SHELL_EDGE_TO_SURFACE or SPOTWELD
o 7	TIED_SHELL_EDGE_TO_SURFACE_OFFSET
c 7	TIED_SHELL_EDGE_TO_SURFACE_CONSTRAINED_OFFSET or SPOTWELD_CONSTRAINED_OFFSET
b 7	TIED_SHELL_EDGE_TO_SURFACE_BEAM_OFFSET or SPOTWELD_BEAM_OFFSET
s 7	SPOTWELD_WITH_TORSION
2	TIED_SURFACE_TO_SURFACE
o 2	TIED_SURFACE_TO_SURFACE_OFFSET
c 2	TIED_SURFACE_TO_SURFACE_CONSTRAINED_OFFSET

Contact Examples:

```
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$  *CONTACT_NODES_TO_SURFACE
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Make a simple contact that prevents the nodes in part 2 from
$  penetrating the segments in part 3.
$
*CONTACT_NODES_TO_SURFACE
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$     ssid      msid     sstyp     mstyp    sboxid    mboxid       spr       mpr
         2         3         3         3
$
$       fs        fd        dc        vc       vdc    penchk        bt        dt

$
$      sfs       sfm       sst       mst      sfst      sfmt       fsf       vsf

$
$    sstype, mstype = 3  id's specified in ssid and msid are parts
$      ssid = 2  use slave nodes in part 2
$      msid = 3  use master segments in part 3
$
$  Use defaults for all parameters.
$
$$$$  Optional Cards A and B not specified (default values will be used).
$
$
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$  *CONTACT_SINGLE_SURFACE
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Create a single surface contact between four parts: 28, 97, 88 and 92
$     - create a part set with set ID = 5, list the four parts
$     - in the *CONTACT_SINGLE_SURFACE definition specify:
$           sstyp = 2  which means the value for ssid is a part set
$           ssid  = 5  use part set 5 for defining the contact surfaces
$
$  Additional contact specifications described below.
$
*CONTACT_SINGLE_SURFACE
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$     ssid      msid     sstyp     mstyp    sboxid    mboxid       spr       mpr
         5                   2
$       fs        fd        dc        vc       vdc    penchk        bt        dt
      0.08      0.05        10                  20                            40.0
$      sfs       sfm       sst       mst      sfst      sfmt       fsf       vsf

$
$       fs = 0.08  static coefficient of friction equals 0.08
$       fd = 0.05  dynamic coefficient of friction equals 0.05
$       dc =   10  exponential decay coefficient, helps specify the transition
$                     from a static slide to a very dynamic slide
$      vdc =   20  viscous damping of 20% critical (damps out nodal
$                     oscillations due to the contact)
$       dt = 40.0  contact will deactivate at 40 ms (assuming time unit is ms)
$
$$$$  Optional Cards A and B not specified (default values will be used).
$
$
*SET_PART_LIST
$      sid
         5
$     pid1      pid2      pid3      pid4
        28        97        88        92
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$  *CONTACT_DRAWBEAD
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Define a draw bead contact:
$    - the draw bead is to be made from the nodes specified in node set 2
$    - the master segments are to be those found in the box defined by box 2
$        that are in part 18
$    - include slave and master forces in interface file (spr, mpr = 1)
$
*CONTACT_DRAWBEAD
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$     ssid      msid     sstyp     mstyp    sboxid    mboxid       spr       mpr
         2        18         4         3                   2         1         1
$
$       fs        fd        dc        vc       vdc    penchk        bt        dt
      0.10
$
$      sfs       sfm       sst       mst      sfst      sfmt       fsf       vsf

$
$$$$ Card 4 required because it's a drawbead contact
$
$  lcdidrf    lcidnf     dbdth     dfscl    numint
         3             0.17436       2.0
$
$  lcdidrf =       3  load curve 3 specifies the bending component of the
$                      restraining force per unit draw bead length
$    dbdth = 0.17436  draw bead depth
$    dfscl =     2.0  scale load curve 3 (lcdidrf) by 2
$
$$$$  Optional Cards A and B not specified (default values will be used).
$
*DEFINE_BOX
$    boxid       xmm       xmx       ymn       ymx       zmn       zmx
         2 0.000E+00 6.000E+00 6.000E+00 1.000E+02-1.000E+03 1.000E+03
$
*SET_NODE_LIST
$      sid       da1       da2       da3       da4
         2
$     nid1      nid2      nid3      nid4      nid5      nid6      nid7      nid8
      2580      2581      2582      2583      2584      2585      2586      2587
      2588      2589      2590
$
*DEFINE_CURVE
$     lcid      sidr      scla      sclo      offa      offo
         3
$                 a                   o
$             DEPTH           FORC/LGTH
```

```
          0.000E+00           0.000E+00
          1.200E-01           1.300E+02
          1.500E-01           2.000E+02
          1.800E-01           5.000E+02
```

*CONTACT_ADD_WEAR

This card associates a wear model to a contact interface for post-processing wear quantities.

This card does not affect the results of a simulation. Wear is associated to friction so the frictional coefficient *must* be nonzero for the associated contact interface. This feature calculates the wear depth, sliding distance and possibly user defined wear history variables according to the specified model and writes it to the intfor database for post-processing. Note that this data is *not* written unless the parameter NWEAR and/or NWUSR are set on the *DATABASE_EXTENT_INTFOR card. *H*-adaptive remeshing is supported with this feature. Implicit analysis is supported, for which mortar is the preferred contact.

Card 1	1	2	3	4	5	6	7	8
Variable	CID	WTYPE	P1	P2	P3	P4	P5	P6
Type	I	I	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

User Defined Wear Parameter Cards. Define as many cards as needed to define P1 parameters if and only if WTYPE.LT.0.

Card n	1	2	3	4	5	6	7	8
Variable	W1	W2	W3	W4	W5	W6	W7	W8
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
CID	Contact interface ID, see *CONTACT_…

VARIABLE **DESCRIPTION**

WTYPE

Wear law

LT.0: User defined wear law, value specifies type used in subroutine.

EQ.0: Archard's wear law.

P1

First wear parameter

WTYPE.EQ.0: Dimensionless scale factor k. If negative the absolute value specifies a table ID with $k = k(p, \dot{d})$ as a function of contact pressure $p \geq 0$ and relative sliding velocity $\dot{d} \geq 0$.

WTYPE.LT.0: Number of user wear parameters for this interface.

P2

Second wear parameter

WTYPE.EQ.0: Slave surface hardness parameter H_s. If negative the absolute value specifies a curve ID with $H_s = H_s(T_s)$ as function of slave node temperature T_s.

WTYPE.LT.0: Number of user wear history variables per contact node, these can be output to the intfor file, see NWUSR on *DATABASE_EXTENT_INTFOR.

P3

Third wear parameter

WTYPE.EQ.0: Master surface hardness parameter H_m. If negative the absolute value specifies a curve ID with $H_m = H_m(T_m)$ as function of master node temperature T_m.

WTYPE.LT.0: Not used.

P4 - P6

Not used.

WN

Nth user defined wear parameter.

Remarks:

Archard's wear law (WTYPE.EQ.0) states that the wear depth w at a contact point evolves with time as

$$\dot{w} = k\frac{p\dot{d}}{H}$$

where $k > 0$ is a dimensionless scale factor, $p \geq 0$ is the contact interface pressure, $\dot{d} \geq 0$ is the relative sliding velocity of the points in contact and $H > 0$ is the surface hardness (force per area). The wear depth for a node in contact is incremented in accordance with this formula, accounting for different hardness of the slave and master side, H_s and H_m, respectively. By using negative numbers for wear parameters P1, P2 or P3, the corresponding parameter is defined by a table or a curve. For P1, the value of k is taken from a table with contact pressure p and sliding velocity $\dot{d}$ as arguments, while for P2 or P3, the corresponding hardness H is taken from curves with the associated contact nodal temperature T as argument. That is, the slave side hardness will be a function of the slave side temperature, and vice versa.

Customized wear laws may be specified as a user-defined subroutine called userwear. This subroutine is called when WTYPE < 0. This subroutine is passed wear parameters (see P1) for this interface as well as number of wear history variables (see P2) per contact node. The wear parameters are defined on additional cards (see WN) and the history variables are updated in the user subroutine. The history variables can be output to the intfor file, see NWUSR on *DATABASE_EXTENT_INTFOR. WTYPE may be used to distinguish between different wear laws, and consequently any number of different laws can be implemented within the same subroutine. For more information, we refer to the source code which contains extensive commentaries and two sample wear laws.

Only one wear law per contact interface can be specified. The procedure for activating this feature involves

1. Using the present keyword to associate wear to a contact interface
2. Setting NWEAR and/or NWUSR on the *DATABASE_EXTENT_INTFOR card.
3. Having a contact interface with friction of a type that is supported.. If SOFT = 2 on optional card A of the contact data, then any valid keyword option is supported. If SOFT = 0 or SOFT = 1, then the following list is supported.

*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE

*CONTACT_FORMING_SURFACE_TO_SURFACE

*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE

*CONTACT_FORMING_SURFACE_TO_SURFACE_MORTAR

*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_MORTAR

*CONTACT_AUTOMATIC_SINGLE_SURFACE_MORTAR

Limitations:

1. _SMOOTH option is not supported

2. MPP “groupable” option is not supported

See also *DATABASE_EXTENT_INTFOR for general guidelines related to the intfor database.

*CONTACT_AUTO_MOVE

Purpose: This feature allows for automatic move of a master surface in a contact definition to close an unspecified gap between a slave and the master surface. The gap may be caused as a result of an initial gravity loading on the slave part. The gap will be closed on a specified time to save CPU time. The master surface in metal forming application will typically be the upper cavity and the slave part will be the blank. This feature is applicable only for sheet metal forming application.

Cards 1	1	2	3	4	5	6	7	8
Variable	ID	CONTID	VID	LCID	ATIME	OFFSET		
Type	I	I	I	I	F	F		
Default	none	none	none	0	0.0	0.0		

VARIABLE	DESCRIPTION
ID	Move ID for this automatic move input. GT.0: velocity controlled tool kinematics (the variable VAD = 0 in *BOUNDARY_PRESCRIBED_MOTION_RIGID) LT.0: displacement controlled tool kinematics (VAD = 2)
CONTID	Contact ID, as in *CONTACT_FORMING_...._ID, which defines the slave and master part set IDs.
VID	Vector ID of a vector oriented in the direction of movement of the master surface, as in *DEFINE_VECTOR. The origin of the vector is unimportant since the direction cosines of the vector are computed and used.
LCID	Load curve defining tooling kinematics, either by velocity versus time or by displacement versus time. This load curve will be adjusted automatically during a simulation to close the empty tool travel.
ATIME	Activation time defining the moment the master surface (tool) to be moved.

VARIABLE	DESCRIPTION
OFFSET	Time at which a master surface will move to close a gap distance, which may happen following the move of another master surface. This is useful in sequential multiple flanging or press hemming simulation. Simulation time (CPU) is much faster based on the shortened tool travel (no change to the termination time).

Example: gravity loading and closing with implicit static

Referring to the partial input deck below and Figure 11-18, a combined simulation of gravity loading and binder closing of a fender outer is demonstrated on the NUMISHEET 2002 benchmark. In this multi-step implicit static set up, the blank is allocated 0.3 “time” units (3 implicit steps for DT0 = 0.1) to be loaded with gravity. At the end of gravity loading, a gap of 12mm was created between the upper die and the blank, Figure 11-19. The upper die is set to be moved at 0.3 “time” units, closing the gap caused by the gravity effect on the blank (Figure 11-20 left). An intermediate closing state is shown at t = 0.743 (Figure 11-20 right) while the final completed closing is shown in Figure 11-21. It is noted that the upper die is controlled with displacement (VAD = 2) in a shape of a right triangular in the displacement versus “time” space as defined by load curve #201, and the ID in *CONTACT_AUTO_MOVE is set to “-1”.

```
*PARAMETER
R grvtime       0.3
R endtime       1.0
R diemv      145.45
*CONTROL_TERMINATION
&endtime
*CONTROL_IMPLICIT_FORMING
2,2,100
*CONTROL_IMPLICIT_GENERAL
$   IMFLAG       DT0
         1      0.10
*CONTROL_ACCURACY
         1        2
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ID
11
....
....
....
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*BOUNDARY_PRESCRIBED_MOTION_RIGID
$#     pid       dof       vad      lcid        sf       vid     death     birth
         2         3         2       201 -1.000000         0       0.0     0.000
*CONTACT_AUTO_MOVE
$       ID   ContID       VID      LCID     ATIME
        -1        11        89       201  &grvtime
*DEFINE_VECTOR
89,0.0,0.0,0.0,0.0,0.0,-10.0
*DEFINE_CURVE
201
0.0,0.0
&grvtime,0.0
1.0,&diemv
```

Similarly, “velocity” controlled tool kinematics is also enabled. In the example keyword below, the “velocity” profile is ramped up initially and then kept constant. It is noted that the variable VAD in *BOUNDARY is set to “0”, and ID in *CONTACT_AUTO_MOVE is set to positive “1” indicating it is a velocity boundary condition.

```
*PARAMETER
R grvtime       0.3
R tramp       0.001
R diemv      145.45
R clsv       1000.0
*PARAMETER_EXPRESSION
R tramp1  tramp+gravtime
R endtime tramp1+(abs(diemv)-0.5*clsv*tramp)/clsv
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ID
11
....
....
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*BOUNDARY_PRESCRIBED_MOTION_RIGID
$#     pid       dof       vad      lcid        sf       vid     death     birth
         2         3         0       201 -1.000000         0       0.0     0.000
*CONTACT_AUTO_MOVE
$       ID    ContID       VID      LCID     ATIME
         1        11        89       201  &grvtime
*DEFINE_VECTOR
89,0.0,0.0,0.0,0.0,0.0,-10.0
*DEFINE_CURVE
201
0.0,0.0
0.2,0.0
&tramp1,&clsv
&endtime,&clsv
```

Example: tool delay in sequential flanging process with explicit dynamic:

The following example demonstrates the use of the variable OFFSET. As shown in Figure 11-22 (left), a total of 5 flange steels are auto-positioned initially according to the initial blank shape. Upon closing of the pressure pad, a first set of 4 flanging steels move to home completing the first stage of the stamping process (Figure 11-22 right).

The gap created by the completion of the first flanging process is closed automatically at a time defined using variables ATIME/OFFSET (Figure 11-23 left). During the second stage of the process, flanging steel *&flg5pid* moves to home completing the final flanging (Figure 11-23 right). An excerpt from the input deck for this model can be found below. This deck was created using LS-PrePost’s eZ-Setup feature (http://ftp.lstc.com/anonymous/outgoing/lsprepost/4.0/metalforming/), with two additional keywords added:*CONTACT_AUTO_MOVE and *DEFINE_VECTOR.

Flanging steel #5 is set to move in a cam angle defined by vector #7 following the completion of the flanging (straight down) process of flanging steel #2. The variables ATIME and OFFSET in *CONTACT_AUTO_MOVE are both defined as &endtime4, which is calculated based on the automatic positioning of tools/blank using *CONTROL_FORMING_AUTOPOSITION. At defined time, flanging steel #5 ‘jumps’ into position to where it just comes into contact with the partially formed down-standing flange, saving some CPU times (Figure 11-23 left). Flanging steel #5 continues to move to its home position completing the simulation (Figure 11-23 right). The CPU time savings is 27% in this case.

```
*KEYWORD
*PARAMETER
...
*PART

  &flg5pid  &flg5sec  &flg5mid
...
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$ Local coordinate system for flanging steel #5 move direction
*DEFINE_COORDINATE_SYSTEM
$#       cid        xo        yo        zo        xl        yl        zl
  &flg5cid  -5.09548   27.6584  -8.98238  -5.43587   26.8608  -9.48034
$#        xp        yp        zp
  -5.82509   27.5484  -8.30742
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$ Auto positioning
*CONTROL_FORMING_AUTOPOSITION_PARAMETER_SET
$       SID       CID       DIR      MPID  POSITION   PREMOVE     THICK  PARORDER
...
  &flg5sid  &flg5cid         3  &blk1sid        -1             &bthick    flg5mv
*PART_MOVE
$    PID             XMOV             YMOV             ZMOV      CID    IFSET
&flg5sid              0.0              0.0         &flg5mv&flg5cid       1
...
*MAT_RIGID
$      MID        RO         E        PR         N    COUPLE         M     ALIAS
  &flg5mid 7.830E-09 2.070E+05      0.28
$      CMO      CON1      CON2
        -1  &flg5cid    110111
$LCO or A1        A2        A3        V1        V2        V3
  &flg5cid
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*CONTACT_AUTO_MOVE
$       ID    CONTID       VID      LCID     ATIME    OFFSET
         1         7         7        10  &endtim4  &endtim4
*DEFINE_VECTOR
$      VID        XT        YT        ZT        XH        YH        ZH
         7       0.0       0.0       0.0-0.5931240 0.5930674-0.5444952
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ID
$      CID
         7
$     SSID      MSID     SSTYP     MSTYP    SBOXID    MBOXID       SPR       MPR
  &blk1sid  &flg5sid         2         2                             1         1
$       FS        FD        DC        VC       VDC    PENCHK        BT        DT
...
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$                    Tool kinematics
$  -------------------------closing
*BOUNDARY_PRESCRIBED_MOTION_RIGID_local
...
  &flg5pid         3         0         4       1.0         0  &endtim4
$  -------------------------flanging
*BOUNDARY_PRESCRIBED_MOTION_RIGID_local
...
  &flg5pid         3         0        10       1.0         0            &endtim4
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*END
```

Revision information:

This feature is implemented in LS-DYNA Revision 64066 and later releases. The variable OFFSET is in Revision 77137 and later releases.

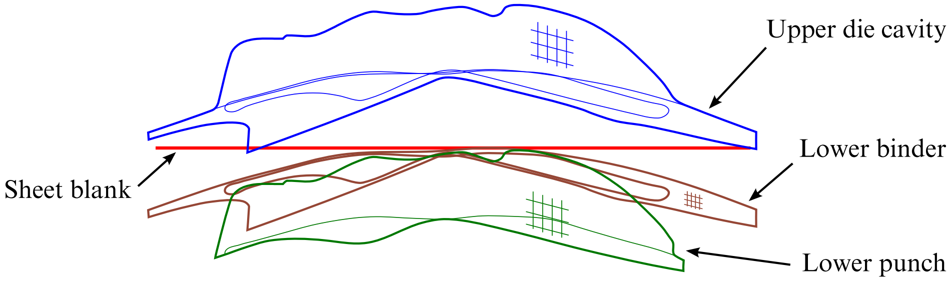


Figure 11-18. Initial parts auto-positioned at t = 0.0.

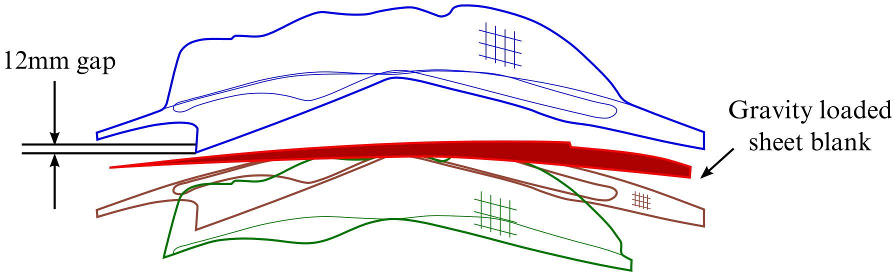


Figure 11-19. Gravity loading on blank at t = 0.2.

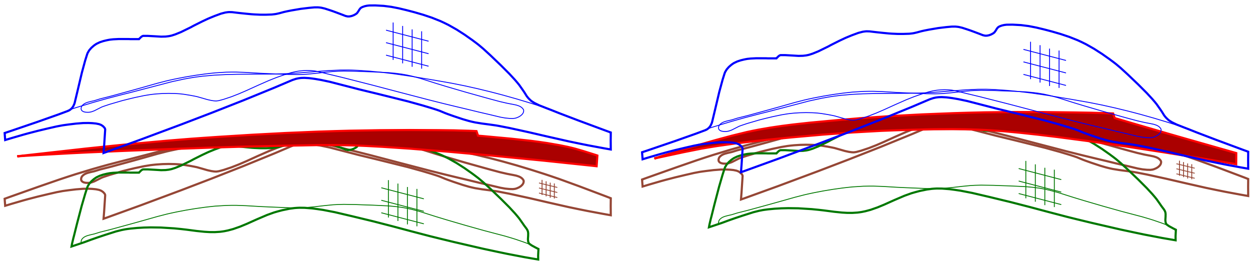


Figure 11-20. Upper die move down at t = 0.3 closing the gap (left); continue closing at t = 0.743 (right).

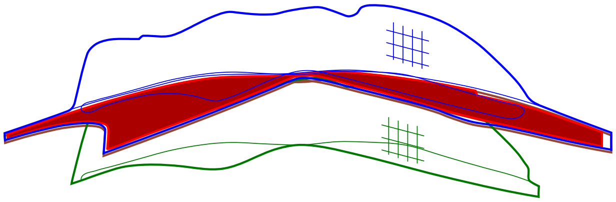


Figure 11-21. Closing complete at t = 1.0.

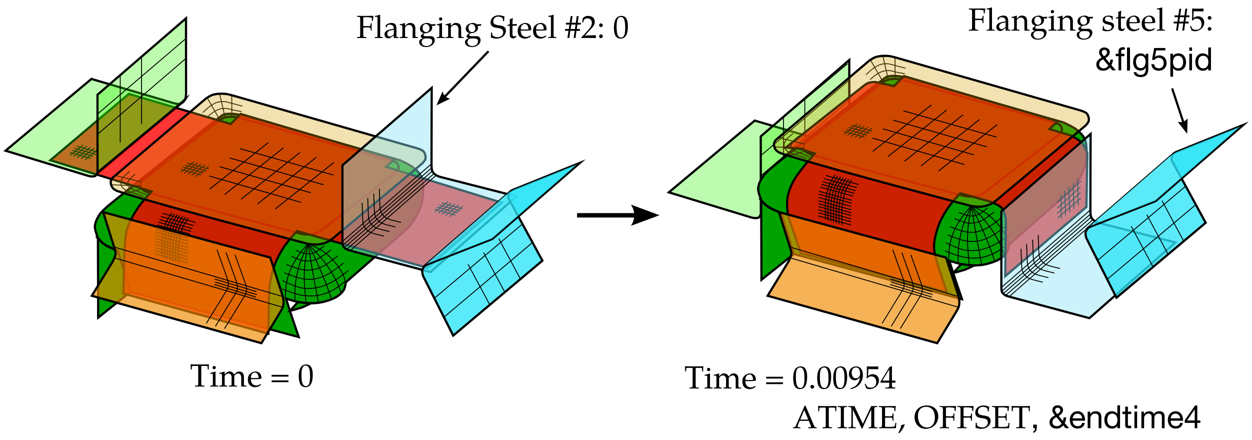


Figure 11-22. A sequential flanging process (left); first set of flanging steels reaching home (right).

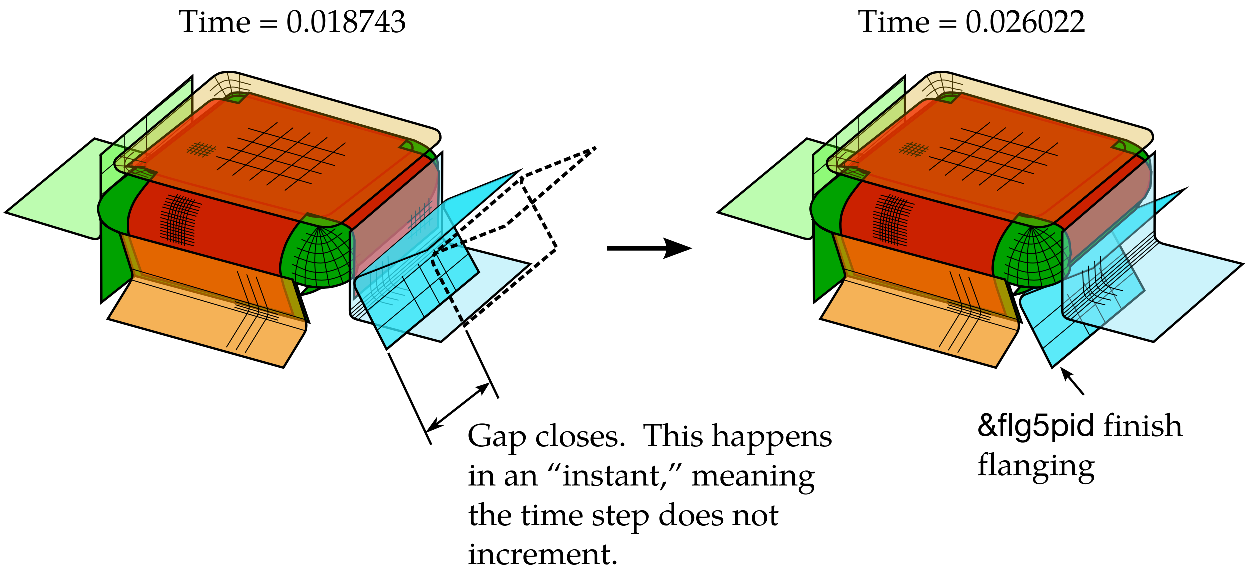

Time = 0.018743
Time = 0.026022
Gap closes. This happens in an "instant," meaning the time step does not increment.
&flg5pid finish flanging

Figure 11-23. Closing the empty travel (left); flanging steel &flg5pid completes flanging process (right).

*CONTACT_COUPLING

Purpose: Define a coupling surface for MADYMO to couple LS-DYNA with deformable and rigid parts within MADYMO. In this interface, MADYMO computes the contact forces acting on the coupling surface, and LS-DYNA uses these forces in the update of the motion of the coupling surface for the next time step. Contact coupling can be used with other coupling options in LS-DYNA.

Card 1	1	2	3	4	5	6	7	8
Variable	ID							
Type	I							
Default	required							

Set Cards. Include on card for each coupled set. The next "*" card terminates this input.

Card 2	1	2	3	4	5	6	7	8
Variable	SID	STYPE						
Type	I	I						
Default	required	0						

VARIABLE	DESCRIPTION
SID	Set ID for coupling. See Remark 1 below.
STYPE	Set type: EQ.0: part set EQ.1: shell element set EQ.2: solid element set EQ.3: thick shell element set

Remarks:

1. Only one coupling surface can be defined. If additional surfaces are defined, the coupling information will be added to the first definition.

2. The units and orientation can be converted by using the CONTROL_COUPLING keyword. It is not necessary to use the same system of units in MADYMO and in LS-DYNA if unit conversion factors are defined.

*CONTACT_ENTITY

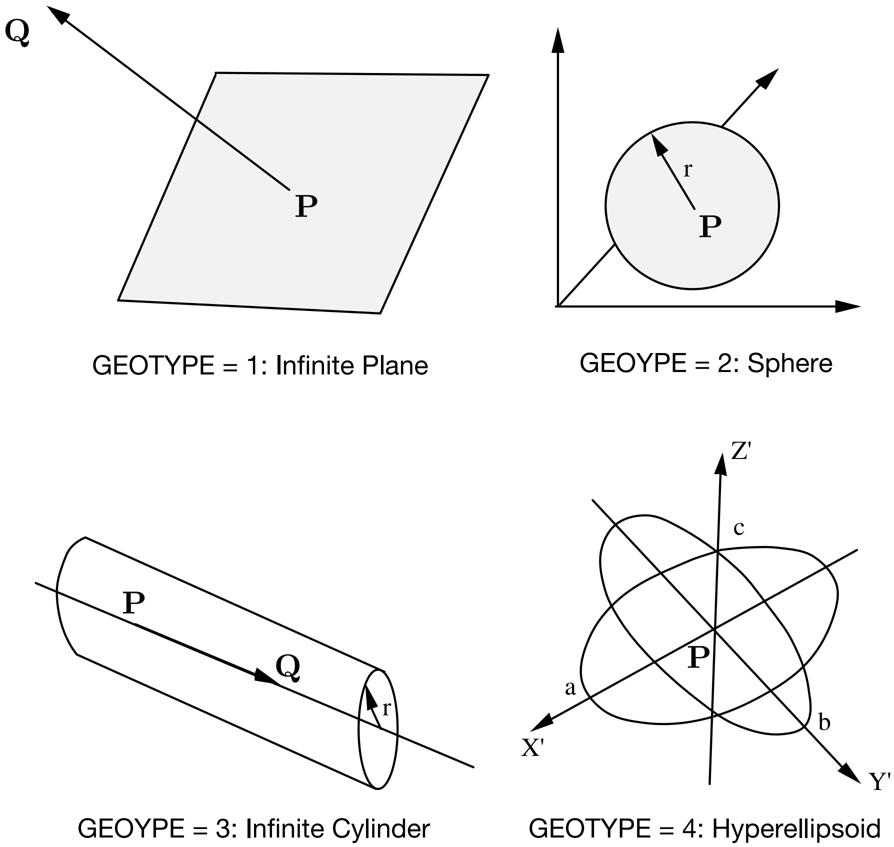
Purpose: Define a contact entity. Geometric contact entities treat the impact between a deformable body defined as a set of slave nodes or nodes in a shell part set and a rigid body. The shape of the rigid body is determined by attaching geometric entities. Contact is treated between these geometric entities and the slave nodes using a penalty formulation. The penalty stiffness is optionally maximized within the constraint of the Courant criterion. As an alternative, a finite element mesh made with shells can be used as geometric entity. Also, axisymmetric entities with arbitrary shape made with multi-linear polygons are possible. The latter is particularly useful for metalforming simulations.

WARNING: If the problem being simulated involves dynamic motion of the entity, care should be taken to insure that the inertial properties of the entity are correct. It may be necessary to use the *PART_INERTIA option to specify these properties.

The data set for *CONTACT_ENTITY consists of 5 cards:

Card 1	1	2	3	4	5	6	7	8
Variable	PID	GEOTYP	SSID	SSTYP	SF	DF	CF	INTORD
Type	I	I	I	I	F	F	F	I
Default	required	required	required	0	1.	0.	0.	0

VARIABLE	DESCRIPTION
PID	Part ID of the rigid body to which the geometric entity is attached, see *PART.



GEOTYPE = 1: Infinite Plane

GEOYPE = 2: Sphere

GEOYPE = 3: Infinite Cylinder

GEOTYPE = 4: Hyperellipsoid

Figure 11-24. Contact Entities.

VARIABLE **DESCRIPTION**

GEOTYP Type of geometric entity:

EQ.1: plane,

EQ.2: sphere,

EQ.3: cylinder,

EQ.4: ellipsoid,

EQ.5: torus,

EQ.6: CAL3D/MADYMO Plane, see Appendix I,

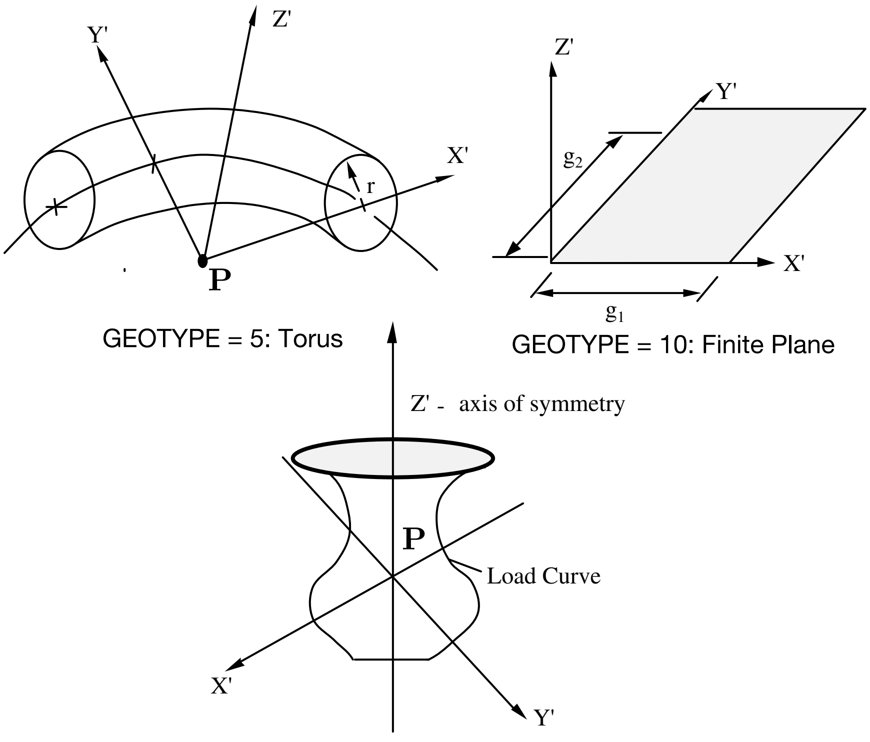
EQ.7: CAL3D/MADYMO Ellipsoid, see Appendix I,

EQ.8: VDA surface, see Appendix L,

EQ.9: rigid body finite element mesh (shells only),

EQ.10: finite plane,

EQ.11: load curve defining line as surface profile of axisymmetric rigid bodies.



GEOTYPE = 5: Torus

GEOTYPE = 10: Finite Plane

GEOTYPE = 11: Load Curve

Figure 11-25. More contact entities.

VARIABLE **DESCRIPTION**

SSID Slave set ID, see *SET_NODE_*OPTION*, *PART, or *SET_PART.

SSTYP Slave set type:

EQ.0: node set,

EQ.1: part ID,

EQ.2: part set ID.

SF Penalty scale factor. Useful to scale maximized penalty.

DF Damping option, see description for *CONTACT_*OPTION*:

EQ.0: no damping,

GT.0: viscous damping in percent of critical, e.g., 20 for 20% damping,

LT.0: DF must be a negative integer. -DF is the load curve ID giving the damping force versus relative normal velocity (see remark 1 below).

VARIABLE	DESCRIPTION
CF	Coulomb friction coefficient. See remark 2 below. EQ.0: no friction GT.0: constant friction coefficient LT.0: CF must be a negative integer. -CF is the load curve ID giving the friction coefficient versus time.
INTORD	Integration order (slaved materials only). This option is not available with entity types 8 and 9 where only nodes are checked: EQ.0: check nodes only, EQ.1: 1 point integration over segments, EQ.2: 2 × 2 integration, EQ.3: 3 × 3 integration, EQ.4: 4 × 4 integration, EQ.5: 5 × 5 integration. This option allows a check of the penetration of the rigid body into the deformable (slaved) material. Then virtual nodes at the location of the integration points are checked.

Remarks:

1. The optional load curves that are defined for damping versus relative normal velocity and for force versus normal penetration should be defined in the positive quadrant. The sign for the damping force depends on the direction of the relative velocity and the treatment is symmetric if the damping curve is in the positive quadrant. If the damping force is defined in the negative and positive quadrants, the sign of the relative velocity is used in the table look-up.

2. If at any time the friction coefficient is >= 1.0, the force calculation is modified to a constraint like formulation which allows no sliding. This is only recommended for entities with constrained motion since the mass of the entity is assumed to be infinite.

Card 2	1	2	3	4	5	6	7	8
Variable	BT	DT	SO	GO	ITHK	SPR		
Type	F	F	I	I	I	I		
Default	0.	1.E+20	0	0	0	0		

VARIABLE	DESCRIPTION

BT
Birth time

DT
Death time

SO
Flag to use penalty stiffness as in surface-to-surface contact:

EQ.0: contact entity stiffness formulation,

EQ.1: surface to surface contact method,

EQ.2: normal force is computed via a constraint-like method. The contact entity is considered to be infinitely massive, so this is recommended only for entities with constrained motion.

LT.0: SO must be an integer: -SO is the load curve ID giving the force versus the normal penetration.

GO
Flag for automatic meshing of the contact entity for entity types 1-5 and 10-11. GO = 1 creates null shells for visualization of the contact entity. Note these shells have mass and will affect the mass properties of the rigid body PID unless *PART_INERTIA is used for the rigid body.

EQ.0: mesh is not generated,

EQ.1: mesh is generated.

ITHK
Flag for considering thickness for shell slave nodes (applies only to entity types 1, 2, 3; SSTYP must be set to zero).

EQ.0: shell thickness is not considered,

EQ.1: shell thickness is considered,

VARIABLE	DESCRIPTION
SPR	Include the slave side in *DATABASE_BINARY_INTFOR interface force files, valid only when SSTYP > 0: EQ.1: slave side forces included.

Card 3	1	2	3	4	5	6	7	8
Variable	XC	YC	ZC	AX	AY	AZ		
Type	F	F	F	F	F	F		
Default	0.	0.	0.	0.	0.	0		

Card 4	1	2	3	4	5	6	7	8
Variable	BX	BY	BZ					
Type	F	F	F					
Default	0.	0.	0.					

VARIABLE	DESCRIPTION
XC	*x*-center, x_c, see remarks below.
YC	*y*-center, y_c, see remarks below.
ZC	*z*-center, z_c. See remarks below.
AX	*x*-direction for local axis **A**, A_x, see remarks below.
AY	y-direction for local axis **A**, A_y, see remarks below.
AZ	z-direction for local axis **A**, A_z, see remarks below.
BX	*x*-direction for local axis **B**, $\mathrm{B_X}$, see remarks below.
BY	*y*-direction for local axis **B**, B_y, see remarks below.

VARIABLE	DESCRIPTION
BZ	z-direction for local axis **B**, B_z, see remarks below.

Remarks::

1. The coordinates, (x_c, y_c, z_c) are the positions of the local origin of the geometric entity in global coordinates. The entity's local A-axis is determined by the vector (A_x, A_y, A_z) and the local B-axis by the vector (B_x, B_y, B_z).

2. Cards 3 and 4 define a local to global transformation. The geometric contact entities are defined in a local system and transformed into the global system. For the ellipsoid, this is necessary because it has a restricted definition for the local position. For the plane, sphere, and cylinder, the entities can be defined in the global system and the transformation becomes $(x_c, y_c, z_c) = (0,0,0)$, $(A_x, A_y, A_z) = (1,0,0)$, and $(B_x, B_y, B_z) = (0,1,0)$.

Card 5	1	2	3	4	5	6	7	8
Variable	INOUT	G1	G2	G3	G4	G5	G6	G7
Type	I	F	F	F	F	F	F	F
Default	0	0.	0.	0.	0.	0.	0.	0.

VARIABLE	DESCRIPTION
INOUT	In-out flag. Allows contact from the inside or the outside (default) of the entity: EQ.0: slave nodes exist outside of the entity, EQ.1: slave nodes exist inside the entity.
G1	Entity coefficient g_1 (CAL3D/MADYMO plane or ellipse number) for coupled analysis (see Appendix I).
G2	Entity coefficient g_2, see remarks below.
G3	Entity coefficient g_3, see remarks below.
G4	Entity coefficient g_4, see remarks below.
G5	Entity coefficient g_5, see remarks below.

VARIABLE	DESCRIPTION
G6	Entity coefficient g_6, see remarks below.
G7	Entity coefficient g_7, see remarks below.

Remarks:

Figures 11-24 and 11-25 show the definitions of the geometric contact entities. The relationships between the entity coefficients and the Figure 11-25 and 11-24 variables are as described below. Note that (P_x, P_y, P_z) defines a point and $\left(Q_x, Q_y, Q_z\right)$ is a direction vector.

GEOTYP = 1

$$\begin{aligned} g_1 &= P_x \qquad & g_4 &= Q_x \\ g_2 &= P_y & g_5 &= Q_y \\ g_3 &= P_z & g_6 &= Q_z \\ & & g_7 &= L \end{aligned}$$

If automatic generation is used, a square plane of length L on each edge is generated which represents the infinite plane. If generation is inactive, then g7 may be ignored.

GEOTYP = 2

$$\begin{aligned} g_1 &= P_x \qquad & g_4 &= r \\ g_2 &= P_y \\ g_3 &= P_z \end{aligned}$$

GEOTYP = 3

$$\begin{aligned} g_1 &= P_x \qquad & g_4 &= Q_X \\ g_2 &= P_y & g_5 &= Q_y \\ g_3 &= P_z & g_6 &= Q_z \\ & & g_7 &= r \end{aligned}$$

If automatic generation is used, a cylinder of length $\sqrt{Q_{\mathrm{x}}^2 + Q_y^2 + Q_{\mathrm{z}}^2}$ and radius r is generated which represents the infinite cylinder.

GEOTYP = 4

$$\begin{aligned} g_1 &= P_x \qquad & g_4 &= a \\ g_2 &= P_y & g_5 &= b \\ g_3 &= P_z & g_6 &= c \\ & & g_7 &= n \text{ (order of the ellipsoid)} \end{aligned}$$

GEOTYP = 5

g_1 = Radius of torus
g_2 = r
g_3 = number of elements along minor circumference
g_4 = number of elements along major circumference

GEOTYP = 8

g_1 = Blank thickness (option to override true thickness)
g_2 = Scale factor for true thickness (optional)
g_3 = Load curve ID defining thickness versus time. (optional)

GEOTYP = 9

g_1 = Shell thickness (option to override true thickness).

NOTE: The shell thickness specification is necessary if the slave surface is generated from solid elements.

g_2 = Scale factor for true thickness (optional)
g_3 = Load curve ID defining thickness versus time. (optional)

GEOTYP = 10

g_1 = Length of edge along X′ axis
g_2 = Length of edge along Y′ axis

GEOTYP = 11

g_1 = Load curve ID defining axisymmetric surface profile about Z-axis. Load curves defined by the keywords *DEFINE_CURVE or *DEFINE_CURVE_ENTITY can be used.

g_2 = Number of elements along circumference

EQ.0: default set to 10

g_3 = Number of elements along axis

EQ.0: default set to 20
EQ.-1: the elements generated from points on the load curve

g_4 = Number of sub divisions on load curve used to calculate contact

EQ.0: default set to 1000

CONTACT_GEBOD_*OPTION

Purpose: Define contact interaction between the segment of a GEBOD dummy and parts or nodes of the finite element model. This implementation follows that of the contact entity, however, it is specialized for the dummies. Forces may be output using the *DATABASE_GCEOUT command. See *COMPONENT_GEBOD and Appendix N for further details.

Conventional *CONTACT_*OPTION* treatment (surface-to-surface, nodes-to-surface, etc.) can also be applied to the segments of a dummy. To use this approach it is first necessary to determine part ID assignments by running the model through LS-DYNA's initialization phase.

The following options are available and refer to the ellipsoids which comprise the dummy. Options involving **HAND** are not applicable for the child dummy since its lower arm and hand share a common ellipsoid.

LOWER_TORSO

MIDDLE_TORSO

UPPER_TORSO

NECK

HEAD

LEFT_SHOULDER

RIGHT_SHOULDER

LEFT_UPPER_ARM

RIGHT_UPPER_ARM

LEFT_LOWER_ARM

RIGHT_LOWER_ARM

LEFT_HAND

RIGHT_HAND

LEFT_UPPER_LEG

RIGHT_UPPER_LEG

LEFT_LOWER_LEG

RIGHT_LOWER_LEG

LEFT_FOOT

RIGHT_FOOT

Card 1	1	2	3	4	5	6	7	8
Variable	DID	SSID	SSTYP	SF	DF	CF	INTORD	
Type	I	I	I	F	F	F	I	
Default	required	required	required	1.	20.	0.5	0	

VARIABLE **DESCRIPTION**

DID Dummy ID, see *COMPONENT_GEBOD_*OPTION.*

SSID Slave set ID, see *SET_NODE_*OPTION*, *PART, or *SET_PART.

SSTYP Slave set type:

EQ.0: node set,

EQ.1: part ID,

EQ.2: part set ID.

SF Penalty scale factor. Useful to scale maximized penalty.

DF Damping option, see description for *CONTACT_*OPTION*:

EQ.0: no damping,

GT.0: viscous damping in percent of critical, e.g., 20 for 20% damping,

LT.0: DF must be an integer. -DF is the load curve ID giving the damping force versus relative normal velocity (see Remark 1 below).

CF Coulomb friction coefficient (see Remark 2 below). Assumed to be constant.

VARIABLE	DESCRIPTION
INTORD	Integration order (slaved materials only). EQ.0: check nodes only, EQ.1: 1 point integration over segments, EQ.2: 2 × 2 integration, EQ.3: 3 × 3 integration, EQ.4: 4 × 4 integration, EQ.5: 5 × 5 integration. This option allows a check of the penetration of the dummy segment into the deformable (slaved) material. Then virtual nodes at the location of the integration points are checked.

Card 2	1	2	3	4	5	6	7	8
Variable	BT	DT	SO					
Type	F	F	I					
Default	0.	1.E+20	0					

VARIABLE	DESCRIPTION
BT	Birth time
DT	Death time
SO	Flag to use penalty stiffness as in surface-to-surface contact: EQ.0: contact entity stiffness formulation, EQ.1: surface to surface contact method, LT.0: In this case SO must be an integer. $\|SO\|$ gives the load curve ID giving the force versus the normal penetration.

Remarks:

1. The optional load curves that are defined for damping versus relative normal velocity and for force versus normal penetration should be defined in the positive quadrant. The sign for the damping force depends on the direction of the

relative velocity and the treatment is symmetric if the damping curve is in the positive quadrant. If the damping force is defined in the negative and positive quadrants, the sign of the relative velocity is used in the table look-up.

2. Insofar as these ellipsoidal contact surfaces are continuous and smooth it may be necessary to specify Coulomb friction values larger than those typically used with faceted contact surfaces.

CONTACT_GUIDED_CABLE_*{OPTION1}_{OPTION2}

Purpose: Define a sliding contact that guides 1D elements, such as springs, trusses, and beams, along a path defined by a set of nodes. Only one 1D element can be in contact with any given node in the node set at a given time. If for some reason, a node is in contact with multiple 1D elements, one guided contact definition must be used for each contact. The ordering of the nodal points and 1D elements in the input is arbitrary.

OPTION1 specifies that a part set ID is given with the single option:

<BLANK>

SET

If not used a part ID is assumed.

OPTION2 specifies that the first card to read defines the heading and ID number of the contact interface and takes the single option:

ID

Title Card. Additional card for ID keyword option.

Title	1	2	3	4	5	6	7	8
Variable	CID	HEADING						
Type	I	A70						

VARIABLE	DESCRIPTION
CID	Contact interface ID. This must be a unique number.
HEADING	Interface descriptor. It is suggested that unique descriptions be used.

Card 1	1	2	3	4	5	6	7	8
Variable	NSID	PID/PSID	SOFT	SSFAC	FRIC			
Type	I	I	I	F	F			
Default	none	none	0	1.0	none			

VARIABLE **DESCRIPTION**

NSID Node set ID that guides the 1D elements.

PID/PSID Part ID or part set ID if SET is included in the keyword line.

SOFT Flag for soft constraint option. Set to 1 for soft constraint.

SSFAC Stiffness scale factor for penalty stiffness value. The default value is unity. This applies to SOFT set to 0 and 1.

FRIC Contact friction.

*CONTACT_INTERIOR

Purpose: Define interior contact for solid elements. Frequently, when soft materials are compressed under high pressure, the solid elements used to discretize these materials may invert leading to negative volumes and error terminations. In order to keep these elements from inverting, it is possible to consider interior contacts between layers of interior surfaces made up of the faces of the solid elements. Since these interior surfaces are generated automatically, the part (material) ID's for the materials of interest are defined here, prior to the interface definitions.

Define as many cards as necessary. Input ends at the next * card. Multiple instances of this keyword may appear in the input.

	1	2	3	4	5	6	7	8
Variable	PSID1	PSID2	PSID3	PSID4	PSID5	PSID6	PSID7	PSID8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
PSID*	Part set ID for which interior contact is desired.

Four attributes should be defined for each part set:

Attribute 1: PSF, penalty scale factor (Default = 1.00).

Attribute 2: Activation factor, F_a (Default = 0.10). When the crushing of the element reaches F_a times the initial thickness the contact algorithm begins to act.

Attribute 3: ED, Optional modulus for interior contact stiffness.

Attribute 4: TYPE, Formulation for interior contact.

EQ.1.0: Default, recommended for uniform compression

EQ.2.0: Designed to control the combined modes of shear and compression. Works for type 1 brick formulation and type 10 tetrahedron formulation.

Define each part set with the *SET_PART_COLUMN option to specify independent attribute values for each part in the part set,

Remarks:

The interior penalty is determined by the formula:

$$K = \frac{\text{SLSFAC} \times \text{PSF} \times \text{Volume}^{2/3} \times \text{E}}{\text{Min. Thickness}}$$

where SLSFAC is the value specified on the *CONTROL_CONTACT card , volume is the volume of the brick element, E is a constitutive modulus, and min. thickness is approximately the thickness of the solid element through its thinnest dimension. If ED, is defined above the interior penalty is then given instead by:

$$K = \frac{\text{Volume}^{2/3} \times \text{ED}}{\text{Min. Thickness}}$$

where the scaling factors are ignored. Generally, ED should be taken as the locking modulus specified for the foam constitutive model.

Caution should be observed when using this option since if the time step size is too large an instability may result. The time step size is not affected by the use of interior contact.

*CONTACT_RIGID_SURFACE

Purpose: Define rigid surface contact. The purpose of rigid surface contact is to model large rigid surfaces, e.g., road surfaces, with nodal points and segments that require little storage and are written out at the beginning of the binary databases. The rigid surface motion, which can be optionally prescribed, is defined by a displacement vector which is written with each output state. The nodal points defining the rigid surface must be defined in the *NODE_RIGID_SURFACE section of this manual. These rigid nodal points do not contribute degrees-of-freedom.

Card 1	1	2	3	4	5	6	7	8
Variable	CID	PSID	BOXID	SSID	FS	FD	DC	VC
Type	I	I	I	I	F	F	F	F
Default	none	none	0	none	0.	0.	0.	0.

Card 2	1	2	3	4	5	6	7	8
Variable	LCIDX	LCIDY	LCIDZ	FSLCID	FDLCID			
Type	I	I	I	I	I			
Default	0	0	0	0	0			

Card 3	1	2	3	4	5	6	7	8
Variable	SFS	STTHK	SFTHK	XPENE	BSORT	CTYPE		
Type	F	F	F	F	I	I		
Default	1.0	0.0	1.0	4.0	10	0		

VARIABLE	DESCRIPTION
CID	Contact interface ID. This must be a unique number.

VARIABLE **DESCRIPTION**

PSID

Part set ID of all parts that may contact the rigid surface. See *SET_PART.

BOXID

Include only nodes of the part set that are within the specified box, see *DEFINE_BOX, in contact. If BOXID is zero, all nodes from the part set, PSID, will be included in the contact.

SSID

Segment set ID defining the rigid surface. See *SET_SEGMENT.

FS

Static coefficient of friction. The frictional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact,

$$\mu_c = \mathrm{FD} + (\mathrm{FS} - \mathrm{FD})e^{-\mathrm{DC}|v_{\mathrm{rel}}|}.$$

If FSLCID is defined, see below, then FS is overwritten by the value from the load curve.

FD

Dynamic coefficient of friction. The frictional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact,

$$\mu_c = \mathrm{FD} + (\mathrm{FS} - \mathrm{FD})e^{-\mathrm{DC}|v_{\mathrm{rel}}|}.$$

If FDLCID is defined, see below, then FD is overwritten by the value from the load curve.

DC

Exponential decay coefficient. The frictional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact

$$\mu_c = \mathrm{FD} + (\mathrm{FS} - \mathrm{FD})e^{-\mathrm{DC}|v_{\mathrm{rel}}|}.$$

VC

Coefficient for viscous friction. This is necessary to limit the friction force to a maximum. A limiting force is computed,

$$F_{\mathrm{lim}} = \mathrm{VC} \times A_{\mathrm{cont}}.$$

A_{cont} being the area of the segment contacted by the node in contact. The suggested value for *VC* is to use the yield stress in shear $\mathrm{VC} = \frac{\sigma_o}{\sqrt{3}}$ where σ_0 is the yield stress of the contacted material.

LCIDX

Load curve ID defining x-direction motion. If zero, there is no motion in the x-coordinate system.

LCIDY

Load curve ID defining y-direction motion. If zero, there is no motion in the y-coordinate system.

VARIABLE	DESCRIPTION
LCIDZ	Load curve ID defining z-direction motion. If zero, there is no motion in the z-coordinate system.
FSLCID	Load curve ID defining the static coefficient of friction as a function of interface pressure. This option applies to shell segments only.
FDLCID	Load curve ID defining the dynamic coefficient of friction as a function of interface pressure. This option applies to shell segments only.
SFS	Scale factor on default slave penalty stiffness, see also *CONTROL_CONTACT.
STTHK	Optional thickness for slave surface (overrides true thickness). This option applies to contact with shell, solid, and beam elements. True thickness is the element thickness of the shell elements. Thickness offsets are not used for solid element unless this option is specified.
SFTHK	Scale factor for slave surface thickness (scales true thickness). This option applies only to contact with shell elements. True thickness is the element thickness of the shell elements.
XPENE	Contact surface maximum penetration check multiplier. If the penetration of a node through the rigid surface exceeds the product of XPENE and the slave node thickness, the node is set free. EQ.0: default is set to 4.0.
BSORT	Number of cycles between bucket sorts. The default value is set to 10 but can be much larger, e.g., 50-100, for fully connected surfaces.
CTYPE	The contact formulation. The default, CTYPE = 0, is equivalent to the ONE_WAY_SURFACE_TO_SURFACE formulation, and CTYPE = 1 is a penalty formulation. If the slave surface belongs to a rigid body, CTYPE = 1 must be used.

Remarks:

Thickness offsets do not apply to the rigid surface. There is no orientation requirement for the segments in the rigid surface, and the surface may be assembled from disjoint, but contiguous, arbitrarily oriented meshes. With disjoint meshes, the global searches must be done frequently, about every 10 cycles, to ensure a smooth movement of a slave node between mesh patches. For fully connected meshes this frequency interval can be safely set to 50-200 steps between searches.

The modified binary database, d3plot, contains the road surface information prior to the state data. This information includes:

NPDS = Total number of rigid surface points in problem.

NRSC = Total number of rigid surface contact segments summed over all definitions.

NSID = Number of rigid surface definitions.

NVELQ = Number of words at the end of each binary output state defining the rigid surface motion. This equals 6 × NSID if any rigid surface moves or zero if all rigid surfaces are stationary.

PIDS = An array equal in length to NPDS. This array defines the ID for each point in the road surface.

XC = An array equal in length to 3 × NPDS. This array defines the global x, y, and z coordinates of each point.

For each road surface define the following NSID sets of data:

ID = Rigid surface ID.

NS = Number of segments in rigid surface.

IXRS = An array equal in length to 4 × NS. This is the connectivity of the rigid surface in the internal numbering system.

At the end of each state, 6 × NVELQ words of information are written. For each road surface the x, y, and z displacements and velocities are written. If the road surface is fixed, a null vector should be output. Skip this section if NVELQ = 0. LS-PrePost currently displays rigid surfaces and animates their motion.

*CONTACT_1D

Purpose: Define one-dimensional slide lines for rebar in concrete.

Card 1	1	2	3	4	5	6	7	8
Variable	NSIDS	NSIDM	ERR	SIGC	GB	SMAX	EXP	
Type	I	I	F	F	F	F	F	
Default	none	none	0.	0.	0.	0.	0.	

VARIABLE	DESCRIPTION
NSIDS	Nodal set ID for the slave nodes, see *SET_NODE.
NSIDM	Nodal set ID for the master nodes, see *SET_NODE.
ERR	External radius of rebar
SIGC	Unconfined compressive strength of concrete, f_c
GB	Bond shear modulus
SMAX	Maximum shear strain
EXP	Exponent in damage curve

Remarks:

With this option the concrete is defined with solid elements and the rebar with truss elements, each with their own unique set of nodal points. A string of spatially consecutive nodes, called slave nodes, related to the truss elements may slide along another string of spatially consecutive nodes, called master nodes, related to the solid elements. The sliding commences after the rebar debonds.

The bond between the rebar and concrete is assumed to be elastic perfectly plastic. The maximum allowable slip strain is given as:

$$u_{\max} = \mathrm{SMAX} \times e^{-\mathrm{EXP} \times D}$$

where D is the damage parameter $D_{n+1} = D_n + \Delta u$. The shear force, acting on area A_S, at time $n + 1$ is given as:

$$f_{n+1} = \min[f_n - \mathrm{GB} \times A_s \times \Delta u, \mathrm{GB} \times A_s \times u_{\max}]$$

***CONTACT_2D**_*OPTION1_{OPTION2}_{OPTION3}*

Purpose: Define a 2-dimensional contact interface or slide line. This option is to be used with 2D solid and shell elements using the plane stress, plane strain or axisymmetric formulations, see *SECTION_SHELL and SECTION_BEAM.

All the 2D contacts are supported in SMP. Only *CONTACT_2D_AUTOMATIC_SINGLE_SURFACE and *CONTACT_2D_AUTOMATIC_SURFACE_TO_SURFACE are supported for MPP.

OPTION1 specifies the contact type. The following options activate kinematic constraints and should be used with deformable materials only, but may be used with rigid bodies if the rigid body is the master surface and all rigid body motions are prescribed. Kinematic constraints are recommended for high pressure hydrodynamic applications.

SLIDING_ONLY

TIED_SLIDING

SLIDING_VOIDS

AUTOMATIC_TIED_ONE_WAY

The following option uses both kinematic constraints and penalty constraints.

AUTOMATIC_TIED

The following options are penalty based. These methods have no rigid-material limitations. They are recommended for lower pressure solid mechanics applications.

PENALTY_FRICTION

PENALTY

AUTOMATIC_SINGLE_SURFACE

AUTOMATIC_SINGLE_SURFACE_MORTAR

AUTOMATIC_SURFACE_TO_SURFACE

AUTOMATIC_SURFACE_TO_SURFACE_MORTAR

AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE

AUTOMATIC_SURFACE_IN_CONTINUUM

For these contact types, the Mortar contact is available *only for implicit* and *only supported in SMP* at the moment.

The following options are used for SPH particles in contact with 2D solid elements (2D shell elements are not supported currently) using the plane stress, plane strain or axisymmetric formulations:

NODE_TO_SOLID

NODE_TO_SOLID_TIED

The following option is used to measure contact forces that are reported as RCFORC output.

FORCE_TRANSDUCER

OPTION2 specifies a thermal contact and takes the single option:

THERMAL

Only the AUTOMATIC contact options: SINGLE_SURFACE, SURFACE_TO_SURFACE, and ONE_WAY_SURFACE_TO_SURFACE may be used with the THERMAL option.

OPTION3 specifies that the first card to read defines the title and ID number of contact interface and takes the single option:

TITLE

Title Card. Additional card for the TITLE keyword potion.

Title	1	2	3	4	5	6	7	8
Variable	CID	NAME						
Type	I	A70						

The 2D contact may be divided into 3 groups, each with a unique input format.

1. The first group were adopted from LS-DYNA2D and originated in the public domain version of DYNA2D from the Lawrence Livermore National Laboratory. Contact surfaces are specified as ordered sets of nodes. These sets define either contact surfaces or slide lines. The keyword options for the first group are:

   SLIDING_ONLY

   TIED_SLIDING

SLIDING_VOIDS

PENALTY_FRICTION

PENALTY

> **NOTE:** TIED_SLIDING, PENALTY_FRICTION and PENALTY options are not recommended since there are automatic options in the second group that are easier to use and provide the same functionality.

2. The second group contains the automatic contacts. These contact surfaces may be defined using part sets or unordered node sets. Segment orientations are determined automatically. The keywords for these are:

   AUTOMATIC_SINGLE_SURFACE

   AUTOMATIC_SINGLE_SURFACE_MORTAR

   AUTOMATIC_SURFACE_TO_SURFACE

   AUTOMATIC_SURFACE_TO_SURFACE_MORTAR

   AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE

   AUTOMATIC_SURFACE_IN_CONTINUUM

   AUTOMATIC_TIED

   AUTOMATIC_TIED_ONE_WAY

   FORCE_TRANSDUCER

3. The third group is used for SPH particles in contact with continuum elements:

   NODE_TO_SOLID

   NODE_TO_SOLID_TIED

Each of the 3 groups has a section below with a description of input and additional remarks.

CONTACT_2D_[SLIDING, TIED, & PENALTY]_*OPTION

This section documents the *CONTACT_2D variations derived from DYNA2D:

SLIDING_ONLY

TIED_SLIDING

SLIDING_VOIDS

PENALTY_FRICTION

PENALTY.

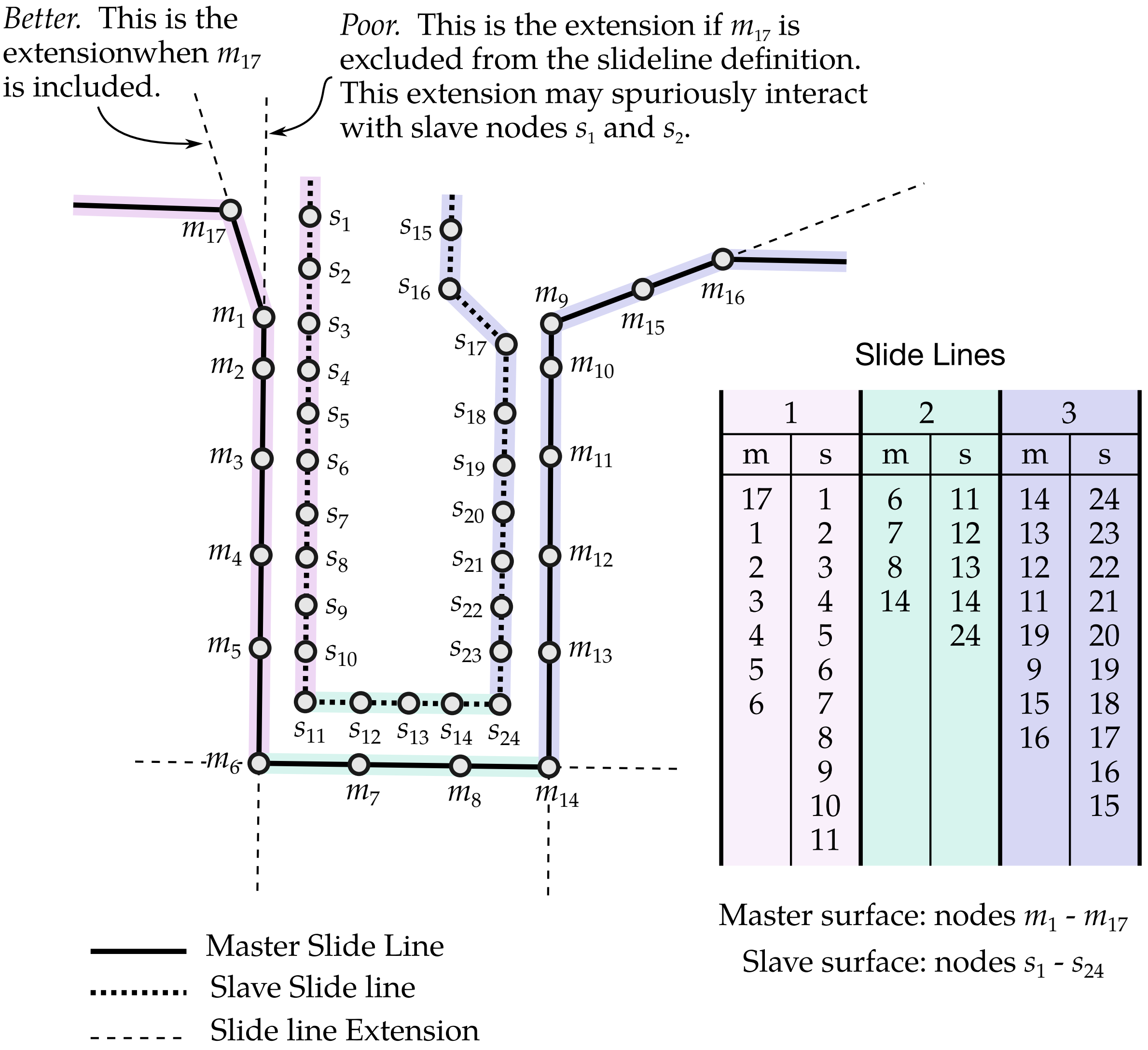
Card 1	1	2	3	4	5	6	7	8
Variable	SSID	MSID	TBIRTH	TDEATH				
Type	I	I	F	F				
Default	none	none	0.	1.e20				

Card 2	1	2	3	4	5	6	7	8
Variable	EXT_PAS	THETA1	THETA2	TOL_IG	PEN	TOLOFF	FRCSCL	ONEWAY
Type	I	F	F	F	F	F	F	F
Default	none	none	none	0.001	0.1	0.025	0.010	0.0

Friction Card. Additional card for the PENALTY_FRICTION keyword option.

Card 3	1	2	3	4	5	6	7	8
Variable	FRIC	FRIC_L	FRIC_H	FRIC_S				
Type	F	F	F	F				

VARIABLE	DESCRIPTION
SSID	Nodal set ID for the slave nodes, see *SET_NODE. The slave surface must be to the left of the master surface.
MSID	Nodal set ID for the master nodes, see *SET_NODE.
TBIRTH	Birth time for contact.
TDEATH	Death time for contact.
EXT_PAS	Slide line extension bypass option. EQ.0: extensions are use EQ.1: extensions are not used
THETA1	Angle in degrees of slide line extension at first master node. EQ.0: extension remains tangent to first master segment.
THETA2	Angle in degrees of slide line extension at last master node. EQ.0: extension remains tangent to last master segment.
TOL_IG	Tolerance for determining initial gaps. EQ.0.0: default set to 0.001
PEN	Scale factor or penalty. EQ.0.0: default set to 0.10
TOLOFF	Tolerance for stiffness insertion for implicit solution only. The contact stiffness is inserted when a node approaches a segment a distance equal to the segment length multiplied by TOLOFF. The stiffness is increased as the node moves closer with the full stiffness being used when the nodal point finally makes contact. EQ.0.0: default set to 0.025.
FRCSCL	Scale factor for the interface friction. EQ.0.0: default set to 0.010


Better. This is the extensionwhen m_{17} is included.

Poor. This is the extension if m_{17} is excluded from the slideline definition. This extension may spuriously interact with slave nodes s_1 and s_2.

Slide Lines

1		2		3	
m	s	m	s	m	s
17	1	6	11	14	24
1	2	7	12	13	23
2	3	8	13	12	22
3	4	14	14	11	21
4	5		24	19	20
5	6			9	19
6	7			15	18
	8			16	17
	9				16
	10				15
	11				

Master surface: nodes m_1 - m_{17}

Slave surface: nodes s_1 - s_{24}

▬▬▬	Master Slide Line
········	Slave Slide line
- - - - - -	Slide line Extension

Figure 11-26. *Slide line Example.* Note: (1) as recommend, for 90° angles each facet is assigned a distinct slide line; (2) the master slide line is more coarsely meshed; (3) the slave is to the left of the master (following the node ordering, see inset table); (4) as shown for slave nodes 1 and 2 it is important the slide line extension does not spuriously come into contact.

VARIABLE	DESCRIPTION
ONEWAY	Flag for one way treatment. If set to 1.0 the nodal points on the slave surface are constrained to the master surface. This option is generally recommended if the master surface is rigid. EQ.1.0: activate one way treatment.
FRIC	Coefficient of friction
FRIC_L	Coefficient of friction at low velocity.

VARIABLE	DESCRIPTION
FRIC_H	Coefficient of friction at high velocity.
FRIC_S	Friction factor for shear.

Remarks:

The SLIDING_ONLY option is a two-surface method based on a kinematic formulation. The two surfaces are allowed to slide arbitrarily large distances without friction, but are not permitted to separate or interpenetrate. Surfaces should be initially in contact. This option performs well when extremely high interface pressures are present. The more coarsely meshed surface should be chosen as the master surface for best performance.

The TIED_SLIDING option joins two parts of a mesh with differing mesh refinement. It is a kinematic formulation so the more coarsely meshed surface should be chosen as the master.

The SLIDING_VOIDS option is a kinematic formulation without friction which permits two surfaces to separate if tensile forces develop across the interface. The surfaces may be initially in contact or initially separated.

The PENALTY_FRICTION and PENATLY options are penalty formulations so the designation of master and slave surfaces is not important. The two bodies may be initially separate or in contact. A rate-dependent Coulomb friction model is available for PENALTY_FRICTION.

Consider two slide line surfaces in contact. It is necessary to designate one as a slave surface and the other as a master surface. Nodal points defining the slave surface are called slave nodes, and similarly, nodes defining the master surface are called master nodes. Each slave-master surface combination is referred to as a slide line.

Many potential problems with the options can be avoided by observing the following precautions:

- Metallic materials should contain the master surface along high explosive-metal interfaces.
- SLIDING_ONLY type slide lines are appropriate along high explosive-metal interfaces. The penalty formulation is not recommended along such interfaces.
- If one surface is more finely zoned, it should be used as the slave surface. If penalty slide lines are used, PENALTY and PENALTY_FRICTION, then the slave-master distinction is irrelevant.
- A slave node may have more than one master segment, and may be included as a member of a master segment if a slide line intersection is defined.

- Angles in the master side of a slide line that approach 90° must be avoided.

  Whenever such angles exist in a master surface, two or more slide lines should be defined. This procedure is illustrated in Figure 11-26. An exception for the foregoing rule arises if the surfaces are tied. In this case, only one slide line is needed.

- Whenever two surfaces are in contact, the smaller of the two surfaces should be used as the slave surface. For example, in modeling a missile impacting a wall, the contact surface on the missile should be used as the slave surface.

- Care should be used when defining a master surface to prevent the extension from interacting with the solution. In Figures 11-26 and 11-27, slide line extensions are shown.

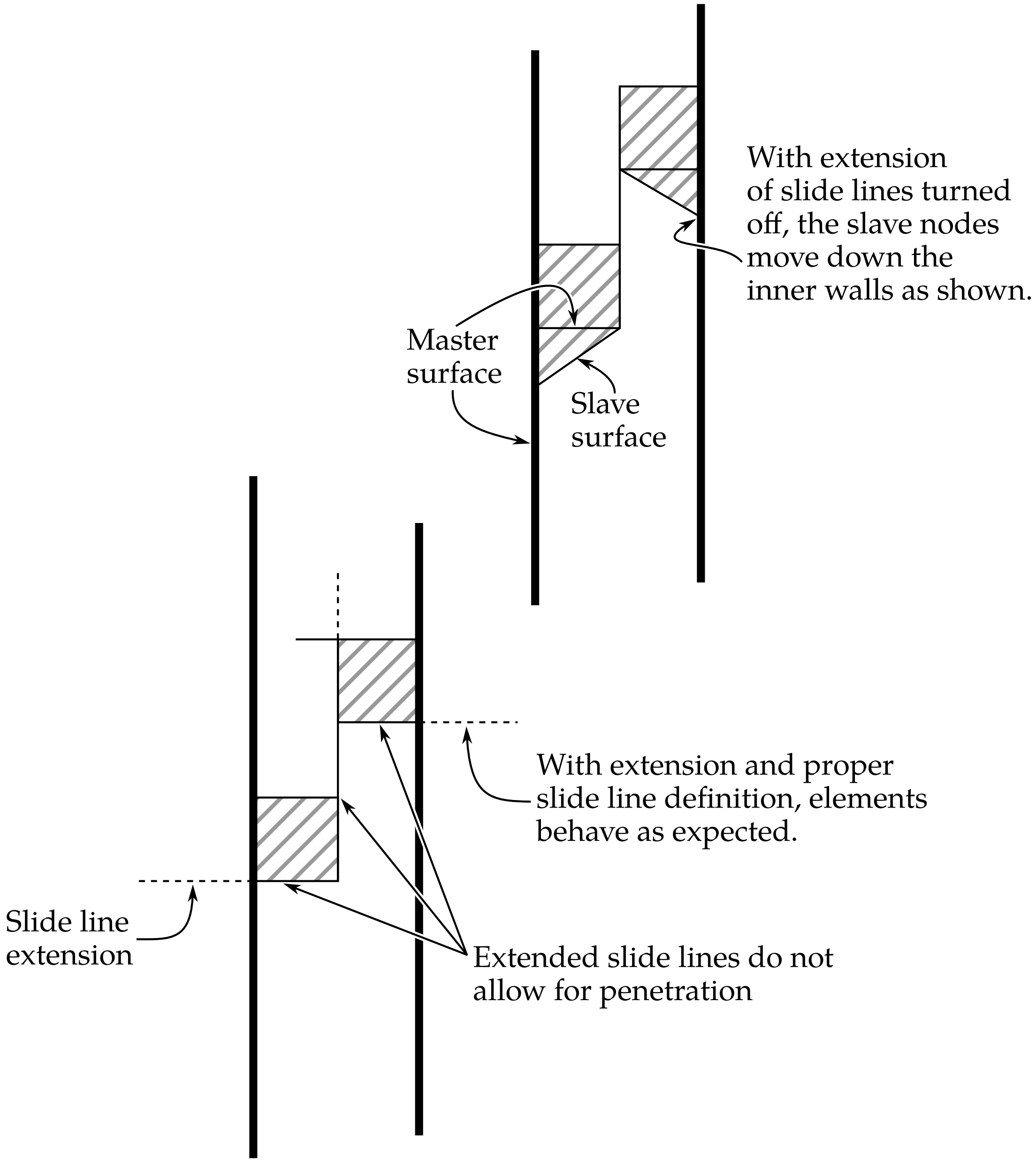


Figure 11-27. *With and without extension.* Extensions may be turned off by setting EXT_PAS (card 2), but, when turned off, slave nodes may "leak" out as shown in the upper version of the figure.

CONTACT_2D_[AUTOMATIC, & FORCE_TRANSDUCER]_*OPTION

This section documents the following variations of *CONTACT_2D:

AUTOMATIC_SINGLE_SURFACE

AUTOMATIC_SINGLE_SURFACE_MORTAR (*see* Remark 13)

AUTOMATIC_SURFACE_TO_SURFACE

AUTOMATIC_SURFACE_TO_SURFACE_MORTAR (*see* Remark 13)

AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE

AUTOMATIC_SURFACE_IN_CONTINUUM

AUTOMATIC_TIED

AUTOMATIC_TIED_ONE_WAY

FORCE_TRANSDUCER

Card 1	1	2	3	4	5	6	7	8
Variable	SIDS	SIDM	SFACT	FREQ	FS	FD	DC	
Type	I	I	F	I	F	F	F	
Default	none	none	1.0	50	0.	0.	0.	
Remarks	1	1						

Card 2	1	2	3	4	5	6	7	8
Variable	TBIRTH	TDEATH	SOS	SOM	NDS	NDM	COF	INIT
Type	F	F	F	F	I	I	I	I
Default	0.	1.e20	1.0	1.0	0	0	0	0
Remarks			3	3	2	2		6

Automatic Thermal Card. Additional card for keywords with both the AUTOMATIC and THERMAL options. For example, *CONTACT_2D_AUTOMATIC_..._THERMAL_.....

Card 3	1	2	3	4	5	6	7	8
Variable	K	RAD	H	LMIN	LMAX	CHLM	BC_FLAG	
Type	F	F	F	F	F	F	I	
Default	none	none	none	none	none	1.0	0	

Automatic Optional Card 1. Optional card for the AUTOMATIC keyword option.

Card 4	1	2	3	4	5	6	7	8
Variable	VC	VDC	IPF	SLIDE	ISTIFF	TIEDGAP	IGAPCL	TIETYP
Type	F	F	I	I	I	R	I	I
Default	0.	10.0	0	0	0		0	0
Remarks				7	8	9		9

VARIABLE	DESCRIPTION
SIDS	Set ID to define the slave surface. If SIDS > 0, a part set is assumed, see *SET_PART. If SIDS < 0, a node set with ID equal to the absolute value of SIDS is assumed, see *SET_NODE.
SIDM	Set ID to define the master surface. If SIDM > 0, a part set is assumed, see *SET_PART. If SIDM < 0, a node set with ID equal to the absolute value of SIDM is assumed, see *SET_NODE. Do not define for single surface contact.
SFACT	Scale factor for the penalty force stiffness.
FREQ	Search frequency. The number of timesteps between bucket sorts. For implicit contact this parameter is ignored and the search frequency is 1.

EQ.0: default set to 50.

FS

Static coefficient of friction. The frictional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact according to the relation given by:

$$\mu_c = \mathrm{FD} + (\mathrm{FS} - \mathrm{FD}) e^{-\mathrm{DC}|v_{rel}|}.$$

FD

Dynamic coefficient of friction. The frictional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact

$$\mu_c = \mathrm{FD} + (\mathrm{FS} - \mathrm{FD}) e^{-\mathrm{DC}|v_{\mathrm{rel}}|}.$$

This parameter does not apply to Mortar contact.

DC

Exponential decay coefficient. The frictional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact

$$\mu_c = \mathrm{FD} + (\mathrm{FS} - \mathrm{FD}) e^{-\mathrm{DC}|v_{\mathrm{rel}}|}.$$

This parameter does not apply to Mortar contact.

TBIRTH

Birth time for contact.

TDEATH

Death time for contact.

SOS

Surface offset from midline for 2D shells of slave surface

EQ.0.0: default to 1.

GT.0.0: scale factor applied to actual thickness

LT.0.0: absolute value is used as the offset

SOM

Surface offset from midline for 2D shells of master surface

EQ.0: default to 1.

GT.0: scale factor applied to actual thickness

LT.0: absolute value is used as the offset

NDS Normal direction flag for 2D shells of slave surface

EQ.0: Normal direction is determined automatically

EQ.1: Normal direction is in the positive direction

EQ.-1: Normal direction is in the negative direction

NDM Normal direction flag for 2D shells of master surface

EQ.0: Normal direction is determined automatically

EQ.1: Normal direction is in the positive direction

EQ.-1: Normal direction is in the negative direction

COF Closing/Opening flag for implicit contact

EQ.0: Recommended for most problem where gaps are only closing.

EQ.1: Recommended when gaps are opening to avoid sticking.

This parameter does not apply to Mortar contact.

INIT Special processing during initialization

EQ.0: No special processing.

EQ.1: Forming option.

K Thermal conductivity (k) of fluid between the slide surfaces. If a gap with a thickness l_{gap} exists between the slide surfaces, then the conductance due to thermal conductivity between the slide surfaces is

$$h_{\mathrm{cond}} = \frac{k}{l_{\mathrm{gap}}}$$

Note that LS- DYNA calculates l_{gap} based on deformation.

RAD Radiation factor, f, between the slide surfaces. A radiant-heat-transfer coefficient (h_{rad}) is calculated (see *BOUNDARY_RADIATION). If a gap exists between the slide surfaces, then the contact conductance is calculated by

$$h = h_{\mathrm{cond}} + h_{\mathrm{rad}}$$

H Heat transfer conductance (h_{cont}) for closed gaps. Use this heat transfer conductance for gaps in the range

$$0 \le l_{gap} \le l_{\min}$$

where l_{min} is GCRIT defined below.

LMIN

Critical gap (l_{min}), use the heat transfer conductance defined (HTC) for gap thicknesses less than this value.

LMAX

No thermal contact if gap is greater than this value (l_{max}).

CHLM

Is a multiplier used on the element characteristic distance for the search routine. The characteristic length is the largest interface surface element diagonal.

EQ.0: Default set to 1.0

BC_FLAG

Thermal boundary condition flag

EQ.0: thermal boundary conditions are on when parts are in contact

EQ.1: thermal boundary conditions are off when parts are in contact

VC

Coefficient for viscous friction. This is used to limit the friction force to a maximum. A limiting force is computed

$$F_{\mathrm{lim}} = \mathrm{VC} \times A_{\mathrm{cont}}.$$

A_{cont} being the area of contacted between segments. The suggested value for VC is to use the yield stress in shear:

$$\mathrm{VC} = \frac{\sigma_o}{\sqrt{3}}$$

where σ_o is the yield stress of the contacted material.

VDC

Viscous damping coefficient in percent of critical for explicit contact. This parameter does not apply to Mortar contact.

IPF

Initial penetration flag for explicit contact.

EQ.0: Allow initial penetrations to remain

EQ.1: Push apart initially penetrated surfaces

SLIDE

Sliding option.

EQ.0: Off

EQ.1: On

ISTIFF

Stiffness scaling option.

EQ.0: Use default option.

EQ.1: Scale stiffness using segment masses and explicit time step (default for explicit contact)

EQ.2: Scale stiffness using segment stiffness and dimensions (default for implicit contact)

TIEDGAP Search gap for tied contacts.

EQ.0: Default, use 1% of the master segment length

GT.0: Use the input value

LT.0: Use –TIEDGAP % of the master segment length.

IGAPCL Flag to close gaps in tied contact

EQ.0: Default, allow gaps to remain

EQ.1: Move slave nodes to master segment to close gaps

TIETYP Flag to control constraint type of tied contact

EQ.0: Default, use kinematic constraints when possible

EQ.1: Use only penalty type constraints

Remarks:

1. The SINGLE_SURFACE, SURFACE_TO_SURFACE, and ONE_WAY_SURFACE_TO_SURFACE options use penalty forces to prevent penetration between 2D shell elements and external faces of 2D continuum elements. Contact surfaces are defined using SIDS and SIDM to reference either part sets or node sets. If part sets are used, all elements and continuum faces of the parts in the set are included in contact. If node sets are used, elements or continuum faces that have both nodes in the set are included in the contact surface. The SINGLE_SURFACE option uses only the slave set and checks for contact between all elements and continuum faces in the set. If SSID is blank or zero, contact will be checked for all elements and continuum faces in the model. With the other options, both SSID and MSID are required.

2. The FORCE_TRANSDUCER option should be used in conjunction with at least one AUTOMATIC contact options. It does nothing to prevent penetration, but measures the forces generated by other contact definitions. The FORCE_-TRANSDUCER option uses only SIDS, and optionally SIDM. If only SIDS is defined, the force transducer measures the resultant contact force on all the elements and continuum faces in the slave surface. If both SIDS and SIDM are defined, then the force transducer measures contact forces between the elements and continuum faces in the slave surface and master surface. The meas-

ured forces are included in the rcforc output. In the case of an axisymmetric analysis, values output to rcforc and ncforc are in units of force per radian (this includes both shell types 14 and 15).

3. By default, the normal direction of 2D shell elements is evaluated automatically for SINGLE_SURFACE, SURFACE_TO_SURFACE and ONE_WAY_SURFACE_TO_SURFACE contact. The user can override the automatic algorithm using NDS or NDM and contact will occur with the positive or negative face of the element.

4. By default, the true thickness of 2D shell elements is taken into account for the SURFACE_TO_SURFACE, SINGLE_SURFACE, and ONE_WAY_SURFACE_TO_SURFACE options. The user can override the true thickness by using SOS and SOM. If the surface offset is reduced to a small value, the automatic normal direction algorithm may fail, so it is best to specify the normal direction using NDS or NDM.

5. For all AUTOMATIC contact options, eroding materials are treated by default. At present, subcycling is not possible.

6. The INIT parameter activates a forming option that is intended for implicit solutions of thin solid parts when back side segments may interfere with the solution. It automatically removes back side segments during initialization. Alternatively, the user can input INIT = 0, and use node set input to limit the contact interface to just the front of a thin part.

7. For the thermal option:

   $h = h_{\mathrm{cont}_,}$ if the gap thickness is $0 \le l_{\mathrm{gap}} \le l_{\mathrm{min}}$

   $h = h_{\mathrm{cond}} + h_{\mathrm{rad}_,}$ if the gap thickness is $l_{\mathrm{min}} \le l_{\mathrm{gap}} \le l_{\mathrm{max}}$

   $h = 0$, if the gap thickness is $l_{\mathrm{gap}} > l_{\mathrm{max}}$

8. The SLIDE parameter activates a sliding option which uses additional logic to improve sliding when surfaces in contact have kinks or corners. This option is off by default.

9. The ISTIFF option allows control of the equation used in calculating the penalty stiffness. For backward compatibility, the default values are different for implicit and explicit solutions. When ISTIFF = 1 is used, the explicit time step appears in the stiffness equation regardless if the calculation is implicit or explicit.

10. The TIED_ONE_WAY contact creates two degree of freedom translational kinematic constraints to nodes on the slave surface which are initially located

on or near master segments. The TIED option creates kinematic constraints between slave nodes and master segments, and also creates penalty constraints between master nodes and slave segments. With either contact option, a kinematic constraint may be switched to penalty if there is a conflict with another constraint. The TIEDGAP parameter determines the maximum normal distance from a segment to a node for a constraint to be formed. Nodes will not be moved to eliminate an initial gap, and the initial gap will be maintained throughout the calculation. If TIETYP = 1, then only penalty constraints will be used.

11. Note that the SURFACE_IN_CONTINUUM option has been deprecated in favor of the *CONSTRAINED_LAGRANGE_IN_SOLID keyword which allows coupling between fluids and structures. However, this option is maintained to provide backward compatibility for existing data.

    For the SURFACE_IN_CONTINUUM option, penalty forces prevent the flow of slave element material (the continuum) through the master surfaces. Flow of the continuum tangent to the surface is permitted. Only 2D solid parts are permitted in the slave part set. Both 2D solid and 2D shell parts are permitted in the master part set. Flow through 2D shell elements is prevented in both directions by default. If NDM is set to ±1, flow in the direction of the normal is permitted. Thickness of 2D shell elements is ignored.

12. When using the SURFACE_IN_CONTINUUM option, there is no need to mesh the continuum around the structure because contact is not with continuum nodes but with material in the interior of the continuum elements. The algorithm works well for Eulerian or ALE elements since the structure does not interfere with remeshing. However, a structure will usually not penetrate the surface of an ALE continuum since the nodes are Lagrangian normal to the surface. Therefore, if using an ALE fluid, the structure should be initially immersed in the fluid and remain immersed throughout the calculation. Penetrating the surface of an Eulerian continuum is not a problem.

13. The Mortar contact (MORTAR) is available for implicit calculations in SMP (MPP *not* supported). The apparent behavior compared to the non-Mortar contact is very similar, the difference lies in details concerning the constitutive relation (contact stress vs relative motion of contact surfaces) and the kinematics (the relative motion of contact surfaces as function of nodal coordinates). Mortar contact is designed for continuity and smoothness that is beneficial for an implicit solution scheme, and is intended to enhance robustness in such a context. For details regarding the 2D Mortar contact, see the LS-DYNA Theory Manual.

CONTACT_2D_NODE_TO_SOLID_*OPTION

This section documents the following variations of *CONTACT_2D:

NODE_TO_SOLID

NODE_TO_SOLID_TIED

Card 1	1	2	3	4	5	6	7	8
Variable	SSID	MSID	TBIRTH	TDEATH				
Type	I	I	F	F				
Default	none	None	0.	1.e20				

Card 2	1	2	3	4	5	6	7	8
Variable	SOFT		VC	OFFD	PEN	FS	FD	DC
Type	I		F	F	F	F	F	F
Default	0		0.0	0.0	1.0/0.1	0.0	0.0	0.0

VARIABLE	DESCRIPTION
SSID	Nodal set ID or part set ID for the slave nodes, If SSID > 0, a nodal set ID is assumed, If SSID < 0 a part set ID is assumed.
MSID	Master part set ID. MSID < 0 since only part set is allowed.
TBIRTH	Birth time for contact.
TDEATH	Death time for contact.

VARIABLE **DESCRIPTION**

SOFT

Soft constraint option:

EQ.0: penalty formulation,

EQ.1: soft constraint formulation.

The soft constraint may be necessary if the material constants of the parts in contact have a wide variation in the elastic bulk moduli. In the soft constraint option, the interface stiffness is based on the nodal mass and the global time step size. The soft constraint option is also recommended for axisymmetric simulations.

VC

Coefficient for viscous friction. This is used to limit the friction force to a maximum. A limiting force is computed

$$F_{\mathrm{lim}} = \mathrm{VC} \times A_{\mathrm{cont}}.$$

A_{cont} being the area of contacted between segments. The suggested value for VC is to use the yield stress in shear:

$$\mathrm{VC} = \frac{\sigma_o}{\sqrt{3}}$$

where σ_o is the yield stress of the contacted material.

OFFD

Contact offset distance for slave nodes (SPH particles), not for tie contact right now. Recommended to be half of the original particle spacing in contact direction.

PEN

Scale factor for penalty.

EQ.0.0: default set to 1.0 for penalty formulation, or 0.1 for soft constraint formulation.

FS

Static coefficient of friction. The frictional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact according to the relationship given by:

$$\mu_c = \mathrm{FD} + (\mathrm{FS} - \mathrm{FD})e^{-\mathrm{DC}|v_{\mathrm{rel}}|}.$$

FD

Dynamic coefficient of friction. The frictional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact

$$\mu_c = \mathrm{FD} + (\mathrm{FS} - \mathrm{FD})e^{-\mathrm{DC}|v_{\mathrm{rel}}|}.$$

VARIABLE	DESCRIPTION
DC	Exponential decay coefficient. The frictional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact $$\mu_c = \mathrm{FD} + (\mathrm{FS} - \mathrm{FD})e^{-\mathrm{DC}\lvert v_{\mathrm{rel}}\rvert}.$$

Remarks:

NODE_TO_SOLID contact is a penalty based contact type used only for SPH particles with solid elements using the plane stress, plane strain or axisymmetric formulation. NODE_TO_SOLID_TIED contact is used only for SPH particles tied with solid elements, an offset of distance h (smooth length) is adopted for each SPH particle.

*CONTROL

The keyword control cards are optional and can be used to change defaults activate solution options such as mass scaling adaptive remeshing and an implicit solution however it is advisable to define the CONTROL_TERMINATION card. **The ordering of the control cards in the input file is arbitrary. To avoid ambiguities define no more than one control card of each type.** The following control cards are organized in alphabetical order

*CONTROL_ACCURACY

*CONTROL_ACOUSTIC

*CONTROL_ADAPSTEP

*CONTROL_ADAPTIVE

*CONTROL_ADAPTIVE_CURVE

*CONTROL_ALE

*CONTROL_BULK_VISCOSITY

CONTROL_CHECK_*SHELL

*CONTROL_COARSEN

*CONTROL_CONTACT

*CONTROL_COUPLING

*CONTROL_CPM

*CONTROL_CPU

*CONTROL_DEBUG

*CONTROL_DISCRETE_ELEMENT

*CONTROL_DYNAMIC_RELAXATION

*CONTROL_EFG

*CONTROL_ENERGY

*CONTROL_EXPLICIT_THERMAL_ALE_COUPLING

*CONTROL_EXPLICIT_THERMAL_BOUNDARY

*CONTROL_EXPLICIT_THERMAL_CONTACT

*CONTROL_EXPLICIT_THERMAL_INITIAL

*CONTROL_EXPLICIT_THERMAL_OUTPUT

*CONTROL_EXPLICIT_THERMAL_PROPERTIES

*CONTROL_EXPLICIT_THERMAL_SOLVER

*CONTROL_EXPLOSIVE_SHADOW

*CONTROL_FORMING_AUTOCHECK

*CONTROL_FORMING_AUTONET

*CONTROL_FORMING_AUTOPOSITION_PARAMETER

*CONTROL_FORMING_BESTFIT

*CONTROL_FORMING_INITIAL_THICKNESS

*CONTROL_FORMING_MAXID

*CONTROL_FORMING_ONESTEP

*CONTROL_FORMING_OUTPUT

*CONTROL_FORMING_PARAMETER_READ

*CONTROL_FORMING_POSITION

*CONTROL_FORMING_PREBENDING

*CONTROL_FORMING_PROJECTION

*CONTROL_FORMING_REMOVE_ADAPTIVE_CONSTRAINTS

*CONTROL_FORMING_SCRAP_FALL

*CONTROL_FORMING_SHELL_TO_TSHELL

*CONTROL_FORMING_STONING

*CONTROL_FORMING_TEMPLATE

*CONTROL_FORMING_TIPPING

*CONTROL_FORMING_TOLERANC

*CONTROL_FORMING_TRAVEL

*CONTROL_FORMING_TRIM_MERGE

*CONTROL_FORMING_TRIMMING

*CONTROL_FORMING_UNFLANGING

*CONTROL_FORMING_USER

*CONTROL_FREQUENCY_DOMAIN

CONTROL_HOURGLASS_*{OPTION}

*CONTROL_IMPLICIT_AUTO

*CONTROL_IMPLICIT_BUCKLE

*CONTROL_IMPLICIT_CONSISTENT_MASS

*CONTROL_IMPLICIT_DYNAMICS

*CONTROL_IMPLICIT_EIGENVALUE

*CONTROL_IMPLICIT_FORMING

*CONTROL_IMPLICIT_GENERAL

*CONTROL_IMPLICIT_INERTIA_RELIEF

*CONTROL_IMPLICIT_JOINTS

*CONTROL_IMPLICIT_MODAL_DYNAMIC

CONTROL_IMPLICIT_MODAL_DYNAMIC_DAMPING_*{OPTION}

CONTROL_IMPLICIT_MODAL_DYNAMIC_MODE_*{OPTION}

CONTROL_IMPLICIT_MODES_*{OPTION}

*CONTROL_IMPLICIT_ROTATIONAL_DYNAMICS

*CONTROL_IMPLICIT_SOLUTION

*CONTROL_IMPLICIT_SOLVER

*CONTROL_IMPLICIT_STABILIZATION

CONTROL_IMPLICIT_STATIC_CONDENSATION_*{OPTION}

*CONTROL_IMPLICIT_TERMINATION

*CONTROL_MAT

*CONTROL_MPP_CONTACT_GROUPABLE

CONTROL_MPP_DECOMPOSITION_ARRANGE_PARTS_*{OPTION}

*CONTROL_MPP_DECOMPOSITION_AUTOMATIC

*CONTROL_MPP_DECOMPOSITION_BAGREF

*CONTROL_MPP_DECOMPOSITION_CHECK_SPEED

*CONTROL_MPP_DECOMPOSITION_CONTACT_DISTRIBUTE

*CONTROL_MPP_DECOMPOSITION_CONTACT_ISOLATE

*CONTROL_MPP_DECOMPOSITION_DISTRIBUTE_ALE_ELEMENTS

*CONTROL_MPP_DECOMPOSITION_DISTRIBUTE_SPH_ELEMENTS

*CONTROL_MPP_DECOMPOSITION_ELCOST

*CONTROL_MPP_DECOMPOSITION_FILE

*CONTROL_MPP_DECOMPOSITION_METHOD

*CONTROL_MPP_DECOMPOSITION_NUMPROC

*CONTROL_MPP_DECOMPOSITION_OUTDECOMP

*CONTROL_MPP_DECOMPOSITION_PARTS_DISTRIBUTE

CONTROL_MPP_DECOMPOSITION_PARTSET_DISTRIBUTE_*{OPTION}

*CONTROL_MPP_DECOMPOSITION_RCBLOG

*CONTROL_MPP_DECOMPOSITION_SCALE_CONTACT_COST

*CONTROL_MPP_DECOMPOSITION_SCALE_FACTOR_SPH

*CONTROL_MPP_DECOMPOSITION_SHOW

*CONTROL_MPP_DECOMPOSITION_TRANSFORMATION

*CONTROL_MPP_IO_BINOUTONLY

*CONTROL_MPP_IO_LSTC_REDUCE

*CONTROL_MPP_IO_NOBEAMOUT

*CONTROL_MPP_IO_NOD3DUMP

*CONTROL_MPP_IO_NODUMP

*CONTROL_MPP_IO_NOFAIL

*CONTROL_MPP_IO_NOFULL

*CONTROL_MPP_IO_SWAPBYTES

*CONTROL_MPP_MATERIAL_MODEL_DRIVER

*CONTROL_MPP_PFILE

*CONTROL_NONLOCAL

*CONTROL_OUTPUT

*CONTROL_PARALLEL

*CONTROL_PORE_FLUID

*CONTROL_REFINE_ALE

*CONTROL_REFINE_ALE2D

*CONTROL_REFINE_MPP_DISTRIBUTION

*CONTROL_REFINE_SHELL

*CONTROL_REFINE_SOLID

*CONTROL_REMESHING

*CONTROL_REQUIRE_REVISION

*CONTROL_RIGID

*CONTROL_SHELL

*CONTROL_SOLID

*CONTROL_SOLUTION

*CONTROL_SPH

*CONTROL_SPOTWELD_BEAM

*CONTROL_START

*CONTROL_STAGED_CONSTRUCTION

*CONTROL_STEADY_STATE_ROLLING

CONTROL_STRUCTURED_*{OPTION}

*CONTROL_SUBCYCLE

*CONTROL_TERMINATION

*CONTROL_THERMAL_EIGENVALUE

*CONTROL_THERMAL_NONLINEAR

*CONTROL_THERMAL_SOLVER

*CONTROL_THERMAL_TIMESTEP

*CONTROL_TIMESTEP

*CONTROL_UNITS

LS-DYNA's implicit mode may be activated in two ways. Using the *CONTROL_IMPLICIT_GENERAL keyword, a simulation may be flagged to run entirely in implicit mode. Alternatively, an explicit simulation may be seamlessly switched into implicit mode at the termination time using the *INTERFACE_SPRINGBACK_SEAMLESS keyword. The seamless switching feature is intended to simplify metal forming springback calculations, where the forming phase can be run in explicit mode, followed immediately by an implicit static springback simulation. In case of difficulty, restart capability is supported. Eight keywords are available to support implicit analysis. Default values are carefully selected to minimize input necessary for most simulations. These are summarized below:

*CONTROL_IMPLICIT_GENERAL
Activates implicit mode, selects time step size.

*CONTROL_IMPLICIT_INERTIA_RELIEF
Allows linear analysis of models with rigid body modes.

*CONTROL_IMPLICIT_SOLVER
Selects parameters for solving system of linear equations [K]{x}={f}.

*CONTROL_IMPLICIT_SOLUTION
Selects linear or nonlinear solution method, convergence tolerances.

*CONTROL_IMPLICIT_AUTO
Activates automatic time step control.

*CONTROL_IMPLICIT_DYNAMICS
Activates and controls dynamic implicit solution using Newmark method.

*CONTROL_IMPLICIT_EIGENVALUE
Activates and controls eigenvalue analysis.

*CONTROL_IMPLICIT_MODES

Activates and controls computation of constraint and attachment modes.

*CONTROL_IMPLICIT_STABILIZATION

Activates and controls artificial stabilization for multi-step spring back.

*CONTROL_ACCURACY

Purpose: Define control parameters that can improve the accuracy of the calculation.

Card 1	1	2	3	4	5	6	7	8
Variable	OSU	INN	PIDOSU	IACC				
Type	I	I	I	I				
Default	0 (off)		optional	0 (off)				

VARIABLE	DESCRIPTION
OSU	Global flag for 2nd order objective stress updates (See Remark 1 below). Generally, for explicit calculations only those parts undergoing large rotations, such as rolling tires, need this option. Objective stress updates can be activated for a subset of part IDs by defining the part set in columns 21-30. EQ.0: Off (default) EQ.1: On
INN	Invariant node numbering for shell and solid elements. (See Remarks 2 and 3). EQ.-4: On for both shell and solid elements except triangular shells EQ.-2: On for shell elements except triangular shells EQ.1: Off (default for explicit) EQ.2: On for shell and thick shell elements (default for implicit) EQ.3: On for solid elements EQ.4: On for shell, thick shell, and solid elements
PIDOSU	Part set ID for objective stress updates. If this part set ID is given only those part IDs listed will use the objective stress update; therefore, OSU is ignored.

VARIABLE	DESCRIPTION
IACC	Implicit accuracy flag, turns on some specific accuracy considerations in implicit analysis at an extra CPU cost. See Remark 4. EQ.0: Off (default) EQ.1: On

Remarks:

1. **Second Order Objective Stress Update.** Second order objective stress updates are occasionally necessary. Some examples include spinning bodies such as turbine blades in a jet engine, high velocity impacts generating large strains in a few time steps, and large time step sizes due to mass scaling in metal forming. There is a significantly added cost which is due in part to the added cost of the second order terms in the stress update when the Jaumann rate is used and the need to compute the strain-displacement matrix at the mid-point geometry. This option is available for one point brick elements, the selective-reduced integrated brick element which uses eight integration points, the fully integrated plane strain and axisymmetric volume weighted (type 15) 2D solid elements, the thick shell elements, and the following shell elements: Belytschko-Tsay, Belytschko-Tsay with warping stiffness, Belytschko-Chiang-Wong, S/R Hughes-Liu, and the type 16 fully integrated shell element.

2. **Invariant Node Numbering for Shell Elements.** Invariant node numbering for shell and thick shell elements affects the choice of the local element shell coordinate system. The orientation of the default local coordinate system is based on the shell normal vector and the direction of the 1-2 side of the element. If the element numbering is permuted, the results will change in irregularly shaped elements. With invariant node numbering, permuting the nodes shifts the local system by an exact multiple of 90 degrees. In spite of its higher costs [<5%], the invariant local system is recommended for several reasons. First, element forces are nearly independent of node sequencing; secondly, the hourglass modes will not substantially affect the material directions; and, finally, stable calculations over long time periods are achievable. The INN parameter has no effect on thick shell form 2 which is always invariant and thick shell from 3 which is never invariant.

3. **Invariant Node Numbering for Solid Elements.** Invariant node numbering for solid elements is available for anisotropic materials only. This option has no effect on solid elements of isotropic material. This option is recommended when solid elements of anisotropic material undergo significant deformation.

4. **Implicit Calculations.** All other things being equal, a single time step of an implicit analysis usually involves a larger time increment and deformation than an explicit analysis. Many of the algorithms in LS-DYNA have been heavily optimized for explicit analysis in ways that are inappropriate for implicit analysis. While an implicit analysis, by default, invokes many measures to ensure accuracy, certain corrections associated with unusual applications or with large computational expense are invoked only by setting IACC = 1. A list of features that are included with this option follows.

   a) Strongly objective treatment of some tied contact interfaces, see *CONTACT.

   b) Fully iterative treatment of some piecewise linear plasticity materials, see *MAT_PIECEWISE_LINEAR_PLASTICITY and *MAT_MODIFIED_-PIECEWISE_LINEAR_PLASTICITY, including smooth decay of stresses down to zero when including failure.

   c) Strong objective treatment of some elements in the context of large rotations, applies to shell element types -16, 16 and 4, beam element types 1, 2 and 9, and solid element types -2, -1, 1, 2 and 16. The superposed rigid body motion is subtracted from these elements before evaluating the response which significantly reduces the presence of spurious strains.

*CONTROL_ACOUSTIC

Purpose: Define control parameters for transient acoustic solutions.

Card 1	1	2	3	4	5	6	7	8
Variable	MACDVP							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
MACDVP	Calculate the nodal displacements and velocities of *MAT_-ACOUSTIC volume elements for inclusion in d3plot and time-history files. EQ.0: Acoustic nodal motions will not be calculated EQ.1: Acoustic nodal motions will be calculated

Remarks:

1. *MAT_ACOUSTIC volume elements (ELFORM = 8 and ELFORM = 14) use the displacement potential as the fundamental unknown. The infinitesimal motions of these acoustic nodes can be found from the gradient of the displacement and velocity potentials. This is purely a post-processing endeavor and has no effect on the predicted pressures and structural response. It will however roughly double the cost of the acoustic solution and for that reason is not done by default.

2. The acoustic theory underpinning *MAT_ACOUSTIC volume elements presumes infinitesimal motions. In the presence of larger motions the pressure calculations will proceed regardless, but the calculation of acoustic nodal motions can then be unreliable.

*CONTROL_ADAPSTEP

Purpose: Define control parameters for contact interface force update during each adaptive cycle.

Card 1	1	2	3	4	5	6	7	8
Variable	FACTIN	DFACTR						
Type	F	F						
Default	1.0	0.01						

VARIABLE	DESCRIPTION
FACTIN	Initial relaxation factor for contact force during each adaptive remesh. To turn this option off set FACTIN = 1.0. Unless stability problems occur in the contact, FACTIN = 1.0 is recommended since this option can create some numerical noise in the resultant tooling forces. A typical value for this parameter is 0.10.
DFACTR	Incremental increase of FACTIN during each time step after the adaptive step. FACTIN is not allowed to exceed unity. A typical value might be 0.01.

Remarks:

1. This command applies to contact with thickness offsets including contact types:

   *CONTACT_FORMING_…_

   *CONTACT_NODES_TO_SURFACE_

   *CONTACT_SURFACE_TO_SURFACE

   *CONTACT_ONE_WAY_SURFACE_TO_SURFACE.

*CONTROL_ADAPTIVE

Purpose: Activate adaptive meshing. The parts which are adaptively meshed are defined by ADPOPT under *PART. Note that "sandwiched" part's adaptivity is available when the variable IFSAND is set to "1" and applies to ADPOPT = 1 and 2 only. Other related keywords include: *CONTROL_ADAPTIVE_CURVE, *DEFINE_-CURVE_TRIM (with variable TCTOL), *DEFINE_BOX_ADAPTIVE (moving adaptive box), and *DEFINE_CURVE_BOX_ADAPTIVITY. This keword is applicable to neither hyperelastic materials nor any material model based on a Total Lagrangian formulation.

Card 1	1	2	3	4	5	6	7	8
Variable	ADPFREQ	ADPTOL	ADPOPT	MAXLVL	TBIRTH	TDEATH	LCADP	IOFLAG
Type	F	F	I	I	F	F	I	I
Default	none	10^{20}	1	3	0.0	10^{20}	0	0

Remaining cards are optional.†

Card 2	1	2	3	4	5	6	7	8
Variable	ADPSIZE	ADPASS	IREFLG	ADPENE	ADPTH	MEMORY	ORIENT	MAXEL
Type	F	I	I	F	F	I	I	I
Default		0	0	0.0	inactive	inactive	0	inactive

Card 3	1	2	3	4	5	6	7	8
Variable	IADPN90	IADPGH	NCFREQ	IADPCL	ADPCTL	CBIRTH	CDEATH	LCLVL
Type	I	I	I	I	F	F	F	F
Default	0	0	none	1	none	0.0	10^{20}	

Card 4	1	2	3	4	5	6	7	8
Variable	CNLA			MMM2D	ADPERR	D3TRACE		IFSAND
Type	F			I	I	I		I
Default	0			0	0	0		0

VARIABLE	DESCRIPTION

ADPFREQ

Time interval between adaptive refinements, see Figures 12-2 and 12-1.

ADPTOL

Adaptive error tolerance in degrees for ADPOPT set to 1 or 2 below. If ADPOPT is set to 8, ADPTOL is the characteristic element size.

ADPOPT

Adaptive options:

EQ.1: angle change in degrees per adaptive refinement relative to the surrounding shells for each shell to be refined.

EQ.2: total angle change in degrees relative to the surrounding shells for each shell to be refined. For example, if the adptol = 5 degrees, the shell will be refined to the second level when the total angle change reaches 5 degrees. When the angle change is 10 degrees the shell will be refined to the third level.

EQ.4: adapts when the shell error in the energy norm, Δe, exceeds ADPTOL/100 times the mean energy norm within the part, which is estimated as:

$$\Delta e = \left(\int_{\Omega_e} \frac{\|\Delta\sigma\|^2}{E} d\Omega \right)^{1/2}$$

where E is Young's modulus. The error of the stresses $\Delta\sigma$ is defined as the difference between the the recovered solution $\sigma^\star$ and the numerical solution, σ^h i.e. $\Delta\sigma \equiv \sigma^\star - \sigma^h$. Various recovery techniques for $\sigma^\star$ and error estimators for Δe are defined by ADPERR. This options works for shell types 2, 4, 16, 18, 20.

EQ.7: 3D r-adaptive remeshing for solid elements. Solid element type 13, a tetrahedron, and 3-D EFG type 41 and 42, are used in the adaptive remeshing process. A com-

VARIABLE	DESCRIPTION
	pletely new mesh is generated which is initialized from the old mesh using a least squares approximation. The mesh size is currently based on the minimum and maximum edge lengths defined on the *CONTROL_REMESHING keyword input. This option remains under development, and, we are not sure of its reliability on complex geometries. EQ.8: 2D *r*-adaptive remeshing for axisymmetric and plane strain continuum elements. A completely new mesh is generated which is initialized from the old mesh using a least squares approximation. The mesh size is currently based on the value, ADPTOL, which gives the characteristic element size. This option is based on earlier work by Dick and Harris [1992]. If ADPTOL is negative, then self-contacting material will not be merged together. The self-merging is often preferred since it eliminates sharp folds in the boundary; however, if the sharp fold is being simulated unexpected results are generated.
MAXLVL	Maximum number of refinement levels. Values of 1, 2, 3, 4, … allow a maximum of 1, 4, 16, 64, … shells, respectively, to be created for each original shell. The refinement level can be overridden by *DEFINE_BOX_ADAPTIVE, or *DEFINE_SET_ADAPTIVE.
TBIRTH	Birth time at which the adaptive remeshing begins, see Figures 12-2 and 12-1.
TDEATH	Death time at which the adaptive remeshing ends, see Figures 12-2 and 12-1.
LCADP	Adaptive interval is changed as a function of time given by load curve ID, LCADP. If this option is nonzero, the ADPFREQ will be replaced by LCADP. The *x*-axis is time and the *y*-axis is the varied adaptive time interval.
IOFLAG	Flag to generate adaptive mesh at exit including *NODE, *ELEMENT_SHELL_THICKNESS, *BOUNDARY_option, and *CONSTRAINED_ADAPTIVITY, to be saved in the file, adapt.msh. EQ.1: generate *h*-adapted mesh.
ADPSIZE	Minimum shell size to be adapted based on element edge length. If undefined the edge length limit is ignored.

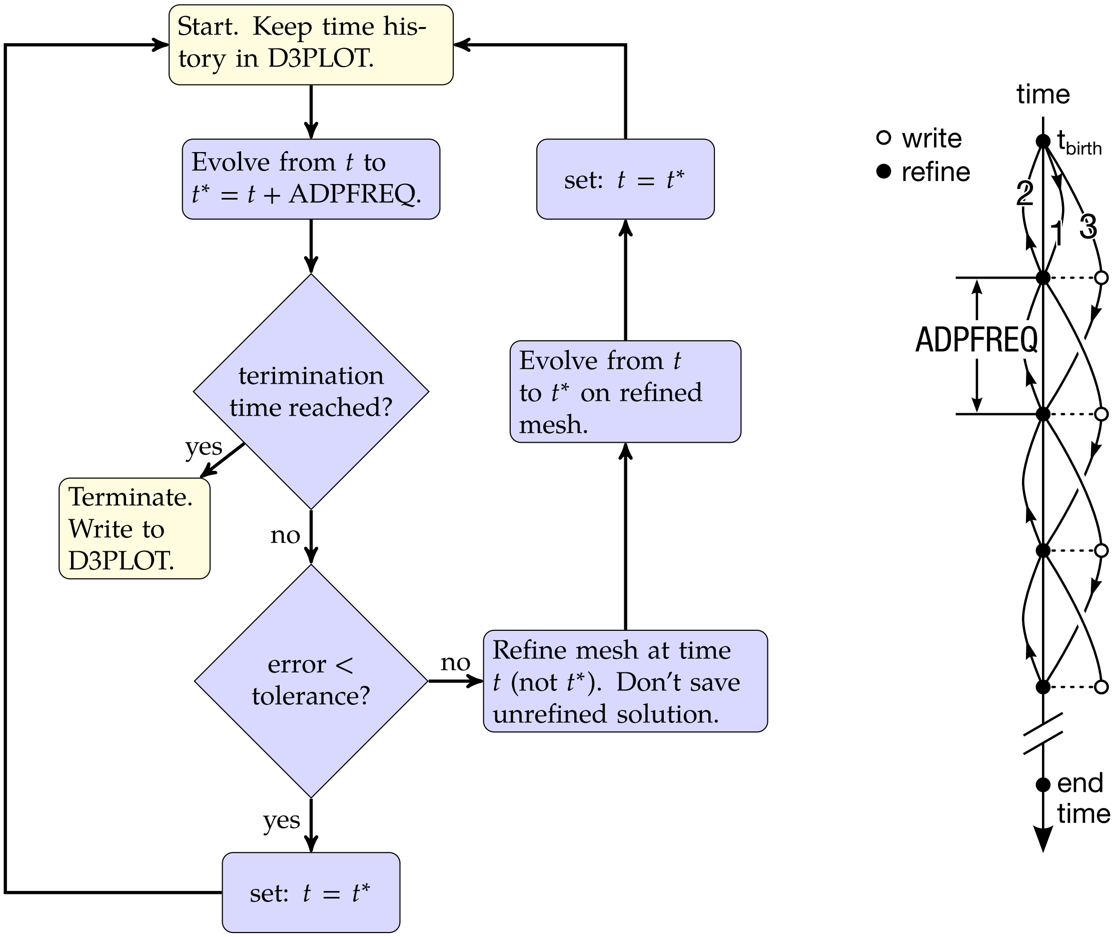

Start. Keep time history in D3PLOT.
Evolve from t to $t^* = t + \text{ADPFREQ}$.
terimination time reached?
yes
Terminate. Write to D3PLOT.
no
error < tolerance?
no
Refine mesh at time t (not t^*). Don't save unrefined solution.
Evolve from t to t^* on refined mesh.
set: $t = t^*$
yes
set: $t = t^*$
time
○ write
● refine
t_{birth}
2
1
3
ADPFREQ
end time

Figure 12-1. Flowchart for ADPASS = 0. While this option is *sometimes* more accurate, ADPASS = 1 is *much* less expensive and recommended when used *with* ADPENE.

VARIABLE **DESCRIPTION**

LT.0: absolute value defines the minimum characteristic element length to be adapted based on square root of the element area, i.e., instead of comparing the shortest element edge with ADPSIZE, it compares the square root of the element area with |ADPSIZE| whenever ADPSIZE is defined by a negative value.

ADPASS One or two pass flag for *h*-adaptivity:

EQ.0: two pass adaptivity as shown in Figure 12-2.

EQ.1: one pass adaptivity as shown in Figure 12-1.

IREFLG Uniform refinement level. A value of 1, 2, 3 … allow 4, 16, 64 … shells, respectively, to be created uniformly for each original shell. If negative, |IREFLG| is taken as a load curve ID. With the curve

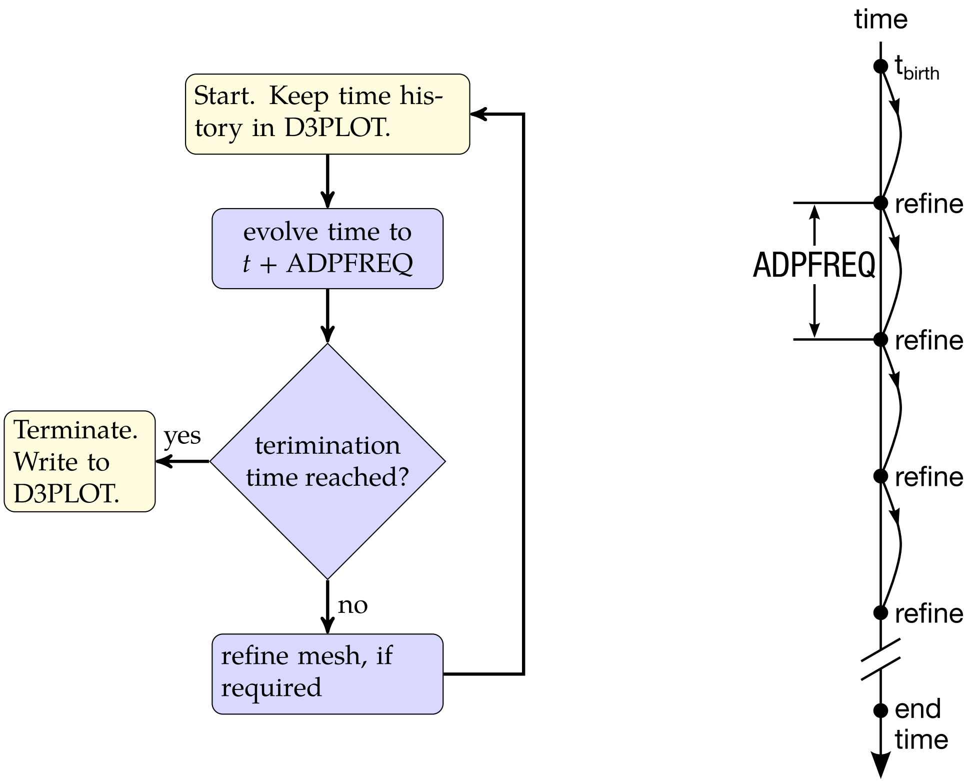

Start. Keep time history in D3PLOT.
evolve time to t + ADPFREQ
terimination time reached?
yes
Terminate. Write to D3PLOT.
no
refine mesh, if required
time
t_{birth}
refine
ADPFREQ
refine
refine
refine
end time

Figure 12-2. Flow chart for ADPASS = 1. This algorithm may be summarized as, "periodically refine" This method is recommended over ADPASS = 0 when used *with* ADPENE, which implements look ahead.

VARIABLE	DESCRIPTION
	option, the abscissa values define the refinement time, and the ordinate values define the minimum element size. Only one refinement level is performed per time step. An advantage of the load curve option is that the mesh is adapted to honor the minimum element size, but with the uniform option, IREFLG > 0, this is not possible. NOTE:If the element size defined with *DEFINE_CURVE is positive, the element size will override the element size defined with *CONTROL_ADAPTIVE and *DEFINE_SET_ADAPTIVE. Also, if the element size defined with *DEFINE_CURVE is negative the element size is used for refinement only.
ADPENE	For shell, *h*-adapt the mesh when the FORMING contact surfaces approach or penetrate the tooling surface depending on whether the value of ADPENE is positive (*approach*) or negative (*penetrates*), respectively. The tooling adaptive refinement is based on the curvature of the tooling. If ADPENE is positive the refinement generally occurs before contact takes place;

VARIABLE **DESCRIPTION**

consequently, it is possible that the parameter ADPASS can be set to 1 in invoke the one pass adaptivity.

For three dimensions *r*-adaptive solid remeshing (ADPOPT = 2 in *PART), the mesh refinement is based on the curvature of the tooling when ADPENE is positive. See Remark 6.

ADPTH

EQ.0.0: This parameter is ignored

GT.0.0: Absolute shell thickness level below which adaptive remeshing should began.

LT.0.0: Element thickness reduction ratio. If the ratio of the element thickness to the original element thickness is less than 1.0+ADPTHK, the element will be refined.

This option works only if ADPTOL is nonzero. If thickness based adaptive remeshing is desired without angle changes, then, set ADPTOL to a large angle.

MEMORY

This flag can have two meanings depending on whether the memory environmental variable is or is not set. The command "setenv LSTC_MEMORY auto" (or for bourne shell "export LSTC_MEMORY=auto") sets the memory environmental variable which causes LS-DYNA to expand memory automatically. Note that automatic memory expansion is not always 100% reliable depending on the machine and operating system level; consequently, it is not yet the default. To see if this is set on a particular machine type the command "env". If the environmental variable is not set then when memory usage reaches this percentage, MEMORY, further adaptivity is prevented to avoid exceeding the memory specified at execution time. Caution is necessary since memory usage is checked after each adaptive step, and, if the memory usage increases by more than the residual percentage, 100-PERCENT, the calculation will terminate.

If the memory environmental variable is set then when the number of words of memory allocated reaches or exceeds this value, MEMORY, further adaptivity is stopped.

ORIENT

This option applies to the FORMING contact option only. If this flag is set to one (1), the user orientation for the contact interface is used. If this flag is set to zero (0), LS-DYNA sets the global orientation of the contact surface the first time a potential contact is observed after the birth time. If slave nodes are found on both sides of the contact surface, the orientation is set based on the

VARIABLE DESCRIPTION

principle of "majority rules". Experience has shown that this principle is not always reliable.

MAXEL Adaptivity is stopped if this number of shells is exceeded.

IADPN90 Maximum number of shells covering 90 degree of radii. See Remark 5.

IADPGH Fission flag for neighbor splitting.

EQ.0: split all neighbor shells

EQ.1: do not split neighbor shells

NCFREQ Frequency of fission to fusion steps. For example, if NCFREQ = 4, then fusion will occur on the fourth, eighth, twelfth, etc., fission steps, respectively. If this option is used NCFREQ > 1 is recommended.

IADPCL Fusion will not occur until the fission level reaches IADPCL. Therefore, if IADPCL = 2, MAXLVL = 5, any shell can be split into 256 shells. If the surface flattens out, the number of elements will be reduced if the fusion option is active, i.e., the 256 elements can be fused and reduced to 16.

ADPCTL Adaptivity error tolerance in degrees for activating fusion. It follows the same rules as ADPOPT above.

CBIRTH Birth time for adaptive fusion. If ADPENE > 0, look-ahead adaptivity is active. In this case, fission, based on local tool curvature, will occur while the blank is still relatively flat. The time value given for CBIRTH should be set to a time later in the simulation after the forming process is well underway.

CDEATH Death time for adaptive fusion.

LCLVL Load curve ID of a curve that defines the maximum refinement level as a function of time

CNLA Limit angle for corner nodes. See Remark 7.

MMM2D If non-zero, common boundaries of all adapted materials will be merged. Only for 2D r-adaptivity

ADPERR 3-digit number, as *“XYY”*, where *“X”* and *“YY”* define the options for the recovery techniques and the error estimators,

VARIABLE **DESCRIPTION**

respectively,

For X:

EQ.0: superconvergent patch recovery (SPR) (default);

EQ.1: the least square fit of the stress to the nodes (Global L2);

EQ.2: error density SPR, as $\Delta\tilde{e} = \Delta e / \mathrm{Area}_{\mathrm{element}}$;

EQ.3: self-weighted SPR, as $\Delta\mathring{e} = \sqrt{\Delta e \times e}$

For YY:

EQ.00: energy norm (default)

EQ.01: Cauchy σ_x

EQ.02: σ_y

EQ.03: σ_z

EQ.04: τ_{xy}

EQ.05: τ_{yz}

EQ.06: τ_{zx}

EQ.07: effective plastic strain, $\varepsilon_{\mathrm{ep}}$

EQ.08: pressure

EQ.09: von Mises

EQ.10: principal deviator stress s11

EQ.11: S_{22}

EQ.12: S_{33}

EQ.13: Tresca

EQ.14: principal stress σ_{11}

EQ.15: σ_{22}

EQ.16: σ_{33}

EQ.20: user subroutine "uadpval" to extract the numerical solutions for recovery, and "uadpnorm" to provide an error estimator.

D3TRACE

Flag that is either 0 or 1. If set to 1 then a d3plot state will be output just before and after an adaptive step even though it may not be requested. The reason for wanting to do this is to allow the LS-PrePost particle trace algorithm to work in the case of

VARIABLE	DESCRIPTION
	adaptivity.
IFSAND	Set this flag to "1" for forming of sandwiched parts with adaptive blank mesh, see Remarks. Currently the adaptivity is limited to only one layer of solid element, and applies to ADPOPT = 1 and 2 only.

Remarks about 3D adaptivity:

1. **Restarting.** The d3dump and runrsf files contain all information necessary to restart an adaptive run. This did not work in version 936 of LS-DYNA.

2. **Related Field in *PART.** In order for this control card to work, the flag ADPOPT=1 must be set in the *PART definition. Otherwise, adaptivity will not function.

3. **Contact Types and Options.** In order for adaptivity to work optimally, the parameter SNLOG=1, must be set on Optional Control Card B in the *CONTACT Section. On disjoint tooling meshes the contact option *CONTACT_-FORMING_… is recommended.

4. **Root ID (RID) File.** A file named "adapt.rid" is left on disk after the adaptive run is completed. This file contains the root ID of all elements that are created during the calculation, and it does not need to be kept if it is not used in post-processing.

5. **Note About IADPN90 Field.** For all metal forming simulation, IADPN90 should be set to -1.

6. **Contact and ADPENE.** In three dimensions when ADPENE>0 it is presumed that the solid part to be adapted is on the slave side of a contact, and the "tooling", consisting of a shell surface, is on the master side of that same contact. ADPENE>0 represents a distance from the tooling surface within which the adapted mesh refinement of the slave part is influenced by the radius of curvature of the tooling surface. This feature is currently *unavailable* in SMP and SOFT=2 in *CONTACT.

Remarks about 2D *r*-adaptivity:

7. **CNLA Field.** In two dimensions *r*-adaptive remeshing, the generated new mesh should have a node at each corner so that corners are not smoothed. By default, the mesher will assume a corner wherever the interior angel between adjacent edges is less than 110 degrees. Setting CNLA larger than 110 enables

angles larger than 110 to be corners. Care should be taken to avoid an unnecessarily large value of CNLA as this may prevent the mesher from generating smooth meshes.

Remarks about mesh adaptivity for sandwiched parts (IFSAND):

8. Sandwiched parts (also called laminates) consist layers of solid elements (core) sandwiched by one layer of shell elements each on top and bottom surface of the solid elements, as shown in Figure 12-3. Common nodes are used for solid and shell interface. Currently mesh adaptivity is limited to only one layer of solid element with mesh refinements in-plane on both solids and shells.

   Note sandwiched parts can be trimmed by setting ITYP = 1 in keyword *CONTROL_FORMING_TRIMMING and with keyword *DEFINE_CURVE_TRIM. Trimming of sandwiched parts allows for multiple layers of solids.

   In a typical forming set up, the following cards need to be changed to activate the sandwiched part mesh adaptivity:

```
*CONTROL_ADAPTIVE
$# adpfreq    adptol    adpopt    maxlvl    tbirth    tdeath     lcadp    ioflag
   &adpfq 4.0000E+00         1         4     0.0001.0000E+20         0         0
$# adpsize    adpass    ireflg    adpene     adpth    memory    orient     maxel
   0.90000         1            10.00000     0.000         0         0         0
$# ladpn90    ladpgh    ncfred    ladpcl    adpctl    cbirth    cdeath     lclvl
        -1         0         0         1     0.000     0.0001.0000E+20         0
$                                                                         IFSAND
                                                                               1
*PART
Mid-core layer of solid elements
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
         1         1         1                                       1
Top layer of shell elements
       100       100         1                                       1
Bottom layer of shell elements
       101       100         1                                       1
```

Note IFSAND in *CONTROL_ADAPTIVE is set to “1” to activate the sandwich part adaptivity; ADPOPT under *PART are all set to “1” to activate the adaptivity.

Revision Information:

9. IFSAND is available starting in Rev 104365 in both SMP and MPP versions. Later revisions may include improvements.

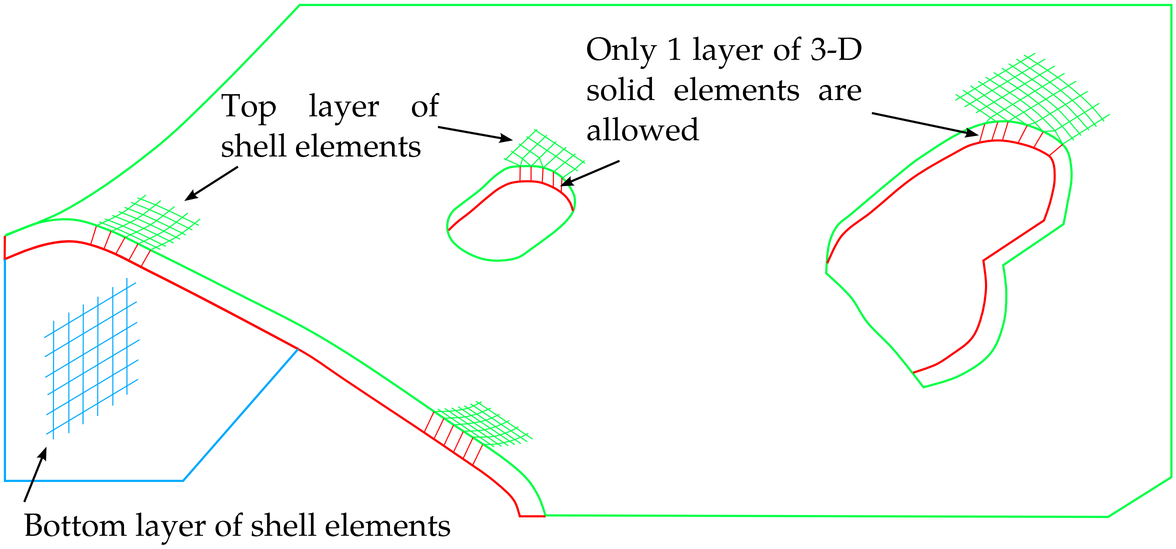

Top layer of shell elements
Only 1 layer of 3-D solid elements are allowed
Bottom layer of shell elements

Figure 12-3. Mesh adaptivity of sandwiched parts (**IFSAND**).

*CONTROL_ADAPTIVE_CURVE

Purpose: To refine the element mesh along a curve during or prior to sheet metal forming simulation. All curves defined by the keyword *DEFINE_CURVE_TRIM are used in the refinement. This option provides additional refinement to that created by *CONTROL_ADAPTIVE. Additionally, pre-mesh refinement along a curve with specific distance/range on both sides of the curve can be modeled when this keyword is used together with *DEFINE_CURVE_TRIM_3D (by activating the variable TCTOL). Lastly, the keyword can be used to refine mesh along a curve during trimming when used together with the keyword *ELEMENT_TRIM. This feature only applies to shell elements.

Card 1	1	2	3	4	5	6	7	8
Variable	IDSET	ITYPE	N	SMIN	ITRIOPT			
Type	I	I	I	F	I			

VARIABLE	DESCRIPTION

IDSET

Set ID

ITYPE

Set type:

EQ.1: IDSET is shell set ID.

EQ.2: IDSET is part set ID.

N

Refinement option:

EQ.1: Refine until there are no adaptive constraints remaining in the element mesh around the curve, subjected to the maximum refinement level of 5.

GT.1: Refine no more than N levels.

SMIN

If the element dimension is smaller than this value, do not refine.

ITRIOPT

Option to refine an enclosed area of a trim curve.

EQ.0: Refine the elements along the trim curve.

EQ.1: Refine the elements along the trim curve and enclosed by the trim curve.

Adaptive mesh refinement along a curve during a simulation:

In Figure 12-4, an example is shown to illustrate the mesh adaptivity along an enclosed curve. Since the mesh refinement is controlled by the refinement level "N" and smallest element size "SMIN", care should be taken so not too many elements are generated during the run.

A partial input example is listed below, where mesh will be refined by four levels, or to no smaller than 0.3mm element edge length, along both sides of the curve defined by IGES format file "adpcurves.iges".

```
*INCLUDE
drawn.dynain
*DEFINE_CURVE_TRIM_3D
$      TCID     TCTYPE       TFLG       TDIR      TCTOL
          1          2
adpcurves.iges
*CONTROL_ADAPTIVE_CURVE
$     IDSET      ITYPE          N       SMIN
          1          2          4        0.3
```

Since this method tends to create too many elements during refinement, the following feature was added to address the issue.

Adaptive mesh refinement along a curve in the beginning of a simulation:

When TCTOL is defined under the keyword *DEFINE_CURVE_TRIM_3D, it is used as a distance definition, and together with *CONTROL_ADAPTIVE_CURVE, the mesh will be refined in the beginning of a (flanging, etc.) simulation, along both sides of the defined curve, limited within the distance specified, as shown in Figures 12-5 and 12-6. In addition, this feature works with the option **3D** only. It is noted that the curve needs to be sufficient close to the part, and this can be accomplished in *LS-PrePost*4.0 under *GeoTol/Project/Closest Proj/Project to Element/By Part*. Furthermore, since the curve is often made from some feature lines of forming tools, it is important the curve is re-positioned closer to the blank, or better yet, is projected onto the blank; otherwise the refinement will not take place. A partial input example is listed below, where mesh will be refined within a range of 4.0mm, formed by 2.0mm distance (TCTOL = 2.0) of both sides of the curve, defined by file "adpcurves.iges". The maximum refine level is 4 and minimum element size allowed is 0.3mm.

```
*INCLUDE
drawn.dynain
*DEFINE_CURVE_TRIM_3D
$      TCID     TCTYPE       TFLG       TDIR      TCTOL
          1          2          0          0      2.000
adpcurves.iges
*CONTROL_ADAPTIVE_CURVE
$     IDSET      ITYPE          N       SMIN
          1          2          4        0.3
```

Mesh refinement along a curve is very useful during line die simulation. For example, in a flanging simulation, a trimmed blank, where it is mostly flat in the flanging break line in draw die, can be refined using a curve generated from the trim post radius. In *LS-PrePost 4.0*, the curve can be generated using *Curve/Spline/From Mesh/By Edge*, check *Prop*, and defining a large *Ang* to create a continuous curve along element edge. This curve can then be projected onto the blank mesh using *GeoTol/Project* feature, to be used as the curve file "adpcurves.iges" here. The mesh pre-refinement along curves are implemented in 'flanging' process starting in *LS-PrePost4.0* eZSetup for metal forming application. In *LS-PrePost4.3* eZSetup, improvements are made so adaptive mesh refinement along a curve can be made without the need to define any tools.

In Figures 12-7, 12-8, 12-9, 12-10and 12-11, mesh pre-refinement along a curve is demonstrated on the fender outer case. The effect of different TCTOL values on the mesh refinement is obvious.

The keyword *INCLUDE_TRIM is recommended to be used at all times to include the dynain file from a previous simulation, except in case where to-be-adapted sheet blank has no stress and strain information (no *INITIAL_STRESS_SHELL, and *INITIAL_-STRAIN_SHELL cards present in the sheet blank keyword or dynain file), then the keyword *INCLUDE must be used.

Adaptive mesh refinement along a curve during trimming:

When this keyword *ELEMENT_TRIM is present, this keyword is used to refine meshes during a trimming simulation. Coarse meshes along the trim curve can be refined prior to trimming, leaving a more detailed and distinctive trim edge. A partial example input deck is shown below:

```
*INCLUDE_TRIM
drawn.dynain
*ELEMENT_TRIM
         1
*DEFINE_CURVE_TRIM_NEW
$#    TCID    TCTYPE      TFLG      TDIR     TCTOL      TOLN    NSEED1    NSEED2
         1         2                   0     0.250         1
doubletrim.iges
*DEFINE_TRIM_SEED_POINT_COORDINATES
$    NSEED        X1        Y1        Z1        X2        Y2        Z2
         1  -184.565    84.755
*CONTROL_ADAPTIVE_CURVE
$#   IDSET     ITYPE         N      SMIN   ITRIOPT
         1         2         3       3.0         0
*CONTROL_CHECK_SHELL
$#    PSID    IFAUTO    CONVEX      ADPT    ARATIO     ANGLE      SMIN
         1         1         1         1      0.25     150.0      0.18
```

Where the keyword *ELEMENT_TRIM is used to define a deformable part set to be trimmed. The keyword *DEFINE_CURVE_TRIM_NEW is used to define the trim curve and type, along with trim tolerance, etc. The keyword *DEFINE_TRIM_SEED_POINT_COORDINATES is used to define a seed point coordinate located on the portion that remains after trimming. The keyword

*CONTROL_ADAPTIVE_CURVE is used to define the adaptive mesh refinement level and minimum element size along the trim curve. Finally, the keyword *CONTROL_CHECK_SHELL is used to repair and fix trimmed elements so they are suitable for next stage simulation. More details can be found in each of the corresponding keyword manual section.

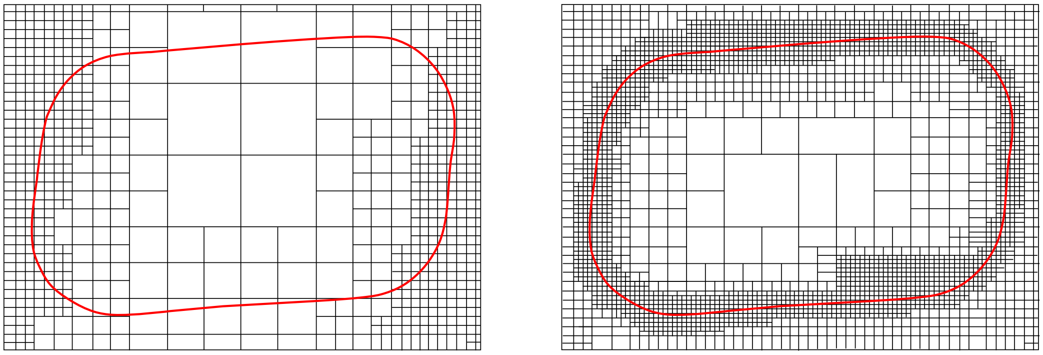


Figure 12-4. Mesh refinement along a curve

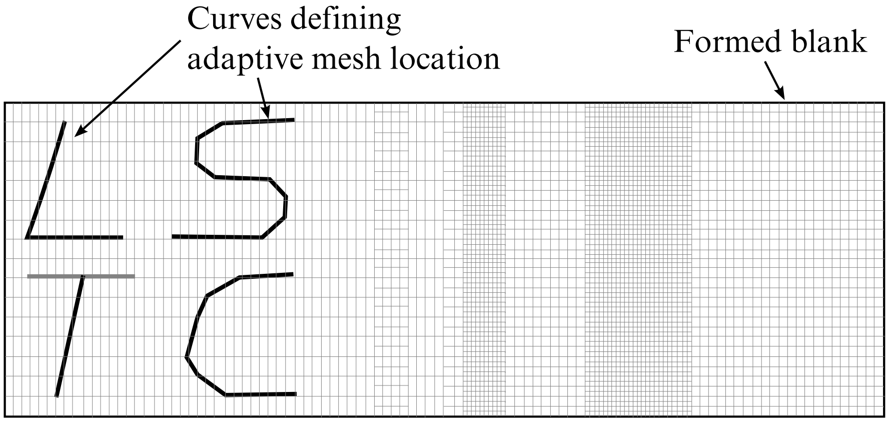


Figure 12-5. Curves can be discontinuous and in one IGES file.

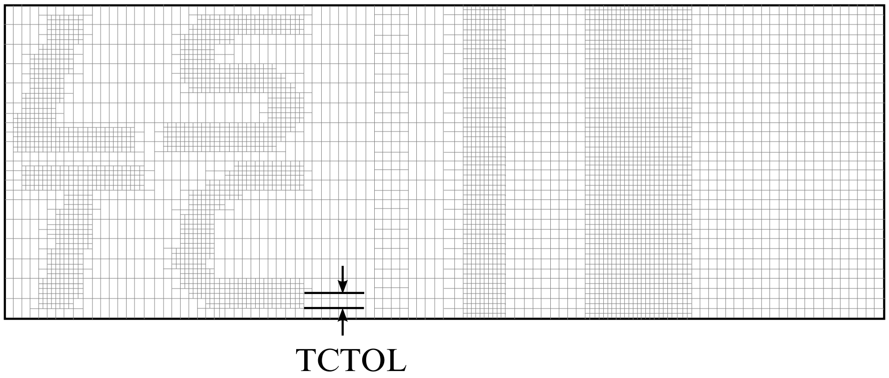


Figure 12-6. Define variable TCTOL to limit the mesh adaptivity area.

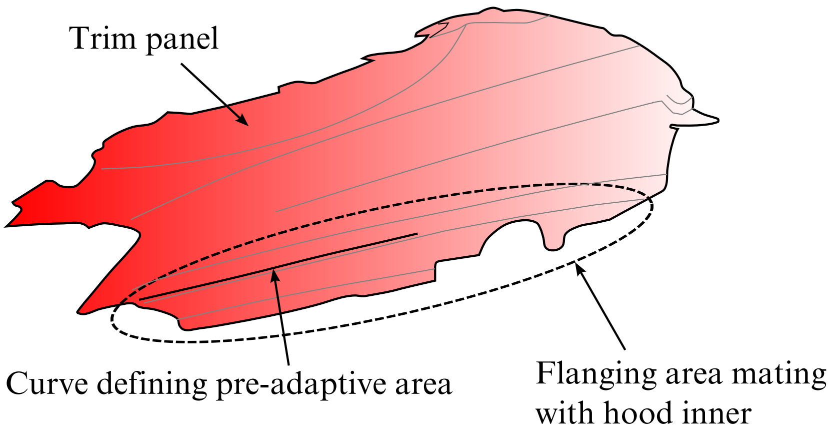

Trim panel
Curve defining pre-adaptive area
Flanging area mating with hood inner

Figure 12-7. A complex mesh refinement example (NUMISHEET2002 Fender).

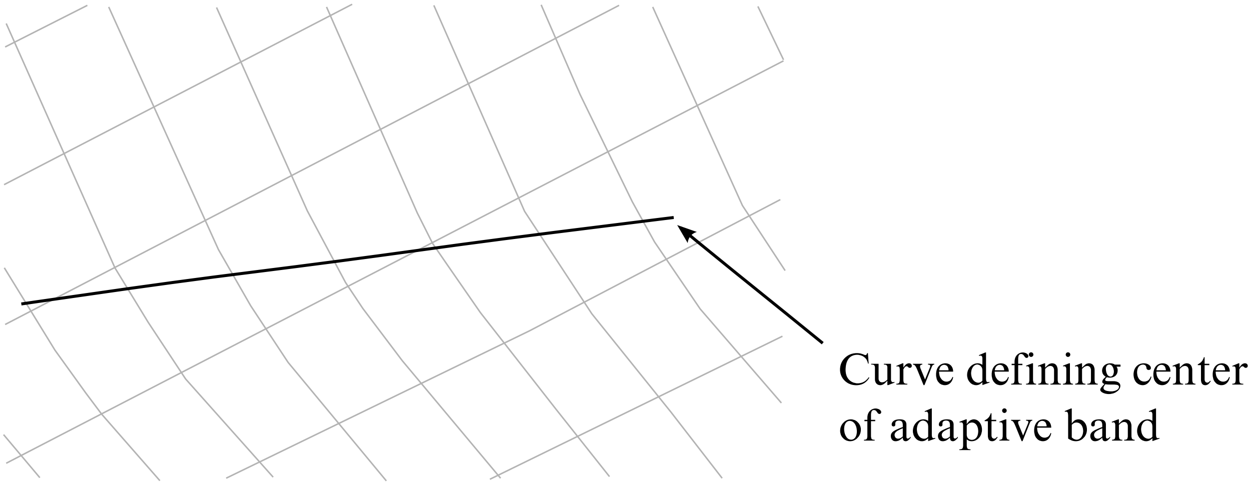


Figure 12-8. Original mesh with target curves defined.

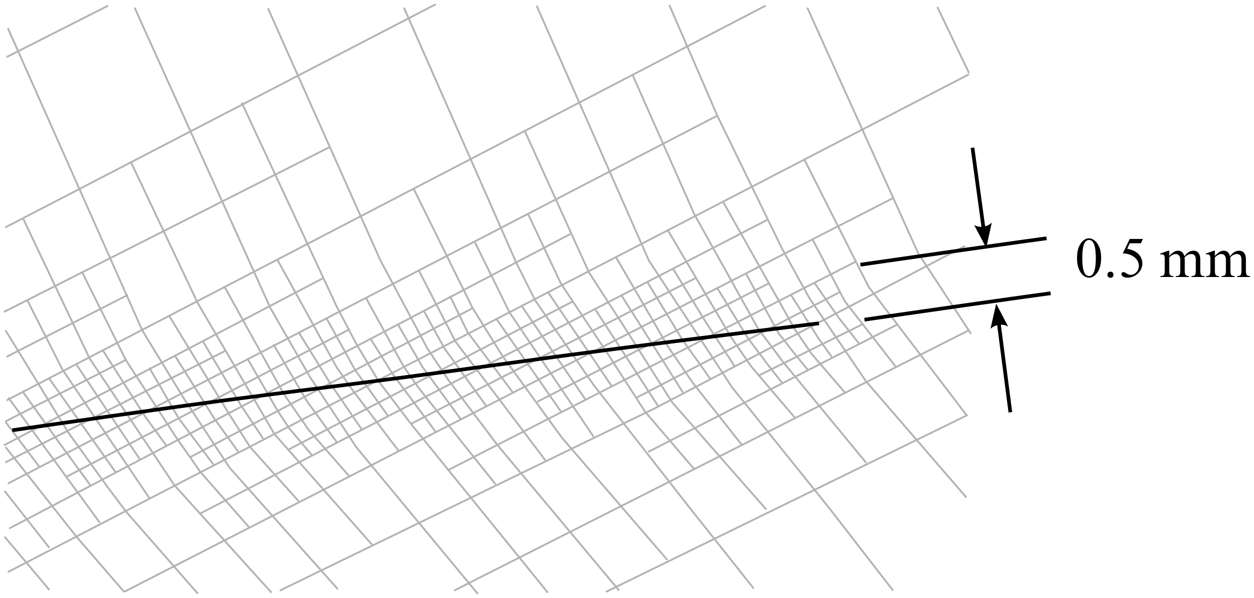


Figure 12-9. Mesh refinement with TCTOL = 0.5.

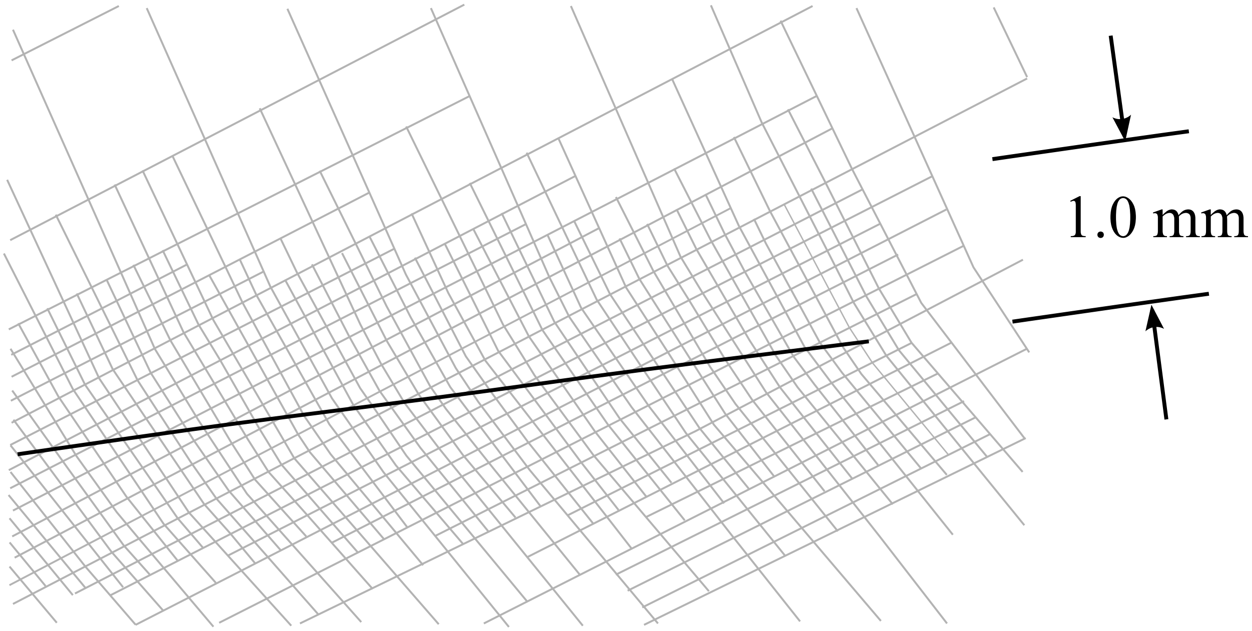


Figure 12-10. Mesh refinement with TCTOL = 1.0.

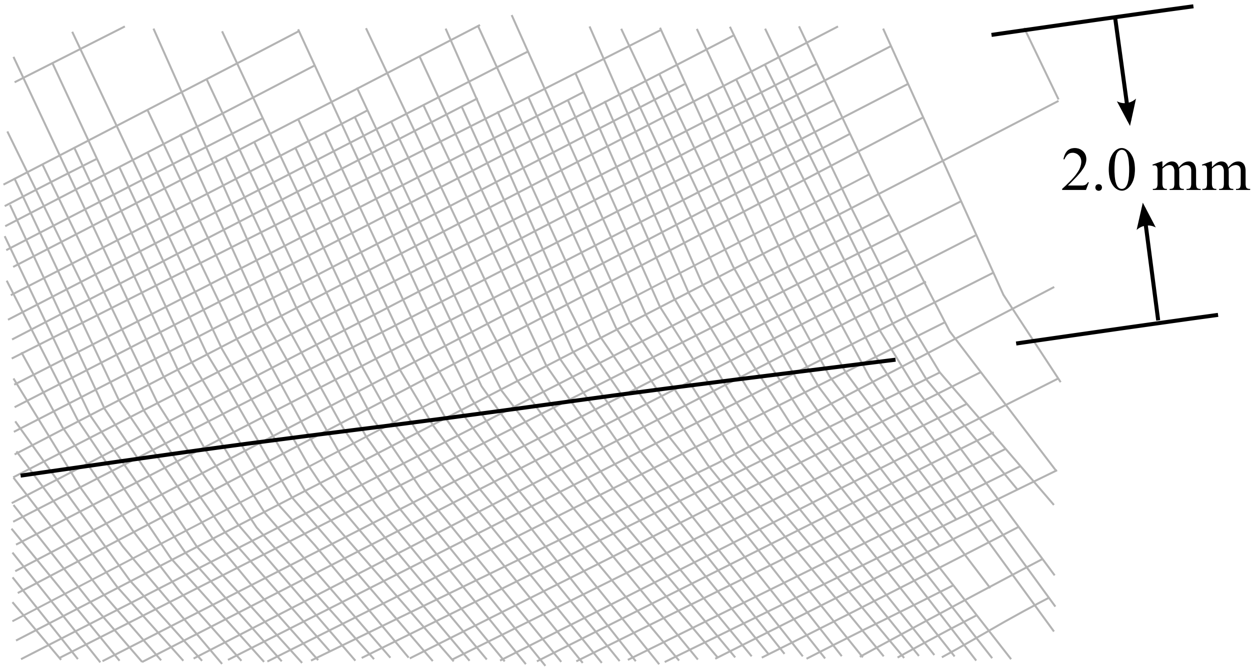


Figure 12-11. Mesh refinement with TCTOL = 2.0.

*CONTROL_ALE

Purpose: Set global control parameters for the Arbitrary Lagrangian-Eulerian (ALE) and Eulerian calculations. This command is required when solid element formulation 5, 6, 7, 11, or 12 is used. Parallel processing using SMP is not recommended when using these element formulations, rather it is better to use MPP for good parallel processing performance. See *CONTROL_MPP_DECOMPOSITION_DISTRIBUTE_ALE_ELEMENTS.

Card 1	1	2	3	4	5	6	7	8
Variable	DCT	NADV	METH	AFAC	BFAC	CFAC	DFAC	EFAC
Type	I	I	I	F	F	F	F	F
Default	1	0	1	0	0	0	0	0

Card 2	1	2	3	4	5	6	7	8
Variable	START	END	AAFAC	VFACT	PRIT	EBC	PREF	NSIDEBC
Type	F	F	F	F	F	I	F	I
Default	0	10^{20}	1	10^{-6}	0.0	0	0.0	none

This card is optional.

Card 3	1	2	3	4	5	6	7	8
Variable	NCPL	NBKT	IMASCL	CHECKR	BEAMIN	MMGPREF	PDIFMX	DTMUFAC
Type	I	I	I	F	F	I	F	F
Default	1	50	0	0.0	0.0	0	0.0	0.0

This card is optional.

Card 4	1	2	3	4	5	6	7	8
Variable	OPTIMPP							
Type	I							
Default	0							

VARIABLE	DESCRIPTION

DCT
Flag to invoke alternate advection logic. Formerly flag to control default continuum treatment:

NE.-1: Use default advection logic.

EQ.-1: Use alternate advection logic; generally recommended, especially for simulation of explosives (see Remark 9).

NADV
Number of cycles between advections (almost always set to 1).

METH
Advection method:

EQ.1: Donor cell with Half Index Shift (HIS), first order accurate.

EQ.2: Van Leer with HIS, second order accurate.

EQ.-2: Van Leer with HIS: Additionally,the monotonicity condition is relaxed during advection process to better preserve *MAT_HIGH_EXPLOSIVE_BURN material interfaces. (See Remark 10)

EQ.3: Donor cell with HIS modified to conserve total energy over each advection step, in contrast to METH = 1 which conserving internal energy (See Remark 5).

AFAC
ALE smoothing weight factor - Simple average:

EQ.-1: turn smoothing off: (See Remark 6).

BFAC
ALE smoothing weight factor – Volume weighting

CFAC
ALE smoothing weight factor – Isoparametric

DFAC
ALE smoothing weight factor – Equipotential

VARIABLE	DESCRIPTION
EFAC	ALE smoothing weight factor – Equilibrium
START	Start time for ALE smoothing or start time for ALE advection if smoothing is not used.
END	End time for ALE smoothing or end time for ALE advection if smoothing is not used.
AAFAC	ALE advection factor (donor cell options, default = 1.0)
VFACT	Volume fraction limit for stresses in single material and void formulation. All stresses are set to zero for elements with lower volume fraction than VFACT. EQ.0.0: set to default 10^{-6}
PRIT	A flag to turn on or off the pressure equilibrium iteration option for multi-material elements (See Remark 1). EQ.0: Off (default) EQ.1: On
EBC	Automatic Eulerian boundary condition (See Remark 2). EQ.0: Off EQ.1: On with stick condition EQ.2: On with slip condition
PREF	Reference pressure to compute the internal forces. (See Remark 3).
NSIDEBC	A node set ID (NSID) which is to be excluded from the EBC constraint.
NCPL	Number of Lagrangian cycles between coupling calculations. This is typically done every cycle; therefore, its default is 1. This is on optional card 3.

VARIABLE **DESCRIPTION**

NBKT

Number of Lagrangian cycles between global bucket-sort searches to locate the position of the Lagrangian structure (mesh) relative to the ALE fluid (mesh). Default is 50. This is on optional card 3.

LT.0: |NBKT| is a *DEFINE_CURVE ID defining a table: time vs NBKT as defined above

EQ.0: (Default) NBKT = 50: If the mesh is moving, NBKT is adapted for the buckets to follow the mesh more closely

GT.0: NBKT remains constant.

IMASCL

A flag for turning ON/OFF mass scaling for ALE parts. The global mass scaling control (parameter DT2MS under *CONTROL_TIMESTEP card) must be ON. If the run dt is lower than the mass scaling dt, then IMASCL has the following effects:

EQ.0: (Default) No mass scaling for ALE parts. Print out maximum 20 warnings.

EQ.1: No mass scaling for ALE parts. Stop the run.

EQ.2: Do mass scaling for ALE parts (the result may not be correct due to this scaling).

EQ.3: No mass scaling for ALE parts. Timestep is taken as the minimum of the ALE timestep and DT2MS.

CHECKR

A parameter for reducing or eliminating an ALE pressure locking pattern. It may range from 0.01 to 0.1 (See Remark 4).

BEAMIN

Flag to align the dynamics of plain strain and axisymmetric beams in 2D FSI ALE models to their shell counterparts in 3D FSI ALE models:

EQ.0.0: Off (default)

EQ.1.0: On

VARIABLE	DESCRIPTION
MMGPREF	MMGPREF selects the method that is used to include a reference pressure in a calculation involving ALE multi-material groups (See Remark 3). LT.0: |MMGPREF| is the id of a table defined by *DEFINE_CURVE where the abscissas are the multi-material group ids and the ordinates are the reference pressures. If a multi-material group is not in the table, its reference pressure is default to PREF. For situations in which the reference pressures are time dependent *DEFINE_TABLE should be used instead of *DEFINE_CURVE. The table should consist of a set of curves indexed by group ID that encode reference pressure as a function of time. If n groups need reference pressure histories, *DEFINE_TABLE will have n lines followed by n corresponding *DEFINE_CURVE. EQ.0: Off (default). EQ.1: Obsolete: Use MMGPREF < 0 instead EQ.2: Obsolete: Use MMGPREF < 0 instead
PDIFMX	Maximum of pressure difference between neighboring ALE elements under which the stresses are zeroed out: EQ.0: Off (default) GT.0: On
DTMUFAC	Scale a time step called DTMU that depends on the dynamic viscosity μ, the initial density ρ, and an element characteristic length ℓ: $$\mathrm{DTMU} = \frac{\rho\ell^2}{2\mu}$$ DTMU is emitted by the element to the solver as an element time step, thereby making DTMU an upper bound on the global time step. EQ.0: Off (default) GT.0: On

VARIABLE	DESCRIPTION
OPTIMPP	Optimize the MPP communications in the penalty coupling (*CONSTRAINED_LAGRANGE_IN_SOLID, CTYPE = 4) and group ALE parts together for the element processing. EQ.0: Off (default) EQ.1: On

Remarks:

1. **The PRIT Field.** Most of the fast transient applications do not need this feature. It could be used in specific slow dynamic problems for which material constitutive laws with very different compressibility are linear and the stresses in multi-material elements require to be balanced.

2. **The EBC Field.** This option, used for EULER formulations. It automatically defines velocity boundary condition constraints for the user. The constraints, once defined, are applied to all nodes on free surfaces of an Eulerian domain. For problems where the normal velocity of the material at the boundary is zero such as injection molding problems, the automatic boundary condition parameter is set to 2. This will play the same role as the nodal single point constraint. For EBC = 1, the material velocity of all free surface nodes of an Eulerian domain is set to zero.

3. **The PREF Field.** The reference pressure PREF is subtracted from the stresses before computing the internal forces. Thus PREF is equivalent to applying a *LOAD_SEGMENT card to balance the internal pressure along the ALE mesh boundaries. PREF is applied to all the materials in the ALE mesh. So, before the subtraction for MMGPREF > 0, PREF is added to the stresses of some materials.

   On another hand, MMGPREF < 0 subtracts a reference pressure depending on ALE MMG ID. The shift of the stresses by PREF is not necessary (and so it can not be seen in the LS-PrePost fringe of the pressures). For example, if a model has 3 ALE groups: air with an initial pressure of 1.0 bar, an explosive material, and water, the reference pressure of the first group would be 1.0 bar whereas the other groups would have none. In that case, PREF = 0.0 bar and MMGPREF = -LCID where LCID is the id of the following table:

```
*DEFINE_CURVE
lcid
1,1.0
2,0.0
3,0.0
```

4. **CHECKR Field for One Point Integration.** Due to one point integration, ALE elements may experience a spatial instability in the pressure field referred to as checker boarding. CHECKR is a scale for diffusive flux calculation to alleviate this problem.

5. **METH=3 for Conserving Total Energy.** Generally, it is not possible to conserve both momentum and kinetic energy (KE) at the same time. Typically, internal energy (IE) is conserved and KE may not be. This may result in some KE loss (hence, total energy loss). For many analyses this is tolerable, but for airbag application, this may lead to the reduction of the inflating potential of the inflator gas. METH=3 tries to eliminate this loss in KE over the advection step by storing any loss KE under IE, thus conserving total energy of the system.

6. **Smoothing Factors.** All the smoothing factors (AFAC, BFAC, CFAC, DFAC, EFAC) are generally most applicable to ELFORM = 5 (single material ALE formulation). The ALE smoothing feature is *not* supported by MPP versions.

7. **First Pass Recommendations.** Although this card has many parameters, only a few are required definitions. Typically, one can try setting NADV=1, METH=1, AFAC=-1 and the rest as "0" as a starting point. Sometimes when needed, PREF should be defined. This is adequate for most cases. Sometimes it may be appropriate to fine-tune the model by changing METH to 2 or 3 depending on the physics.

8. **Pressure Checker Boarding.** Because the internal forces are located at the nodes, while the pressure is stored at the element center, sometimes a "checkerboard pattern" arises in the pressure distribution. It is a kind of locking effect that normally occurs only in problems having very small volumetric strains, i.e., at small pressures. "CHECKR" is designed for alleviating this problem.

9. **The DCT Field.** Starting with the R5 the DCT field can be used to invoke an alternate advection scheme. DCT=-1 is recommended over the default scheme, especially for simulating explosives and includes the following major changes:

   a) Relaxes an artificial limit on the expansion ratio limit. The default limit improves stability in some situations but can overestimate the explosive impulse.

   b) Corrects redundant out-flux of material at corner elements. The redundancy can lead to negative volume.

   c) Removes several artificial constraints in the advection which were originally implemented to assist in stability but are no longer needed.

10. **METH = -2.** The METH = -2 advection type is the same as METH = 2 with only one exception. It employs a looser constraint on monotonicity requirement during ALE advection. When METH = 2, for each advection process along three directions (front/back, top/bottom, left/right), the maximum/minimum values for advected history variables in the three elements along that direction are capped. METH = -2 relaxed the monotonicity condition so that the advected value is capped at the maximum/minimum value in the element itself and its neighboring 26 elements. This option, in certain conditions, can better preserve the material interface for materials defined with *MAT_HIGH_EXPLOSIVE_BURN.

*CONTROL_BULK_VISCOSITY

Purpose: Reset the default values of the bulk viscosity coefficients globally. This may be advisable for shock wave propagation and some materials. Bulk viscosity is used to treat shock waves. A viscous term q is added to the pressure to smear the shock discontinuities into rapidly varying but continuous transition regions. With this method the solution is unperturbed away from a shock, the Hugoniot jump conditions remain valid across the shock transition, and shocks are treated automatically.

Card 1	1	2	3	4	5	6	7	8
Variable	Q1	Q2	TYPE	BTYPE				
Type	F	F	I	I				
Default	1.5	.06	1	0				

VARIABLE	DESCRIPTION

Q1 Default quadratic viscosity coefficient.

Q2 Default linear viscosity coefficient.

TYPE Default bulk viscosity type, IBQ (Default = 1)

EQ.-2: standard (also types 2, 4, 10, 16, and 17). With this option the internal energy dissipated by the viscosity in the shell elements is computed and included in the overall energy balance.

EQ.-1: standard (also types 2, 4, 10, 16, and 17 shell elements). The internal energy is not computed in the shell elements.

EQ.+1: standard: Solid elements only and internal energy is always computed and included in the overall energy balance.

EQ.+2: Richards-Wilkins: Two-dimensional plane strain and axisymmetric solid elements only. Internal energy is always computed and included in the overall energy balance.

VARIABLE	DESCRIPTION
BTYPE	Beam bulk viscosity type (Default = 0) EQ.0: The bulk viscosity is turned off for beams. EQ.1: The bulk viscosity is turned on for beam types 1 and 11. The energy contribution is not included in the overall energy balance. EQ.2: The bulk viscosity is turned on for beam type 1 and 11. The energy contribution is included in the overall energy balance.

Remarks:

The bulk viscosity creates an additional additive pressure term given by:

$$q = \begin{cases} \rho l\left(Q_1 l\dot{\varepsilon}_{kk}^2 - Q_2 a\dot{\varepsilon}_{kk}\right) & \dot{\varepsilon}_{kk} < 0 \\ 0 & \dot{\varepsilon}_{kk} \geq 0 \end{cases}$$

where Q_1 and Q_2 are dimensionless input constants which default to 1.5 and .06, respectively, and l is a characteristic length given as the square root of the area in two dimensions and as the cube root of the volume in three, a is the local sound speed, Q_1 defaults to 1.5 and Q_2 defaults to .06. See Chapter 21 in the LS-DYNA Theory Manual for more details.

The Richards-Wilkins, see [Richards 1965, Wilkins 1976], bulk viscosity considers the directional properties of the shock wave. This has the effect of turning off the bulk viscosity in converging geometries minimizing the effects of “q-heating”. The standard option is active whenever the volumetric strain rate is undergoing compression even though no shock waves are present.

*CONTROL_CHECK_SHELL

Purpose: Check for various problems in the mesh.

Part cards. Include one card for each part or part set to be checked. The next keyword ("*") card terminates this input.

Card 1	1	2	3	4	5	6	7	8
Variable	PSID	IFAUTO	CONVEX	ADPT	ARATIO	ANGLE	SMIN	
Type	I	I	I	I	F	F	F	
Default	0	0	1	1	0.25	150.0	0.0	

VARIABLE	DESCRIPTION
PSID	Part or part set ID to be checked: EQ.0: do not check GT.0: part ID LT.0: part set ID
IFAUTO	Flag to automatically correct bad elements: EQ.0: write warning message only EQ.1: fix bad element, write message
CONVEX	Check element convexity (internal angles less than 180 degrees) EQ.0: do not check EQ.1: check
ADPT	Check adaptive constraints EQ.0: do not check EQ.1: check
ARATIO	Minimum allowable aspect ratio. Elements which do not meet minimum aspect ratio test will be treated according to IFAUTO above.
ANGLE	Maximum allowable internal angle. Elements which fail this test will be treated according to IFAUTO above.

VARIABLE	DESCRIPTION
SMIN	Minimum element size. Elements which fail this test will be treated according to IFAUTO above.

Remarks:

1. For the SHELL option, shell element integrity checks which have been identified as important in metal forming applications are performed. These checks can improve springback convergence and accuracy. This option will repair bad elements created, for example, during trimming operations.

2. If the convexity test is activated, all failed elements will be fixed regardless of IFAUTO.

3. In addition to illegal constraint definitions (slave which is also a master), checks are performed for mesh connectivities which have been found to cause convergence trouble in implicit springback applications.

4. Variable SMIN should be set to 1/4 to 1/3 of smallest pre-trim element length. In an example below, smallest element length pre-trim is 0.6mm, which makes SMIN to be 0.18:

```
*CONTROL_CHECK_SHELL
1,1,1,1,0.25,150.0,0.18
$ smin=(0.25~0.3)*smallest pre-trim element length, which is ~0.6 mm.
```

5. Shell checking is done during the input phase (in sprinback input deck) in LS-DYNA R5 Revision 63063 and prior releases. After the Revision, it is done after trimming is completed. Therefore the keyword should be included in a trimming input deck.

*CONTROL_COARSEN

Purpose: Adaptively de-refine (coarsen) a shell mesh by selectively merging four adjacent elements into one. Adaptive constraints are added and removed as necessary.

Card 1	1	2	3	4	5	6	7	8
Variable	ICOARSE	ANGLE	NSEED	PSID	SMAX			
Type	I	F	I	I	F			
Default	0	none	0	0	0			

Card 2	1	2	3	4	5	6	7	8
Variable	N1	N2	N3	N4	N5	N6	N7	N8
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

VARIABLE	DESCRIPTION
ICOARSE	Coarsening flag: EQ.0: do not coarsen (default) EQ.1: coarsen mesh at beginning of simulation for forming model EQ.2: coarsen mesh at beginning of simulation for crash model
ANGLE	Allowable angle change between neighboring elements. Adjacent elements which are flat to within ANGLE degrees are merged. (Suggested starting value = 8.0 degrees)
NSEED	Number of seed nodes (optional). EQ.0: use only automatic searching. GT.0: the number of seed nodes with which to supplement the search algorithm. See Remark 2. NSEED must be an integer less than or equal to 8.

VARIABLE	DESCRIPTION
PSID	Part set ID. All the parts defined in this set will be prevented from been coarsened.
SMAX	Maximum element size. For ICOARSE = 2, no elements larger than this size will be created.
N1, …, N8	Optional list of seed node IDs for extra searching. If no seed nodes are specified, leave card 2 blank.

Remarks:

1. Coarsening is performed at the start of a simulation. The first plot state represents the coarsened mesh. By setting the termination time to zero and including the keyword *INTERFACE_SPRINGBACK_LSDYNA a keyword input deck can be generated containing the coarsened mesh.

2. By default, an automatic search is performed to identify elements for coarsening. In some meshes, isolated regions of refinement may be overlooked. Seed nodes can be identified in these regions to assist the automatic search. Seed nodes identify the central node of a four-element group which is coarsened into a single element if the angle criterion is satisfied.

3. The keyword *DEFINE_BOX_COARSEN can be used to indicate regions of the mesh which are protected from coarsening.

*CONTROL_CONTACT

Purpose: Change defaults for computation with contact surfaces.

Card 1	1	2	3	4	5	6	7	8
Variable	SLSFAC	RWPNAL	ISLCHK	SHLTHK	PENOPT	THKCHG	ORIEN	ENMASS
Type	F	F	I	I	I	I	I	I
Default	.1	none	1	0	1	0	1	0
Remarks			3					

Card 2	1	2	3	4	5	6	7	8
Variable	USRSTR	USRFRC	NSBCS	INTERM	XPENE	SSTHK	ECDT	TIEDPRJ
Type	I	I	I	I	F	I	I	I
Default	0	0	10-100	0	4.0	0	0	0

Remaining cards are optional.†

The optional cards apply only to the following contact types:

*SINGLE_SURFACE

*AUTOMATIC_GENERAL

*AUTOMATIC_SINGLE_SURFACE

*AUTOMATIC_NODES_TO_…

*AUTOMATIC_SURFACE_…

*AUTOMATIC_ONE_WAY_…

*ERODING_SINGLE_SURFACE.

The friction coefficients SFRIC, DFRIC, EDC, and VFC are active only when *PART_-CONTACT is invoked with FS = -1 in *CONTACT, and the corresponding frictional coefficients in *PART_CONTACT are set to zero. This keyword's TH, TH_SF, and PEN_SF override the corresponding parameters in *CONTACT, but will not override corresponding nonzero parameters in *PART_CONTACT.

Card 3	1	2	3	4	5	6	7	8
Variable	SFRIC	DFRIC	EDC	VFC	TH	TH_SF	PEN_SF	PTSCL
Type	F	F	F	F	F	F	F	F
Default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Card 4	1	2	3	4	5	6	7	8
Variable	IGNORE	FRCENG	SKIPRWG	OUTSEG	SPOTSTP	SPOTDEL	SPOTHIN	
Type	I	I	I	I	I	I	F	
Default	0	0	0	0	0	0	inactive	

Card 5	1	2	3	4	5	6	7	8
Variable	ISYM	NSEROD	RWGAPS	RWGDTH	RWKSF	ICOV	SWRADF	ITHOFF
Type	I	I	I	F	F	I	F	I
Default	0	0	0	0.	1.0	0	0.	0

Card 6	1	2	3	4	5	6	7	8
Variable	SHLEDG	PSTIFF	ITHCNT	TDCNOF	FTALL		SHLTRW	IGACTC
Type	I	I	I	I	I		F	I
Default	0	0	0	0	0		0.	0

VARIABLE **DESCRIPTION**

SLSFAC

Scale factor for sliding interface penalties, SLSFAC:

EQ.0: default = .1.

RWPNAL

Scale factor for rigid wall penalties, which treat nodal points interacting with rigid walls, RWPNAL. The penalties are set so that an absolute value of unity should be optimal; however, this penalty value may be very problem dependent. If rigid/deformable materials switching is used, this option should be used if the switched materials are interacting with rigid walls.

LT.0.0: all nodes are treated by the penalty method. This is required for implicit calculations. Since seven (7) variables are stored for each slave node, only the nodes that may interact with the wall should be included in the node list.

EQ.0.0: the constraint method is used and nodal points which belong to rigid bodies are not considered.

GT.0.0: rigid bodies nodes are treated by the penalty method and all other nodes are treated by the constraint method.

ISLCHK

Initial penetration check in contact surfaces with indication of initial penetration in output files (see Remark 3):

EQ.0: the default is set to 1,

EQ.1: no checking,

EQ.2: full check of initial penetration is performed.

VARIABLE **DESCRIPTION**

SHLTHK

Flag for consideration of shell thickness offsets in non-automatic surface-to-surface and non-automatic nodes-to-surface type contacts. Shell thickness offsets are always included in single surface, constraint-based, automatic surface-to-surface, and automatic nodes-to-surface contact types (See Remarks 1 and 2):

EQ.0: thickness is not considered,

EQ.1: thickness is considered but rigid bodies are excluded,

EQ.2: thickness is considered including rigid bodies.

PENOPT

Penalty stiffness value option. For default calculation of the penalty value please refer to the LS-DYNA Theory Manual.

EQ.0: the default is set to 1,

EQ.1: minimum of master segment and slave node (default for most contact types),

EQ.2: use master segment stiffness (old way),

EQ.3: use slave node value,

EQ.4: use slave node value, area or mass weighted,

EQ.5: same as 4 but inversely proportional to the shell thickness. This may require special scaling and is not generally recommended.

Options 4 and 5 can be used for metal forming calculations.

THKCHG

Shell thickness changes considered in single surface contact:

EQ.0: no consideration (default),

EQ.1: shell thickness changes are included.

ORIEN

Optional automatic reorientation of contact interface segments during initialization. See Remark 4.

EQ.0: default is set to 1.

EQ.1: active for automated (part) input only. Contact surfaces are given by *PART definitions.

EQ.2: active for manual (segment) and automated (part) input.

EQ.3: inactive for non-forming contact.

EQ.4: inactive for *CONTACT_FORMING types and *CONTACT_DRAWBEAD.

VARIABLE **DESCRIPTION**

ENMASS

Treatment of the mass of eroded nodes in contact. This option affects all contact types where nodes are removed after surrounding elements fail. Generally, the removal of eroded nodes makes the calculation more stable; however, in problems where erosion is important the reduction of mass will lead to incorrect results. ENMASS is not supported when SOFT = 2 on optional card A.

EQ.0: eroding nodes are removed from the calculation.

EQ.1: eroding nodes of solid elements are retained and continue to be active in contact.

EQ.2: the eroding nodes of solid and shell elements are retained and continue to be active in contact.

USRSTR

Storage per contact interface for user supplied interface control subroutine, see Appendix F. If zero, no input data is read and no interface storage is permitted in the user subroutine. This storage should be large enough to accommodate input parameters and any history data. This input data is available in the user supplied subroutine.

USRFRC

Storage per contact interface for user supplied interface friction subroutine, see Appendix G. If zero, no input data is read and no interface storage is permitted in the user subroutine. This storage should be large enough to accommodate input parameters and any history data. This input data is available in the user supplied subroutine.

NSBCS

Number of cycles between contact searching using three dimensional bucket searches, defaults recommended. For Mortar contact (option MORTAR on the CONTACT card), the default is 100.

INTERM

Flag for intermittent searching in old surface-to-surface contact using the interval specified as NSBCS above:

EQ.0: off,

EQ.1: on.

VARIABLE **DESCRIPTION**

XPENE

Contact surface maximum penetration check multiplier. If the small penetration checking option, PENCHK, on the contact surface control card is active, then nodes whose penetration then exceeds the product of XPENE and the element thickness are set free, see *CONTACT_*OPTION*_…

EQ.0: default is set to 4.0.

SSTHK

Flag for using actual shell thickness in single surface contact logic-types 4, 13, 15 and 26. See Remarks 1 and 2.

EQ.0: Actual shell thickness is not used in the contacts. (default),

EQ.1: Actual shell thickness is used in the contacts. (sometimes recommended for metal forming calculations).

ECDT

Time step size override for eroding contact:

EQ.0: contact time size may control Dt.

EQ.1: contact is not considered in Dt determination.

TIEDPRJ

Bypass projection of slave nodes to master surface in types:

*CONTACT_TIED_NODES_TO_SURFACE

*CONTACT_TIED_SHELL_EDGE_TO_SURFACE

*CONTACT_TIED_SURFACE_TO_SURFACE

Tied interface options:

EQ.0: eliminate gaps by projection nodes,

EQ.1: bypass projection: Gaps create rotational constraints which can substantially affect results.

SFRIC

Default static coefficient of friction (see *PART_CONTACT)

DFRIC

Default dynamic coefficient of friction (see *PART_CONTACT)

EDC

Default exponential decay coefficient (see *PART_CONTACT)

VFC

Default viscous friction coefficient (see *PART_CONTACT)

TH

Default contact thickness (see *PART_CONTACT)

TH_SF

Default thickness scale factor (see *PART_CONTACT)

VARIABLE	DESCRIPTION
PEN_SF	Default local penalty scale factor (see *PART_CONTACT)
PTSCL	Scale factor on the contact stress exerted onto shells formulations 25, 26, and 27. When DOF = 3 the scale factor also applies to shell formulations 2, and 16.
IGNORE	Ignore initial penetrations in the *CONTACT_AUTOMATIC options. In the SMP contact this flag is not implement for the AUTOMATIC_GENERAL option. "Initial" in this context refers to the first timestep that a penetration is encountered. This option can also be specified for each interface on the third optional card under the keyword, *CONTACT. The value defined here will be the default. EQ.0: move nodes to eliminate initial penetrations in the model definition. EQ.1: allow initial penetrations to exist by tracking the initial penetrations. EQ.2: allow initial penetrations to exist by tracking the initial penetrations. However, penetration warning messages are printed with the original coordinates and the recommended coordinates of each slave node given.
FRCENG	Flag to activate the calculation of frictional sliding energy: EQ.0: do not calculate, EQ.1: calculate frictional energy in contact and store as "Surface Energy Density" in the binary INTFOR file. Convert mechanical frictional energy to heat when doing a coupled thermal-mechanical problem. When PKP_SEN = 1 on the keyword card *DATABASE_EXTENT_BINARY, it is possible to identify the energies generated on the upper and lower shell surfaces, which is important in metal forming applications. This data is mapped after each H-adaptive remeshing. EQ.2: Same as behavior as above (set to 1) except that frictional energy is *not* converted to heat.

VARIABLE **DESCRIPTION**

SKIPRWG

Flag not to display stationary rigid wall by default.

EQ.0: generate 4 extra nodes and 1 shell element to visualize stationary planar rigid wall.

EQ.1: do not generate stationary rigid wall.

OUTSEG

Flag to output each beam spot weld slave node and its master segment for contact type: *CONTACT_SPOTWELD into the d3hsp file.

EQ.0: no, do not write out this information.

EQ.1: yes, write out this information.

SPOTSTP

If a spot weld node or face, which is related to a *MAT_-SPOTWELD beam or solid element, respectively, cannot be found on the master surface, should an error termination occur?

EQ.0: no, silently delete the weld and continue,

EQ.1: yes, print error message and terminate,

EQ.2: no, delete the weld, print a message, and continue,

EQ.3: no, keep the weld. This is not recommended as it can lead to instabilities.

SPOTDEL

This option controls the behavior of spotwelds when the parent element erodes. When SPOTDEL is set to 1, the beam or solid spotweld is deleted and the tied constraint is removed when the parent element erodes. Parent element is the element to which the slave node is attached using the TIED interface. This option also works for SPRs, i.e. they automatically fail if at least one of the parent elements fails. To avoid instabilities, this option is recommended to be set to 1 for any situation in which the parent element is expected to erode.

EQ.0: no, do not delete the spot weld beam or solid element or SPR,

EQ.1: yes, delete the weld elements or SPRs when the attached shells on one side of the element fail.

On vector processors this option can significantly slow down the calculation if many weld elements fail since the vector lengths are reduced. On non-vector processors the cost-penalty is minimal.

VARIABLE **DESCRIPTION**

SPOTHIN

Optional thickness scale factor. If active, define a factor greater than zero, but less than one. Premature failure of spot welds can occur due to contact of the spot welded parts in the vicinity of the spot weld. This contact creates tensile forces in the spot weld.

Although this may seems physical, the compressive forces generated in the contact are large enough to fail the weld in tension before failure is observed in experimental test. With this option, the thickness of the parts in the vicinity of the weld are automatically scaled, the contact forces do not develop, and the problem is avoided. We recommend setting the IGNORE option to 1 or 2 if SPOTHIN is active. This option applies only to the AUTOMATIC_SINGLE_SURFACE option. See Remark 5.

ISYM

Symmetry plane option default for automatic segment generation when contact is defined by part ID's:

EQ.0: off,

EQ.1: do not include faces with normal boundary constraints (e.g., segments of brick elements on a symmetry plane).

This option is important to retain the correct boundary conditions in the model with symmetry.

NSEROD

Flag to use one-way node to surface erosion

EQ.0: use two-way algorithm

EQ.1: use one-way algorithm

RWGAPS

Flag to add rigid wall gap stiffness, see parameter RWGDTH below.

EQ.1: add gap stiffness

EQ.2: do not add gap stiffness

RWGDTH

Death time for gap stiffness. After this time the gap stiffness is no longer added.

RWKSF

Rigid wall penalty scale factor for contact with deformable parts during implicit calculations. This value is independent of SLSFAC and RWPNAL. If RWKSF is also specified in *RIGIDWALL_PLANAR, the stiffness is scaled by the product of the two values.

VARIABLE **DESCRIPTION**

ICOV

Invokes the covariant formulation of Konyukhov and Schweizerhof in the FORMING contact option. This option is available in the third revision of version 971, but is not recommended since it is still being implemented.

EQ.0: standard formulation (default)

EQ.1: covariant contact formulation.

SWRADF

Spot weld radius scale factor for neighbor segment thinning

EQ.0: neighbor segments not thinned (default)

GT.0: The radius of beam spot welds are scaled by SWRADF when searching for close neighbor segments to thin.

ITHOFF

Flag for offsetting thermal contact surfaces for thick thermal shells

EQ.0: No offset, if thickness is not included in the contact the heat will be transferred between the mid-surfaces of the corresponding contact segments (shells).

EQ.1: Offsets are applied so that contact heat transfer is always between the outer surfaces of the contact segments (shells).

SHLEDG

Flag for assuming edge shape for shells when measuring penetration. This is available for segment based contact (see SOFT on *CONTACT)

EQ.0: Shell edges are assumed round (default),

EQ.1: Shell edges are assumed square and are flush with the nodes

PSTIFF

Flag to choose the method for calculating the penalty stiffness. This is available for segment based contact (see SOFT on *CONTACT). See Remark 6.

EQ.0: Based on material density and segment dimensions (default),

EQ.1: Based on nodal masses.

VARIABLE **DESCRIPTION**

ITHCNT

Thermal contact heat transfer methodology

LT.0: conduction evevenly distributed (pre R4)

EQ.0: default set to 1

EQ.1: conduction weighted by shape functions, reduced intergration

EQ.2: conduction weighted by shape functions, full integration

TDCNOF

Tied constraint offset contact update option.

EQ.0: Update velocities and displacements from accelerations

EQ.1: Update velocities and accelerations from displacements. This option is recommended only when there are large angle changes where the default does not maintain a constant offset to a small tolerance. This latter option is not as stable as the default and may require additional damping for stability. See *CONTROL_BULK_VISCOSITY and *DAMPING_PART_STIFFNESS.

FTALL

Option to output contact forces to RCFORC for all 2 surface force transducers when the force transducer surfaces overlap. See Remark 7.

EQ.0: Output to the first force transducer that matches (default)

EQ.1: Output to all force transducers that match.

SHLTRW

Optional shell thickness scale factor for contact with rigid walls. Shell thickness is not considered when SHLTRW = 0 (default). SHLTRW = 0.5 will result in an offset of half of shell thickness in contact with rigid walls.

IGACTC

Options to use isogeometric shells for contact detection when contact involves isogeometric shells:

EQ.0: contact between interpolated nodes and interpolated shells

EQ.1: contact between interpolated nodes and isogeometric shells

Remarks:

1. **Shell Thickness.** The shell thickness change option (ISTUPD) must be activated in *CONTROL_SHELL and a nonzero flag specified for SHLTHK above before the shell thickness changes can be included in the surface-to-surface contact types. If thickness changes are to be included in the single surface contact algorithms, an additional flag must be set, see THKCHG above. Although the contact algorithms that include the shell thickness are relatively recent, they work in parallel (MPI) Dyna are fully optimized. The searching in these algorithms is considerably more extensive and therefore slightly more expensive.

2. **Upper Limit on Thickness.** In the single surface contacts types SINGLE_SURFACE, AUTOMATIC_SINGLE_SURFACE, AUTOMATIC_GENERAL, AUTOMATIC_GENERAL_INTERIOR and ERODING_SINGLE_SURFACE, the default contact thickness is taken as the smaller of two values — the shell thickness or 40% of the minimum edge length. (NOTE: Minimum edge length is calculated as min(N4-to-N1, N1-to-N2, N2-to-N3). N3-to-N4 is neglected owing to the possibility of the shell being triangular.) This may lead to unexpected results if it is the intent to include thickness effects when the in-plane shell element dimensions are less than the thickness. The default is based on years of experience where it has been observed that sometimes rather large nonphysical thicknesses are specified to achieve high stiffness values. Since the global searching algorithm includes the effects of shell thicknesses, nonphysical thickness dimensions slow the search down considerably.

3. **Initial Penetration Check.** As of version 950 the initial penetration check option is always performed regardless of the value of ISLCHK. If you do not want to remove initial penetrations then set the contact birth time (see *CONTACT_…) so that the contact is not active at time 0.

4. **Automatic Reorientation.** Automatic reorientation requires offsets between the master and slave surface segments. The reorientation is based on segment connectivity and, once all segments are oriented consistently based on connectivity, a check is made to see if the master and slave surfaces face each other based on the right hand rule. If not, all segments in a given surface are reoriented. This procedure works well for non-disjoint surfaces. If the surfaces are disjoint, the AUTOMATIC contact options, which do not require orientation, are recommended. In the FORMING contact options automatic reorientation works for disjoint surfaces.

5. **Neighbor Segment Thinning Option.** If SPOTHIN is greater than zero and SWRADF is greater than zero, a neighbor segment thinning option is active. The radius of a beam spot weld is scaled by SWRADF, and then a search is

made for shell segments that are neighbors of the tied shell segments that are touched by the weld but not tied by it.

6. **Segment Masses for Penalty Stiffness.** Segment based contact (see SOFT = 2 on *CONTACT) calculates a penalty stiffness based on the solution time step and the masses of the segments in contact. By default, segment masses are calculated using the material density of the element associated with the segment and the volume of the segment. This method does not take into account added mass introduced by lumped masses or mass scaling and can lead to stiffness that is too low. Therefore, a second method (PSTIFF = 1) was added which estimates the segment mass using the nodal masses. Setting a PSTIFF values here will set the default values for all interfaces. The PSTIFF option can also be specified for individual contact interfaces by defining PSTIFF on optional card F of *CONTACT.

7. **Force Transducer Search Option.** Two surface force transducers measure the contact force from any contact interfaces that generate force between the slave and master surfaces of the force transducer. When contact is detected, a search is made to see if the contact force should be added to any 2 surface force transducers. By default, when a force transducer match is found, the force is added and the search terminates. When FTALL = 1, the search continues to check for other two surface force transducer matches. This option is useful when the slave and master force transducer surfaces overlap. If there is no overlap, the default is recommended.

*CONTROL_COUPLING

Purpose: Change defaults for MADYMO3D/CAL3D coupling, see Appendix I.

Card 1	1	2	3	4	5	6	7	8
Variable	UNLENG	UNTIME	UNFORC	TIMIDL	FLIPX	FLIPY	FLIPZ	SUBCYL
Type	F	F	F	F	I	I	I	I
Default	1.	1.	1.	0.	0	0	0	1

VARIABLE	DESCRIPTION
UNLENG	Unit conversion factor for length. MADYMO3D/GM-CAL3D lengths are multiplied by UNLENG to obtain LS-DYNA lengths.
UNTIME	Unit conversion factor for time, UNTIME. MADYMO3D/GM-CAL3D time is multiplied by UTIME to obtain LS-DYNA time.
UNFORC	Unit conversion factor for force, UNFORC. MADYMO3D/GM-CAL3D force is multiplied by UNFORC to obtain LS-DYNA force.
TIMIDL	Idle time during which CAL3D or MADYMO is computing and LS-DYNA remains inactive. Important for saving computer time.
FLIPX	Flag for flipping X-coordinate of CAL3D/MADYMO3D relative to the LS-DYNA model: EQ.0: off, EQ.1: on.
FLIPY	Flag for flipping Y-coordinate of CAL3D/MADYMO3D relative to the LS-DYNA model: EQ.0: off, EQ.1: on.
FLIPZ	Flag for flipping Z-coordinate of CAL3D/MADYMO3D relative to the LS-DYNA model: EQ.0: off, EQ.1: on.

VARIABLE	DESCRIPTION
SUBCYL	CAL3D/MADYMO3D subcycling interval (# of cycles): EQ.0: Set to 1, GT.0: SUBCYL must be an integer equal to the number of LS-DYNA time steps between each CAL3D/MADYMO3D step. Then the position of the contacting rigid bodies is assumed to be constant for n LS-DYNA time steps. This may result in some increase in the spikes in contact, thus this option should be used carefully. As the CAL3D/MADYMO3D programs usually work with a very small number of degrees of freedom, not much gain in efficiency can be achieved.

*CONTROL_CPM

Purpose: Global control parameters for CPM (Corpuscular Particle Method).

Card 1	1	2	3	4	5	6	7	8
Variable	CPMOUT	NP2P	NCPMTS	CPMERR	SFFDC			
Type	I	I	I	I	F			
Default	11	5	0	0	1.0			

VARIABLE	DESCRIPTION

CPMOUT
Control CPM output database to the d3plot files: (see Remark 1)

EQ.11: full CPM database in version 3 format (default)

EQ.21: full CPM database in version 4 format

EQ.22: CPM coordinates only in version 4 format

EQ.23: CPM summary only in version 4 format

NP2P
Number of cycles for repartition particle among processors. This option is only used in LS-DYNA/MPP. (Default = 5)

NCPMTS
Time step size estimation:

EQ.0: not consider CPM (default)

EQ.1: use 1 micro-second as CPM time step size. This provides a better time step size if the model is made by rigid body.

CPMERR
EQ.0: disable checking and only output warning messages (Default)

EQ.1: enable error checking. If LS-DYNA detects any problem, it will either error terminate the job or try to fix the problem. Activated checks include:

1. *Airbag* integrity (see Remark 2)

2. *Chamber* integrity: this step applies the airbag integrity check to the chamber.

3. Inconsistent orientation between the shell refer-

VARIABLE	DESCRIPTION
	ence geometry and FEM shell connectivity.
SFFDC	Scale factor for the force decay constant. The default value is 1 and allowable arrange is [0.01,100]. (See Remark 3)

Remarks:

1. **D3PLOT Version.** "Version 3" is an older format than "Version 4". Version 4 stores data more efficiently than version 3 and has options for what data is stored, but may not be readable by old LS-PrePost executables.

2. **Airbag Integrity Checking.** The bag's volume is used to evaluate all bag state variables. If the volume is ill-defined or inaccurate, then the calculation will fail. Therefore, it is vital that that the volume be closed, and that all shell normal vectors point in the same direction.

   When CPMERR = 1 the calculation will error terminate if either the bag's volume is not closed or if one of its parts is not internally oriented (meaning that it contains elements that are not consistently oriented). Once it is verified that each part has a well-defined orientation, an additional check is performed to verify that all of bag's constituent parts are consistently oriented with respect to each other. If they are not, then the part orientations are flipped until the bag is consistently oriented with an *inward* pointing normal vector.

3. **Force Decay Constant.** Particle impact force is gradually applied to airbag segment by a special smoothing function with the following form.

$$F_{\text{apply}} = \left[1 - \exp\left(\frac{-dt}{SFFDC \times \tau}\right)\right](F_{\text{current}} + F_{\text{stored}})$$

   Where τ is the force decay constant stored in LS-DYNA.

*CONTROL_CPU

Purpose: Control CPU time.

Card 1	1	2	3	4	5	6	7	8
Variable	CPUTIM	IGLST						
Type	F	I						

VARIABLE **DESCRIPTION**

CPUTIM Seconds of CPU time:

EQ.0.0: no CPU time limit set

GT.0.0: time limit for cumulative CPU of the entire simulation, including all restarts.

LT.0.0: absolute value is the CPU time limit in seconds for the first run and for each subsequent restart.

IGLST Flag for outputting CPU and elapsed times in the "glstat" file

EQ.0: no

EQ.1: yes

Remarks:

The CPU limit is not checked until after the initialization stage of the calculation. Upon reaching the CPU limit, the code will output a restart dump file and terminate. The CPU limit can also be specified on the LS-DYNA execution line via "c=". If a value is specified on both the execution line and in the input deck, the minimum value will be used.

*CONTROL_DEBUG

Purpose: Write supplemental information to the messag file(s). One effect of this command is that the sequence of subroutines called during initialization and memory allocation is printed. Aside from that, the extra information printed pertains only to a select few features, including:

1. Spot weld connections which use *MAT_100_DA and *DEFINE_CONNECTION_PROPERTIES.

2. The GISSMO damage model invoked using *MAT_ADD_EROSION. (Supplemental information about failed elements is written.)

*CONTROL_DISCRETE_ELEMENT

Purpose: Define global control parameters for discrete element spheres.

Card 1	1	2	3	4	5	6	7	8
Variable	NDAMP	TDAMP	FRICS	FRICR	NORMK	SHEARK	CAP	VTK
Type	F	F	F	F	F	F	I	I
Default	0	0	0	0	0.01	2/7	0	0

Capillary Card. Additional card for CAP ≠ 0.

Card 2	1	2	3	4	5	6	7	8
Variable	GAMMA	VOL	ANG	GAP			NBUF	PARALLEL
Type	F	F	F	F			I	I
Default	0	0	0	0			6	0

Card 3 is optional. If optional Card 3 is used, then Optional Card 2 must be defined.

Card 3	1	2	3	4	5	6	7	8
Variable	LNORM	LSHEAR		FRICD	DC			
Type	I	I		F	F			
Default	0	0		FRICS	0			

VARIABLE	DESCRIPTION
NDAMP	Normal damping coefficient
TDAMP	Tangential damping coefficient

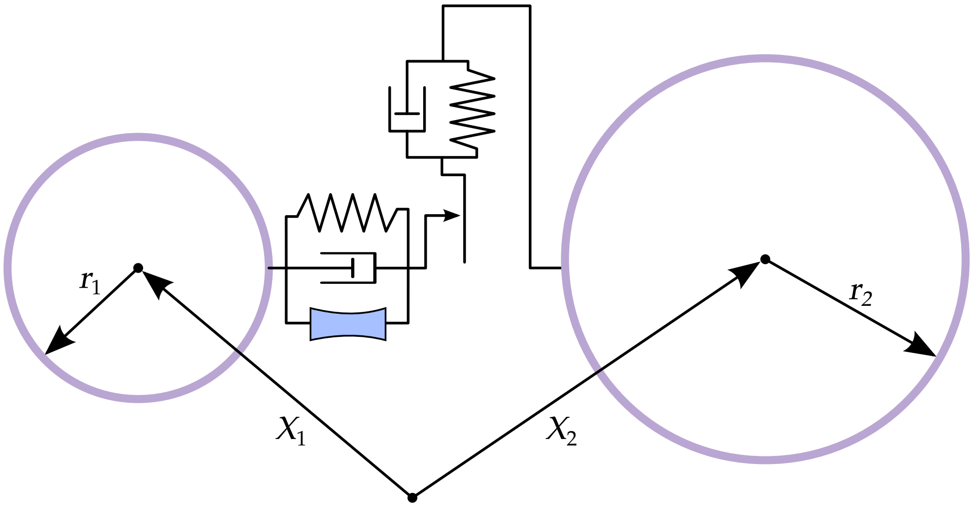


Figure 12-12. Schematic representation of sphere-sphere interaction

VARIABLE **DESCRIPTION**

FRICS

Static coefficient of friction

EQ.0: 3 DOF

NE.0: 6 DOF (consider rotational DOF)

FRICR

Rolling friction coefficient

NORMK

Optional: scale factor of normal spring constant. Norm contact stiffness is calculated as $K_n = \begin{cases} \frac{k_1 r_1 k_2 r_2}{k_1 r_1 + k_2 r_2} NORMK & if\ NORMK > 0 \\ |NORMK| & if\ NORMK < 0 \end{cases}$ (Default = 0.01)

SHEARK

Optional: ratio between SHEARK/NORMK (Default = 2/7). Tangential stiffness is calculated as $K_t = SHEARK \bullet K_n$

CAP

EQ.0: dry particles

NE.0: wet particles, consider capillary force and need additional input card. See Remark 1.

VTK

Output DES in VTK format for ParaView

EQ.0: no

EQ.1: yes

GAMMA

Liquid surface tension, γ

VOL

Volume fraction

VARIABLE	DESCRIPTION
ANG	Contact angle, θ
GAP	Optional parameter affecting the spatial limit of the liquid bridge. CAP.EQ.0: GAP is ignored, if the CAP field is 0 and the simulation is modeling dry particles. CAP.NE.0: A liquid bridge exists when δ, as illustrated in Figure 12-13, is less or equal to $\min(\text{GAP}, d_{\text{rup}})$ where d_{rup} is the rupture distance of the bridge automatically calculated by LS-DYNA (see CAP field).
NBUF	GE.0: Factor of memory use for asynchronous message buffer (Default = 6) LT.0: Disable asynchronous scheme and use minimum memory for data transfer
PARALLEL	EQ.0: skip contact force calculation for bonded DES (Default) EQ.1: consider contact force calculation for bonded DES
LNORM	Load curve ID of a curve that defines function for normal stiffness with respect to norm penetration ratio. See Remark 2.
LSHEAR	Load curve ID of a curve that defines function for shear stiffness with respect to norm penetration ratio. See Remark 3.
FRICD	Dynamic coefficient of friction. By default, FRICD = FRICS. The frictional coefficient is assumed to be dependent on the relative velocity $v_{rel}$of the two DEM in contact $\mu_c = FRICD + (FRICS - FRICD)e^{-DC\bullet\lvert v_{rel}\rvert}$.
DC	Exponential decay coefficient. The frictional coefficient is assumed to be dependent on the relative velocity $v_{rel}$of the two DEM in contact $\mu_c = FRICD + (FRICS - FRICD)e^{-DC\bullet\lvert v_{rel}\rvert}$.

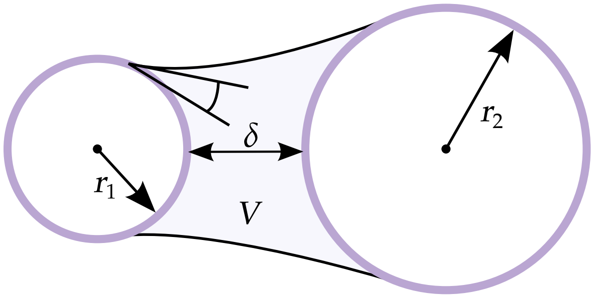


Figure 12-13. Schematic representation of capillary force model.

Background:

This method models all parts as being comprised of rigid spheres. These sphere interact with both conventional solids and other spheres. Sphere-sphere interactions are modeled in contact points using springs and dampers as illustrated in Figure 12-12. [Cundall & Strack 1979]

Remarks:

1. **Capillary Forces to Model Cohesion.** This extension is enabled using the CAP field. Capillary force between wet particles is based on the following reference. "Capillary Forces between Two Spheres with a Fixed Volume Liquid Bridges: Theory and Experiment", Yakov I. Rabinovich et al. Langmuir 2005, 21, 10992-10997. See Figure 12-13.

   The capillary force is given by

$$F = -\frac{2\pi R\gamma cos\theta}{1 + \frac{\delta}{2d}},$$

   where,

$$d = \frac{\delta}{2}\left(-1 + \sqrt{1 + \frac{2V}{\pi R\delta^2}}\right),$$

   and,

$$R = \frac{2r_1r_2}{r_1 + r_2}.$$

2. For two interacting DEMs with user defined curve for norm stiffness $y = f(\mathrm{x})$, where $x = \frac{\delta}{\min(r_1,r_2)}$ is relative penetration, and δ is penetration; the normal spring force is calculated as

$$F_n = k_{eff} \bullet y \bullet min^2(r_1, r_2)$$

where k_{eff} is the effective bulk modulus of two interacting DEM particles $k_{eff} = \frac{k_1 k_2}{k_1 + k_2}$. If curve is defined as $y = c \bullet x$, the behavior is the same as $NORMK = c$.

3. For two interacting DEMs with user defined curve for shear stiffness $y = f(\mathrm{x})$, where $x = \frac{\delta}{\min(r_{1,}r_2)}$ is relative penetration, and δ is penetration; the tangential stiffness is calculated as $K_s = y \bullet K_n$, where K_n is norm stiffness defined by NORMK or user defined curve. If curved is defined as $y = c$, the behavior is the same *as SHEARK* $= c$.

*CONTROL_DYNAMIC_RELAXATION

Purpose: Initialize stresses and deformation in a model to simulate a preload. Examples of preload include load due to gravity, load due to a constant angular velocity, and load due to torquing of a bolt. After the preloaded state is achieved by one of the methods described below, the time resets to zero and the normal phase of the solution automatically begins from the preloaded state.

IDRFLG controls the manner in which the preloaded state is computed. If IDRFLG is 1 or -1, a transient "dynamic relaxation" analysis is invoked in which an explicit analysis, damped by means of scaling nodal velocities by the factor DRFCTR each time step, is performed. When the ratio of current distortional kinetic energy to peak distortional kinetic energy (the convergence factor) falls below the convergence tolerance (DRTOL) or when the time reaches DRTERM, the dynamic relaxation analysis stops and the current state becomes the initial state of the subsequent normal analysis.

Distortional kinetic energy is defined as total kinetic energy less the kinetic energy due to rigid body motion. A history of the distortional kinetic energy computed during the dynamic relaxation phase is automatically written to a file called "relax". This file can be read as an ASCII file by LS-PrePost and its data plotted. The "relax" file also includes a history of the convergence factor.

To create a binary output database having the same format as a d3plot database but which pertains to the dynamic relaxation analysis, use *DATABASE_BINARY_D3-DRLF. The output interval is given by this command as an integer representing the number of convergence checks between output states. The frequency of the convergence checks is controlled by the parameter NRCYCK.

Dynamic relaxation will be invoked if SIDR is set to 1 or 2 in any of the *DEFINE_-CURVE commands, even if IDRFLG = 0 in *CONTROL_DYNAMIC_RELAXATION. Curves so tagged are applicable to the dynamic relaxation analysis phase. Curves with SIDR set to 0 or 2 are applicable to the normal phase of the solution. Dynamic relaxation will always be skipped if IDRFLAG is set to -999.

At the conclusion of the dynamic relaxation phase and before the start of the normal solution phase, a binary dump file (d3dump01) and a "prescribed geometry" file (drdisp.sif) are written by LS-DYNA. Either of these files can be used in a subsequent analysis to quickly initialize to the preloaded state without having to repeat the dynamic relaxation run. The binary dump file is utilized via a restart analysis (see the *RESTART section of the LS-DYNA User's Manual). The drdisp.sif file is utilized by setting IDRFLG=2 as described below and discussed in Remark 1.

If IDRFLG is set to 2, the preloaded state is quickly reached by linearly ramping nodal displacements, rotations, and temperatures to prescribed values over 100 time steps, or over a number of time steps as indicated by the variable NC. See the optional cards pertaining to IDRFLG = 2 and also Remarks 1 and 5.

If IDRFLG is set to 5, an implicit analysis is performed to obtain the preloaded state and in this case, the preload analysis completes when 'time' is equal to DRTERM. The implicit step size is specified with a *CONTROL_IMPLICIT_GENERAL command. The implicit analysis is, by default, static but can be made transient via the *CONTROL_IMPLICIT_DYNAMICS command (see Remark 3).

IDRFLG = 6 also performs an implicit analysis as with IDRFLG = 5 but only for the part subset specified with DRPSET.

Card 1	1	2	3	4	5	6	7	8
Variable	NRCYCK	DRTOL	DRFCTR	DRTERM	TSSFDR	IRELAL	EDTTL	IDRFLG
Type	I	F	F	F	F	I	F	I
Default	250	0.001	0.995	infinity	TSSFAC	0	0.04	0
Remarks				3				1, 2, 3

Additional card for IDRFLG = 3 or 6.

Card 2	1	2	3	4	5	6	7	8
Variable	DRPSET							
Type	I							
Default	0							
Remarks	4							

Additional card for IDRFLG = 2.

Card 2	1	2	3	4	5	6	7	8
Variable	NC	NP						
Type	I	I						
Default	100	0						

NP Additional cards for IDRFLG = 2.

Card 3	1	2	3	4	5	6	7	8
Variable	PSID	VECID						
Type	I	I						
Default	0	0						

VARIABLE	DESCRIPTION
NRCYCK	Number of time steps between convergence checks for explicit dynamic relaxation.
DRTOL	Convergence tolerance for explicit dynamic relaxation (default = 0.001).
DRFCTR	Dynamic relaxation factor (default = .995).
DRTERM	Optional termination time for dynamic relaxation. Termination occurs at this time or when convergence is attained (default = infinity).
TSSFDR	Scale factor for computed time step during explicit dynamic relaxation. If zero, the value is set to TSSFAC defined on *CONTROL_TIMESTEP. After converging, the scale factor is reset to TSSFAC.

VARIABLE **DESCRIPTION**

IRELAL

Automatic control for dynamic relaxation option based on algorithm of Papadrakakis [1981]:

EQ.0: not active,

EQ.1: active.

EDTTL

Convergence tolerance on automatic control of dynamic relaxation.

IDRFLG

Dynamic relaxation flag for stress initialization:

EQ.-999: dynamic relaxation not activated even if specified on a load curve, see *DEFINE_CURVE.

EQ.-1: dynamic relaxation is activated and time history output is produced during dynamic relaxation, see Remark 2.

EQ.0: not active,

EQ.1: dynamic relaxation is activated,

EQ.2: initialization to a prescribed geometry, see Remark 1,

EQ.3: dynamic relaxation is activated as with IDRFLG = 1, but with a part set ID for convergence checking,

EQ.5: initialize implicitly, see Remark 3.

EQ.6 initialize implicity but only for the part set specified by DRPSET.

DRPSET

Part set ID for convergence checking (for IDRFLG = 3 or 6 only)

NC

Number of time steps for initializing geometry of IDRFLG = 2.

NP

Number of part sets specified for IDRFLG = 2.

PSID

Part set ID for IDRFLG = 2.

VECID

Vector ID for defining origin and axis of rotation for IDRFLG = 2. See Remark 5.

Remarks:

1. When IDRFLG = 2, an ASCII file specified by "m=" on the LS-DYNA execution line is read which describes the initialized state. The ASCII file contains each

node ID with prescribed values of nodal displacement (x, y, z), nodal rotation (x, y, z) and nodal temperature in (I8, 7E15.0) format.

2. If IDRFLG is set to -1 the dynamic relaxation proceeds as normal but time history data is written to the d3thdt file in addition to the normal data being written to the d3drlf file. At the end of dynamic relaxation, the problem time is reset to zero. However, information is written to the d3thdt file with an increment to the time value. The time increment used is reported at the end of dynamic relaxation.

3. When IDRFLG = 5 or 6, LS-DYNA performs an implicit analysis for the preload phase of the simulation. Parameters for controlling the implicit preload solution are defined using appropriate *CONTROL_IMPLICIT keywords to specify solver type, implicit time step, etc. When using this option, one must specify DRTERM to indicate the termination "time" of the implicit preload analysis. When DRTERM is reached, the implicit preload phase terminates and LS-DYNA begins the next phase of the analysis according to IMFLAG in *CONTROL_IMPLICIT_GENERAL. For example, if it is desired to run an implicit preload phase and switch to the explicit solver for the subsequent transient phase, IDRFLG should be set to 5 and IMFLAG should be set to 0.

4. When IDRFLG = 3, a part set ID is used to check for convergence. For example, if only the tires are being inflated on a vehicle, it may be sufficient in some cases to look at convergence based on the part ID’s in the tire and possibly the suspension system. You can also use IDRFLG = 6 to perform the initialization using implicit on the part set.

5. When the displacements for IDRFLG = 2 are associated with large rotations, the linear interpolation of the displacement field introduces spurious compression and tension into the part. If a part set is specified with a vector, the displacement is interpolated by using polar coordinates with the tail of the vector specifying the origin of the coordinate system and the direction specifying the normal to the polar coordinate plane.

*CONTROL_EFG

Purpose: Define controls for the mesh-free computation.

Card 1	1	2	3	4	5	6	7	8
Variable	ISPLINE	IDILA	ININT					
Type	I	I	I					
Default	0	0	12					
Remarks			1					

Card 2	1	2	3	4	5	6	7	8
Variable	IMLM	ETOL	IDEB	HSORT	SSORT			
Type	I	F	I	I	I			
Default	0	1.0E-4	0	0	0			

VARIABLE	DESCRIPTION
ISPLINE	Optional choice for the mesh-free kernal functions: EQ.0: Cubic spline function (default) EQ.1: Quadratic spline function EQ.2: Cubic spline function with circular disk
IDILA	Optional choice for the normalized dilation parameter: EQ.0: Maximum distance based on the background element EQ.1: Maximum distance based on surrounding nodes
ININT	This is the factor needed for the estimation of maximum workspace (MWSPAC) that can be used during the initialization phase.

VARIABLE	DESCRIPTION
IMLM	Optional choice for the matrix operation, linear solving and memory usage: EQ.1: Original BCSLIB-EXT solvers EQ.2: EFGPACK
ETOL	Error tolerance in the IMLM. When IMLM = 2 is used, ININT in card one becomes redundant. IMLM = 2 is recommended.
IDEB	Output internal debug message
HSORT	Not used
SSORT	Automatic sorting of background triangular shell elements to FEM #2 when EFG shell type 41 is used EQ.0: no sorting EQ.1: full sorting

Remarks:

1. The mesh-free computation requires calls to use BCSLIB-EXT solvers during the initialization phase. The maximum workspace (MWSPAC) that can be used during the call is calculated as

$$\text{MWSPAC} = \text{ININT}^3 \times \text{NUMNEFG},$$

where NUMNEFG is the total number of mesh-free nodes. ININT, which is the number of nodes that a node influences along each cardinal direction, defaults to 12. When the normalized dilation parameters (DX, DY, DZ) in *SECTION_SOLID_EFG are increased ININT must likewise increase.

2. When ISPLINE = 2 is used, the input of the normalized dilation parameters (DX, DY, DZ) for the kernel function in *SECTION_SOILD_EFG and SECTIOL_SHELL_EFG only requires the DX value.

3. EFGPACK was added to automatically compute the required maximum workspace in the initialization phase and to improve efficiency in the matrix operations, linear solving, and memory usage. The original BCSLIB-EXT solver requires an explicit workspace (ININT) for the initialization.

*CONTROL_ENERGY

Purpose: Provide controls for energy dissipation options.

Card 1	1	2	3	4	5	6	7	8
Variable	HGEN	RWEN	SLNTEN	RYLEN				
Type	I	I	I	I				
Default	1	2	1	1				

VARIABLE **DESCRIPTION**

HGEN

Hourglass energy calculation option. This option requires significant additional storage and increases cost by ten percent:

EQ.1: hourglass energy is not computed (default),

EQ.2: hourglass energy is computed and included in the energy balance. The hourglass energies are reported in the ASCII files glstat and matsum, see *DATABASE_*OPTION*.

RWEN

Rigidwall energy (a.k.a. stonewall energy) dissipation option:

EQ.1: energy dissipation is not computed,

EQ.2: energy dissipation is computed and included in the energy balance (default). The rigidwall energy dissipation is reported in the ASCII file glstat, see *DATABASE_*OPTION*.

SLNTEN

Sliding interface energy dissipation option (This parameter is always set to 2 if contact is active. The option SLNTEN = 1 is not available.):

EQ.1: energy dissipation is not computed,

EQ.2: energy dissipation is computed and included in the energy balance. The sliding interface energy is reported in ASCII files glstat and sleout, see *DATABASE_*OPTION*.

VARIABLE	DESCRIPTION
RYLEN	Rayleigh energy dissipation option (damping energy dissipation): EQ.1: energy dissipation is not computed (default), EQ.2: energy dissipation is computed and included in the energy balance. The damping energy is reported in ASCII file glstat and matsum, see *DATABASE_*OPTION*.

*CONTROL_EXPLICIT_THERMAL

The *CONTROL_EXPLICIT_THERMAL_SOLVER keyword activates an explicit finite volume code solving heat transfers by conduction. Enthalpies and temperatures are element centered. The elements supported by the thermal solver are beams, shells, solids, multi-material 3D ALE elements. The *CONTROL_EXPLICIT_THERMAL_PROPERTIES keyword defines the heat capacities and conductivities by parts. These 2 keywords are mandatory to properly run the solver. Other keywords can be used to set the initial and boundary conditions and control the outputs. They are all listed below in alphabetical order:

*CONTROL_EXPLICIT_THERMAL_ALE_COUPLING

*CONTROL_EXPLICIT_THERMAL_BOUNDARY

*CONTROL_EXPLICIT_THERMAL_CONTACT

*CONTROL_EXPLICIT_THERMAL_INITIAL

*CONTROL_EXPLICIT_THERMAL_OUTPUT

*CONTROL_EXPLICIT_THERMAL_PROPERTIES

*CONTROL_EXPLICIT_THERMAL_SOLVER

*CONTROL_EXPLICIT_THERMAL_ALE_COUPLING

Purpose: Define the shell and solid parts involved in an explicit finite volume thermal coupling with multi-material ALE groups. This keyword requires *CONSTRAINED_LAGRANGE_IN_SOLID, CTYPE = 4.

Card 1	1	2	3	4	5	6	7	8
Variable	PARTSET	MMGSET						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
PARTSET	Part set ID (See *SET_PART)
MMGSET	Multi-material set ID (see *SET_MULTI-MATERIAL_GROUP_LIST)

Remarks:

*CONTROL_EXPLICIT_THERMAL_BOUNDARY

Purpose: Set temperature boundaries with segment sets for an explicit finite volume thermal analysis.

Card 1	1	2	3	4	5	6	7	8
Variable	SEGSET	LCID						
Type	I	F						
Default	none	none						

VARIABLE	DESCRIPTION
SEGSET	Segment set ID (See *SET_SEGMENT)
LCID	*DEFINE_CURVE ID defining the temperature in function of time

Remarks:

1. **Boundary elements.** The boundary temperatures are set at segment centers. If shells or beams have all their nodes in the segment set, these elements would be considered as boundary elements: the temperatures at their centers will be controlled by the curve LCID.

*CONTROL_EXPLICIT_THERMAL_CONTACT

Purpose: Define the beam, shell and solid parts involved in an explicit finite volume thermal contact.

Card 1	1	2	3	4	5	6	7	8
Variable	PARTSET	NCYCLE						
Type	I	F						
Default	none	1						

VARIABLE	DESCRIPTION
PARTSET	Part set ID (See *SET_PART)
NCYCLE	Number of cycle between checks of new contact

Remarks:

*CONTROL_EXPLICIT_THERMAL_INITIAL

Purpose: Initialize the temperature centered in beams, shells or solids involved in an explicit finite volume thermal analysis.

Card 1	1	2	3	4	5	6	7	8
Variable	SET	SETYP	TEMPINI					
Type	I	F	F					
Default	none	none	0.0					

VARIABLE	DESCRIPTION
SET	set ID
SETYP	Type of set: EQ.1: solid set EQ.2: shell set EQ.3:
TEMPINI	Initial temperature (see Remark 1)

Remarks:

1. **Material with *EOS.** The volumetric enthalpy is the sum of the pressure and volumetric internal energy (as defined in *EOS). If the material has an equation of state, the enthalpy should not be initialized by the temperature but by the initial volumetric internal energy and pressure set in *EOS.

*CONTROL_EXPLICIT_THERMAL_OUTPUT

Purpose: Output temperatures and enthalpies (see Remark 1) for an explicit finite volume thermal analysis.

Card 1	1	2	3	4	5	6	7	8
Variable	DTOUT	DTOUTYP	SET	SETYP				
Type	F	I	I	I				
Default	none	0	0	0				

VARIABLE	DESCRIPTION
DTOUT	Time interval between outputs
DTOUTYP	Type of DTOUT: EQ.0: DTOUT is a constant EQ.1: DTOUT is the ID of *DEFINE_CURVE defining a table of time vs DTOUT
SET	set ID (see Remark 2)
SETYP	Type of set: EQ.1: solid set (see *SET_SOLID) EQ.2: shell set (see *SET_SHELL) EQ.3: beam set (see *SET_BEAM

Remarks:

1. **Output centered element data.** The temperatures and enthalpies are at the element center. They are output in d3plot format files with the following base-name: xplcth_output. The temperatures and enthalpies are respectively fringed by FriComp > X-stress and Y-stress (select Mid for the integration points in shells).

2. **Output by element.** If a set of elements SET is defined, the temperature and enthalpy histories are output by element in a .xy format. The file names are

temperature_{beam,shell,solid}ID.xy and enthalpy_{beam,shell,solid}ID.xy. The binary file xplcth_output is not output.

*CONTROL_EXPLICIT_THERMAL_PROPERTIES

Purpose: Define the thermal properties of beam, shell and solid parts involved in an explicit finite volume thermal analysis.

Card 1	1	2	3	4	5	6	7	8
Variable	PARTSET	CP	CPTYP	VECID1	VECID2	LOCAL		
Type	I	F	I	I	I	I		
Default	none	none	0	0	0	0		

Card 2	1	2	3	4	5	6	7	8
Variable	Kxx	Kxy	Kxz	KxxTYP	KxyTYP	KxzTYP		
Type	F	F	F	I	I	I		
Default	0.0	0.0	0.0	0	0	0		

Card 3	1	2	3	4	5	6	7	8
Variable	Kyx	Kyy	Kyz	KyxTYP	KyyTYP	KyzTYP		
Type	F	F	F	I	I	I		
Default	0.0	0.0	0.0	0	0	0		

Card 4	1	2	3	4	5	6	7	8
Variable	Kzx	Kzy	Kzz	KzxTYP	KzyTYP	KzzTYP		
Type	F	F	F	I	I	I		
Default	0.0	0.0	0.0	0	0	0		

VARIABLE	DESCRIPTION
PARTSET	Part set ID (See *SET_PART)
CP	Heat capacity
CPTYP	Type of CP: EQ.0: CP is a constant EQ.1: CP is the ID of *DEFINE_CURVE defining a table of temperature vs heat capacity
VECID1, VECID2	*DEFINE_VECTOR IDs to define a specific coordinate system. VECID1 and VECID2 give the *x*- and *y*-direction respectively. The *z*-vector is a cross product of VECID1 and VECID2. If this latter is not orthogonal to VECID1, its direction will be corrected with a cross-product of *z*- and *x*-vectors. The conductivity matrix Kij is applied this coordinate system.
LOCAL	Flag to activate an element coordinate system: EQ.0: The vectors VECIDj are considered in a global coordinate system. EQ.1: The vectors VECIDj are considered in a local system attached to the element. For shells and solids, the system is the same as DIREC = 1 and CTYPE = 12 in *CONSTRAINED_LAGRANGE_IN_SOLID. For shells, the edge centers replace the face centers. For beams, the *x*-direction is aligned with the first 2 nodes in *ELEMENT_BEAM and there should be a 3rd node for the *y*-direction.
Kij	Heat conductivity matrix

VARIABLE **DESCRIPTION**

KijTYP Type of Kij:

EQ.0: Kij is a constant

EQ.1: Kij is the ID of *DEFINE_CURVE defining a table of temperature vs heat conductivity

Remarks:

*CONTROL_EXPLICIT_THERMAL_SOLVER

Purpose: Define the beam, shell and solid parts involved in a finite volume thermal analysis. The enthalpies and temperatures are explicitly updated in time.

Card 1	1	2	3	4	5	6	7	8
Variable	PARTSET	DTFAC						
Type	I	F						
Default	none	1.0						

VARIABLE	DESCRIPTION
PARTSET	Part set ID (See *SET_PART)
DTFAC	Time step factor (see Remark 1)

Remarks:

1. **Time step.** The time step is a minimum of the mechanical and thermal time steps. The thermal time step is a minimum of the element thermal time steps, which are half the enthalpies divided by the right hand side of the heat equation (conductivity * temperature laplacian). The thermal time step is scaled by DTFAC (=1 by default)

CONTROL_EXPLOSIVE_SHADOW_*{OPTION}

Available option includes:

<BLANK>

SET

Purpose: Compute detonation times in explosive elements for which there is no direct line of sight. If this command is not included in the input, the lighting time for an explosive element is computed using the distance from the center of the element to the nearest detonation point, L_d; the detonation velocity, D; and the lighting time for the detonator, t_d:

$$t_L = t_d + \frac{L_d}{D}$$

The detonation velocity for this option is taken from the element whose lighting time is computed and does not account for the possibilities that the detonation wave may travel through other explosives with different detonation velocities or that the line of sight may pass outside of the explosive material.

If this command is present, the lighting time of each explosive element is based on the shortest path through the explosive material from the associated detonation point(s) to the explosive element. If inert obstacles exist within the explosive material, the lighting time will account for the extra time required for the detonation wave to travel around the obstacles. The lighting times also automatically accounts for variations in the detonation velocity if different explosives are used.

The SET option requires input of a set ID of two-dimensional shell elements or three-dimensional solid elements for which explosive shadowing is active. If the SET option is not used, Card 1 should be omitted and shadowing is active for all explosive elements.

See also *INITIAL_DETONATION and *MAT_HIGH_EXPLOSIVE.

Card 1. Card for SET keyword option.

Card 1	1	2	3	4	5	6	7	8
Variable	SETID							
Type	I							
Default	None							

VARIABLE	DESCRIPTION
SETID	Set ID of a *SET_SHELL or *SET_SOLID. If the SET option is active, the lighting times are computed for a set of shells (*SET_SHELL in two dimensions) or solids (*SET_SOLID in three dimensions).

*CONTROL_FORMING

Purpose: Set parameters for metal forming related features.

*CONTROL_FORMING_AUTOCHECK

*CONTROL_FORMING_AUTO_NET

*CONTROL_FORMING_AUTOPOSITION

*CONTROL_FORMING_BESTFIT

*CONTROL_FORMING_INITIAL_THICKNESS

*CONTROL_FORMING_MAXID

*CONTROL_FORMING_ONESTEP

*CONTROL_FORMING_OUTPUT

*CONTROL_FORMING_PARAMETER_READ

*CONTROL_FORMING_POSITION

*CONTROL_FORMING_PRE_BENDING

*CONTROL_FORMING_PROJECTION

*CONTROL_FORMING_REMOVE_ADAPTIVE_CONSTRAINTS

*CONTROL_FORMING_SCRAP_FALL

*CONTROL_FORMING_SHELL_TO_TSHELL

*CONTROL_FORMING_STONING

*CONTROL_FORMING_TEMPLATE

*CONTROL_FORMING_TIPPING

*CONTROL_FORMING_TOLERANC

*CONTROL_FORMING_TRAVEL

*CONTROL_FORMING_TRIM_MERGE

*CONTROL_FORMING_TRIMMING

*CONTROL_FORMING_UNFLANGING

*CONTROL_FORMING_USER

*CONTROL_FORMING_AUTOCHECK

Purpose: This keyword detects and corrects flaws in the mesh for the *rigid body* that models the tooling. Among its diagnostics are checks for duplicated elements, overlapping elements, skinny/long elements, degenerated elements, disconnected elements, and inconsistent element normal vectors.

This feature also automatically orients each tool's element normal vectors so that they face the blank. Additionally an offset can be specified to create another tool (tool physical offset) based on the corrected tool meshes. Note that this keyword is distinct from *CONTROL_CHECK_SHELL, which checks and corrects mesh quality problem after trimming, to prepare the trimmed mesh for the next stamping process. This keyword only applies to shell elements.

The tool offset feature is now available in *LS-PrePost 4.2* under *Application* → *MetalForming* → *Easy Setup*.

Card 1	1	2	3	4	5	6	7	8
Variable	ICHECK	IGD	IOFFSET	IOUTPUT				
Type	I	I	I	I				
Default	0	none	0	none				

VARIABLE	DESCRIPTION
ICHECK	Tool mesh checking/correcting flag: ICHECK.EQ.0: Do not activate mesh checking/correcting feature. ICHECK.EQ.1: Activate comprehensive mesh check and correct problematic tool meshes (see Figure 12-15). This option reduces the likelihood of unreasonable forming results and/or error termination. This is only for regular forming simulations. The calculation will continue after the tool mesh checking/correcting phase is completed. See Example 1. The corrected tool meshes can be viewed and recovered from the resulting d3plot files. If the termination time is set to "0.0" or the keyword *CONTROL_TERMINATION is absent all together, the simulation will terminate as soon as checking/correcting is completed, and corrected tool meshes can be extracted from the d3plot files.
IGD	Not used.
IOFFSET	Tool mesh offset flag. This variable works only when IOUTPUT is defined, and ICHECK is set to "1": IOFFSET.EQ.0: Do not offset rigid tool mesh. The sheet blank does not need to be present. In this case the output files rigid_offset.inc and rigid_offset_before.inc will be identical. See Example 2. IOFFSET.EQ.1: Perform rigid tool mesh offset using the variable MST (see Figure 12-14) as specified on a *CONTACT_FORMING_… card. The blank must be defined and positioned completely above or below the rigid tool to be offset. Both part ID and part SID (MSTYP) can be used in defining the MSID. IOUTPUT must also be defined.

VARIABLE **DESCRIPTION**

IOUTPUT Output option flag:

IOUTPUT.EQ.1: Output offset rigid tool meshes into a keyword file rigid_offset.inc, and terminates the simulation.

IOUTPUT.EQ.2: Output offset rigid tool meshes as well as nodes used to define draw beads into a keyword file rigid_offset.inc, and terminates the simulation. See Example 4.

IOUTPUT.EQ.3: Output checked/corrected tool as well as offset rigid tool meshes into two separate keyword files, rigid_offset_before.inc, and rigid_offset.inc, respectively, and terminates the simulation. See Example 3.

IOUTPUT.EQ.4: Output checked/corrected tool meshes, offset rigid tool meshes as well as the nodes used to define draw beads into two separate keyword files, rigid_offset_before.inc, and rigid_offset.inc, respectively, and terminates the simulation.

Remarks:

In sheet metal forming, tools are typically modelled as rigid bodies and their meshes are prepared from CAD (IGES or STEP) files according to the following procedure:

1. The user imports the CAD data into a preprocessor, such as *LS-PrePost*.

2. The preprocessor automatically generates a mesh. *LS-PrePost* features a streamlined GUI for this application.

3. Export the generated mesh to LS-DYNA input files. The *LS-PrePost eZ-Setup* user interface provides quick access to generate the necessary input files for metal forming applications.

Ideally, this process should produce a good mesh requiring no manual intervention. Often, though, such meshes that have been automatically generated from CAD data have flaws severe enough to prevent an accurate or complete calculation. This feature, *CONTROL_FORMING_AUTOCHECK is intended to make LS-DYNA more robust with respect to tooling mesh quality.

This keyword *requires* that the tooling meshes represent rigid bodies. Also, when this keyword is used, a part ID or a part set ID, corresponding to MSTYP = 2 or 3 on the *CONTACT_FORMING_… card, may be used to define the master side, MSID. Segment set ID input, MSTYP = 0, is not supported.

Some cases of incoming bad tooling meshes which can be corrected by this keyword are shown in Figure 12-15. This keyword can be inserted anywhere in the input deck. To include the corrected tooling mesh into the d3plot the ICHECK field must be defined. The corrected mesh is written to rigid_offset_before.inc file if IOFFSET and IOUTPUT are defined.

When IOFFSET = 1 and IOUTPUT is defined, the tool meshes will first be checked, corrected, and reoriented correctly towards the blank. Then the tool is offset by an amount of 0.5|MST| either on the same or opposite side of the blank, depending on the signs of the MST field on the *CONTACT_FORMING_… card (Figure 12-14). A new keyword file, "rigid_offset.inc" file, will be output as containing the corrected, reoriented, and offset tooling mesh.

The tool offset feature is now available in *LS-PrePost 4.2* under *Application* → *MetalForming* → *Easy Setup*. The *offset from Die* button under *Binder* can be used to create offset tools.

Note this keyword does not work with the SMOOTH option in *CONTACT_FORMING_… prior to Revision 95456, see Revision information.

Example 1 - Mesh checking/correction in a regular forming simulation:

The keyword can be inserted anywhere in a regular forming simulation input deck. A partial input example of checking, correcting the tool meshes and reorienting all tools' normals is provided below. Note that although MST is defined between blank and die contact interface, die meshes will not be offset, since IOFFSET is not defined. Simulation will continue if "&endtime" is not zero, but will terminate as soon as the checking and correcting are done if "&endtime" is set to "0.0", or *CONTROL_TERMINATION is absent all together. Corrected and reoriented tool meshes can be viewed and recovered from d3plot files.

```
*KEYWORD
*INCLUDE
Tool_blank.k
*CONTROL_FORMING_AUTOCHECK
$   ICHECK       IGD   IOFFSET   IOUTOUT
         1
*CONTROL_TERMINATION
&endtime
⋮
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ID
         1  blank to punch
         1         2         2         2                             1         1
 0.110E+00 0.000E+00 0.000E+00 0.000E+00 0.200E+02          0.0000E+00 0.100E+21
```

```
     0.000     0.000                 0.0
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ID
         2  blank to die
$     SSID      MSID     SSTYP     MSTYP    SBOXID    MBOXID       SPR       MPR
         1         3         2         2                             1         1
$       FS        FD        DC        VC       VDC    PENCHK        BT        DT
 0.110E+00 0.000E+00 0.000E+00 0.000E+00 0.200E+02           0.000E+00 0.100E+21
$      SFS       SFM       SST       MST      SFST      SFMT       FSF       VSF
     0.000     0.000              -1.600
$     SOFT    SOFSCL    LCIDAB    MAXPAR    PENTOL     DEPTH     BSORT    FRCFRQ
         0
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ID
         3  blank to binder
         1         4         2         2                             1         1
 0.110E+00 0.000E+00 0.000E+00 0.000E+00 0.200E+02           0.000E+00 0.100E+21
     0.000     0.000                 0.0
⋮
*END
```

Example 2 - Mesh checking/correction only for rigid tool mesh (sheet blank not required):

A much shorter but complete input example of checking, correcting the tool meshes and reorienting all tools' normals is shown below. Note the sheet blank does not need to be present, and both rigid_offset.inc and rigid_offset_before.inc will be the same, representing the checked, corrected, and reoriented tool mesh file, since IOFFSET is undefined (no tool offset will be done).

```
*KEYWORD
*INCLUDE
toolmesh.k
*CONTROL_FORMING_AUTOCHECK
$   ICHECK       IGD   IOFFSET   IOUTOUT
         1                             1
*PARAMETER_EXPRESSION
I toolpid          3
*PART

$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
  &toolpid         2         2
*MAT_RIGID
$      MID        RO         E        PR         N    COUPLE         M     ALIAS
         2  7.83E-09  2.07E+05      0.28
$      CMO      CON1      CON2
         1         4         7
$LCO or A1        A2        A3        V1        V2        V3

*SECTION_SHELL
$    SECID    ELFORM      SHRF       NIP     PROPT   QR/IRID     ICOMP     SETYP
         2         2       1.0       3.0       0.0
$       T1        T2        T3        T4      NLOC
       1.0       1.0       1.0       1.0
*END
```

Example 3 - Mesh checking/correction and tool offset (sheet blank required):

In addition to checking, correcting and reorienting all tools' normal, the following partial input will offset the die meshes in toolmesh.k by 0.88 mm (using the MST value defined for the die) on the opposite side of the blank, and output the offset tool meshes in a file rigid_offset.inc. The checked/corrected original die meshes will be written to rigid_offset_before.inc. The simulation will terminate as soon as the files are written, regardless of what the “&endtime” value is. In fact, the keyword *CONTROL_TERMINATION can be omitted all together.

```
*KEYWORD
*INCLUDE
Tool_blank.k
*PARAMETER_EXPRESSION
R blankt         0.8
R offset        -1.1
R mst           blankt*offset*2.0
*CONTROL_FORMING_AUTOCHECK
$   ICHECK       IGD   IOFFSET   IOUTOUT
         1                   1         3
*CONTROL_TERMINATION
&endtime
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ID
         1  blank to punch
         1         2         2         2                             1         1
 0.110E+00 0.000E+00 0.000E+00 0.000E+00 0.200E+02          0.0000E+00 0.100E+21
     0.000     0.000               0.0
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ID
         2  blank to die
$     SSID      MSID     SSTYP     MSTYP    SBOXID    MBOXID       SPR       MPR
         1         3         2         2                             1         1
$       FS        FD        DC        VC       VDC    PENCHK        BT        DT
 0.110E+00 0.000E+00 0.000E+00 0.000E+00 0.200E+02           0.000E+00 0.100E+21
$      SFS       SFM       SST       MST      SFST      SFMT       FSF       VSF
     0.000     0.000                &mst
$     SOFT    SOFSCL    LCIDAB    MAXPAR    PENTOL     DEPTH     BSORT    FRCFRQ
         0
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ID
         3  blank to binder
         1         4         2         2                             1         1
 0.110E+00 0.000E+00 0.000E+00 0.000E+00 0.200E+02           0.000E+00 0.100E+21
     0.000     0.000               0.0
*END
```

Example 4 - Mesh checking/correction and tool offset, bead nodes output (sheet blank required):

In addition to checking, correcting and reorienting all tools’ normal, the following partial input will create an offset tool in the file rigid_offset.inc on the same side of the blank; the file will also contain the nodes used to define the contact draw beads #1 and #2.

```
*KEYWORD
*INCLUDE
Tool_blank.k
R blankt         0.8
R offset         1.1
R mst           blankt*offset*2.0
*CONTROL_FORMING_AUTOCHECK
$   ICHECK       IGD   IOFFSET   IOUTOUT
         1                   1         2
*CONTROL_TERMINATION
&endtime
```

```
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ID
         1  blank to punch
         1         2         2         2                             1         1
 0.110E+00 0.000E+00 0.000E+00 0.000E+00 0.200E+02          0.0000E+00 0.100E+21
     0.000     0.000                 0.0
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ID
         2  blank to die
$     SSID      MSID     SSTYP     MSTYP    SBOXID    MBOXID       SPR       MPR
         1         3         2         2                             1         1
$       FS        FD        DC        VC       VDC    PENCHK        BT        DT
 0.110E+00 0.000E+00 0.000E+00 0.000E+00 0.200E+02           0.000E+00 0.100E+21
$      SFS       SFM       SST       MST      SFST      SFMT       FSF       VSF
     0.000     0.000                &mst
$     SOFT    SOFSCL    LCIDAB    MAXPAR    PENTOL     DEPTH     BSORT    FRCFRQ
         0
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ID
         3  blank to binder
         1         4         2         2                             1         1
 0.110E+00 0.000E+00 0.000E+00 0.000E+00 0.200E+02           0.000E+00 0.100E+21
     0.000     0.000                 0.0
⋮
*CONTACT_DRAWBEAD_ID
     10001  Draw bead #1
         1         1         4         2                   0         0         0
 0.110E+00 0.000E+00 0.000E+00 0.000E+00 0.200E+00                0.0615 0.100E+21
     0.200     0.200
$   LCIDRF    LCIDNF     DBDTH     DFSCL    NUMINT
        10         9 0.100E+02 0.700E+00
*CONSTRAINED_EXTRA_NODES_SET
        40         1
*SET_NODE_LIST
         1
    915110    915111    915112    915113    915114    915115    915116    915117
    915118    915119    915120    915121    915122
*CONTACT_DRAWBEAD_ID
     10002  Draw bead #2
         2         1         4         2                   0         0         0
 0.110E+00 0.000E+00 0.000E+00 0.000E+00 0.200E+00                0.0615 0.100E+21
     0.200     0.200
$   LCIDRF    LCIDNF     DBDTH     DFSCL    NUMINT
        11         9 0.100E+02 0.400E+00
*CONSTRAINED_EXTRA_NODES_SET
$      PID      NSID
        40         2
*SET_NODE_LIST
         2
    915123    915124    915125    915126    915127    915128    915129    915130
    915131    915132    915133    915134    915135    915136    915137
⋮
*END
```

Revision information:

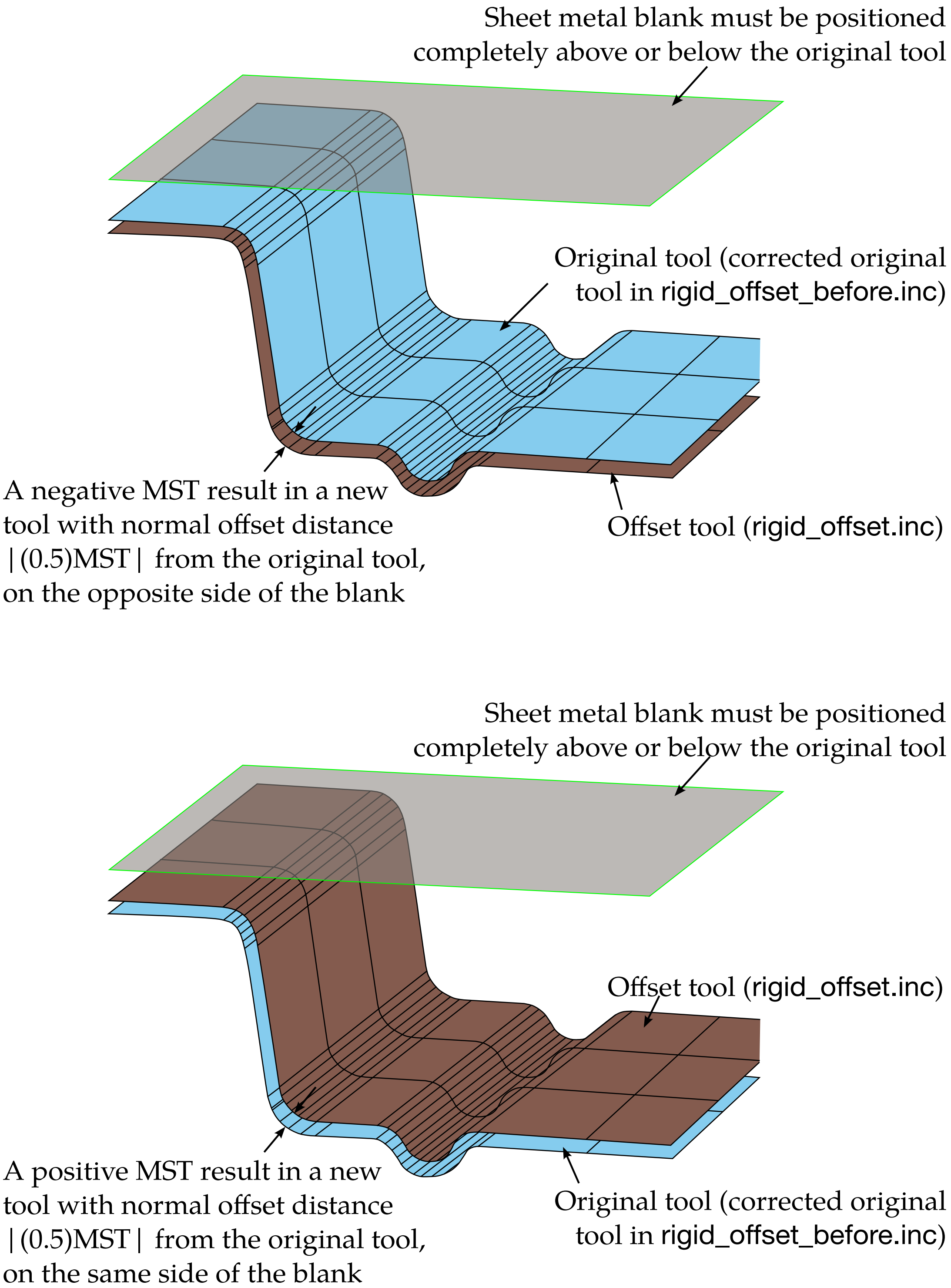
This feature is available starting from LS-DYNA Revision 91737, in both SMP, MPP and double precision.

1. IOFFSET, and IOUTPUT = 1 are available starting in Revision 94521. The latest beta revisions should offer better and improved offset meshes.

2. IOUTPUT = 2 is available starting in Revision 95357.

3. Support of SMOOTH contact option in *CONTACT_FORMING...: is available starting in Revision 95456.

4. IOUTPUT = 3, and 4 are available starting in Revision 96592.



Sheet metal blank must be positioned completely above or below the original tool

Original tool (corrected original tool in rigid_offset_before.inc)

A negative MST result in a new tool with normal offset distance |(0.5)MST| from the original tool, on the opposite side of the blank

Offset tool (rigid_offset.inc)

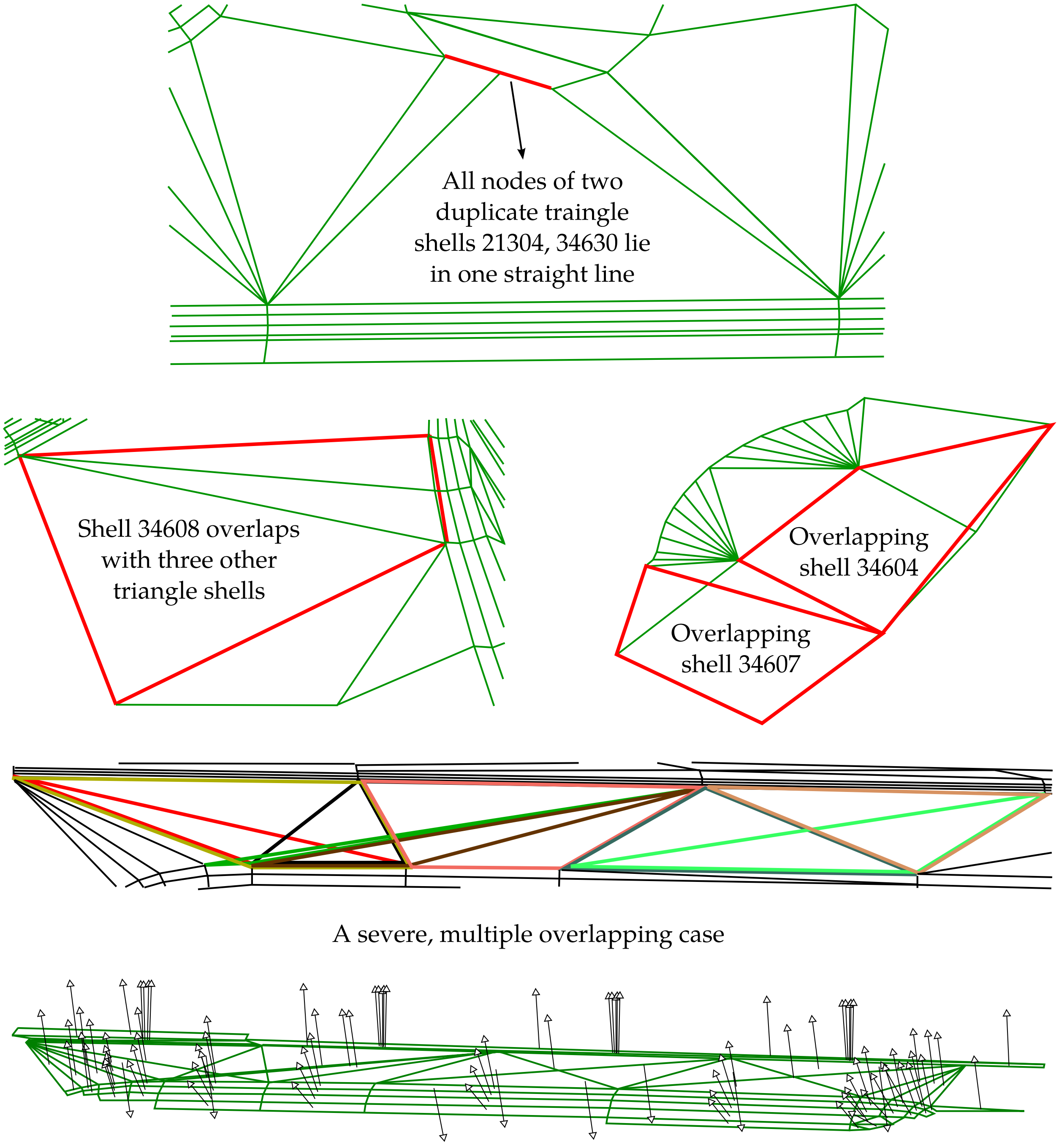
Sheet metal blank must be positioned completely above or below the original tool

Offset tool (rigid_offset.inc)

A positive MST result in a new tool with normal offset distance |(0.5)MST| from the original tool, on the same side of the blank

Original tool (corrected original tool in rigid_offset_before.inc)

Figure 12-14. Offset using the MST value defined in *CONTACT_-FORMING_…


All nodes of two duplicate traingle shells 21304, 34630 lie in one straight line
Shell 34608 overlaps with three other triangle shells
Overlapping shell 34604
Overlapping shell 34607

A severe, multiple overlapping case

A case of typical incoming, inconsistent shell normals

Figure 12-15. A few cases of the tooling mesh problems handled by this keyword.

*CONTROL_FORMING_AUTO_NET

Purpose: This keyword is used for simulating springback when the stamping panel is resting on the nets of a checking fixture. With this keyword, rectangular nets are automatically generated according to specified dimensions and positions.

Include one pair of Cards 1 and 2 per net. Add to the deck as many pairs of cards as needed. This section is terminated by the next keyword ("*") card. In general, for N nets add 2N cards.

Card 1	1	2	3	4	5	6	7	8
Variable	IDNET	ITYPE	IDV	IDP	X	Y	Z	
Type	I		I	I	F	F	F	
Default	none		0	0	0.0	0.0	0.0	

Card 2	1	2	3	4	5	6	7	8
Variable	SX	SY	OFFSET					
Type	F	F	F					
Default	0.0	0.0	0.0					

VARIABLE	DESCRIPTION
IDNET	ID of the net; must be unique.
ITYPE	Not used at this time.
IDV	Vector ID for surface normal of the net. See *DEFINE_VECTOR. If not defined, the normal vector will default to the global z-axis.
IDP	Part ID of the panel undergoing springback simulation.
X	The x-coordinate of a reference point for the net to be generated.
Y	The y-coordinate of a reference point for the net to be generated.

VARIABLE	DESCRIPTION
Z	The z-coordinate of a reference point for the net to be generated.
SX	Length of the net along the first tangential direction. (The x-axis when the normal is aligned along the global z-axis).
SY	Length of the net along the second tangential direction. (The y-axis when the normal is aligned along the global z-axis).
OFFSET	The net center will be offset a distance of OFFSET in the direction of its surface normal. For positive values, the offset is parallel to the normal; for negative values, antiparallel.

General remarks:

1. The IDNET field of card 1 sets the "net ID," which is distinct from the part ID of the net; the net ID serves distinguishes *this* net from *other* nets.

2. The part ID assigned to the net is generated by incrementing the largest part ID value in the model.

3. Other properties such as section, material, and contact interfaces between the panel and nets are likewise automatically generated.

4. The auto nets use contact type *CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE.

An example:

The excerpted input file (see below) specifies four auto nets having IDs 1 through 4. The vector with ID = 89 is normal to the net. The nets are offset 4 mm *below* their reference points; the direction is *below* because the normal vector (ID = 89) is parallel to the z-axis and the offset is negative. This example input can be readily adapted to a typical gravity-loaded springback simulation obviating the need for SPC constraints (see *CONSTRAINED_COORDINATE).

```
*CONTROL_FORMING_AUTO_NET
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$    IDNET     ITYPE       IDV       IDP         X         Y         Z
         1                  89         5   2209.82  -33.6332   1782.48
$       SX        SY    OFFSET
      15.0      15.0      -4.0
$    IDNET     ITYPE       IDV       IDP         X         Y         Z
         2                  89         5   3060.23  -33.6335   1782.48
$       SX        SY    OFFSET
      15.0      15.0      -4.0
$    IDNET     ITYPE       IDV       IDP         X         Y         Z
         3                  89         5   3061.21   31.4167   1784.87
$       SX        SY    OFFSET
```

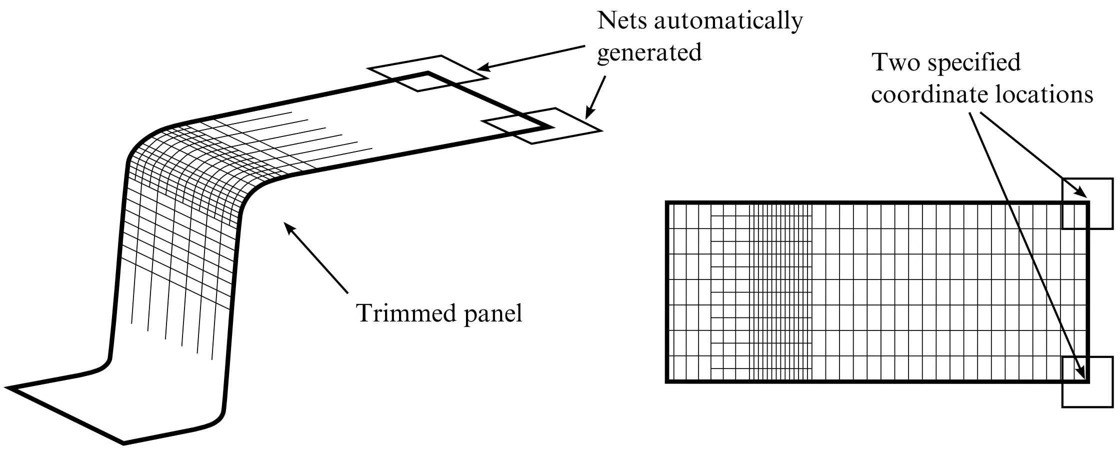

Nets automatically generated
Two specified coordinate locations
Trimmed panel

Figure 12-16. An example problem.

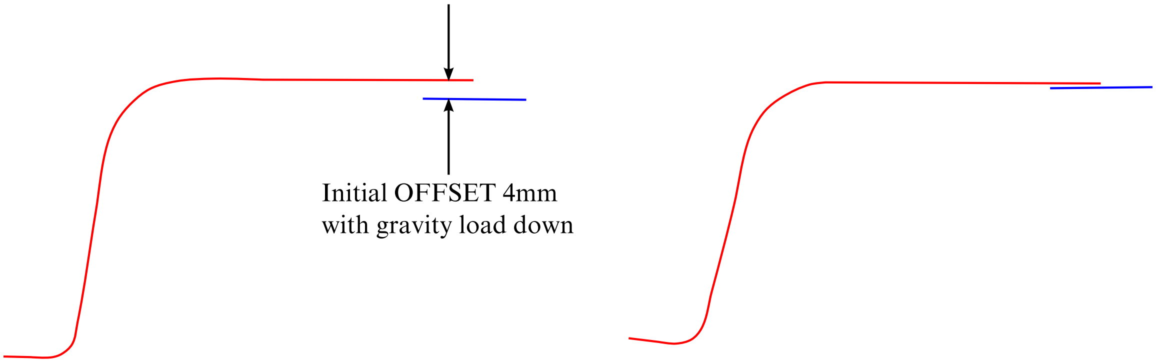
```
      15.0      15.0      -4.0
$    IDNET     ITYPE       IDV       IDP         X         Y         Z
         4                  89         5   2208.84   31.4114   1784.87
$       SX        SY    OFFSET
      15.0      15.0      -4.0
*DEFINE_VECTOR
$ VID, Tail X, Y, Z, Head X, Y, Z
89,0.0,0.0,0.0,0.0,0.0,100.0
```

Discussion of Figures:

Figure 12-16 shows a formed and trimmed panel of a hat-shaped channel with an auto net at two corners. The nets are offset 4mm away from the panel. When gravity loading is downward the nets must be below the panel (Figure 12-17 left) so that the panel comes into contact with the nets after springback as expected (Figure 12-17 right). As shown in Figure 12-18 the situation must be reversed when gravity loading points upward.

Revision information:

This feature is now available starting in implicit static in double precision LS-DYNA Revision 62781.

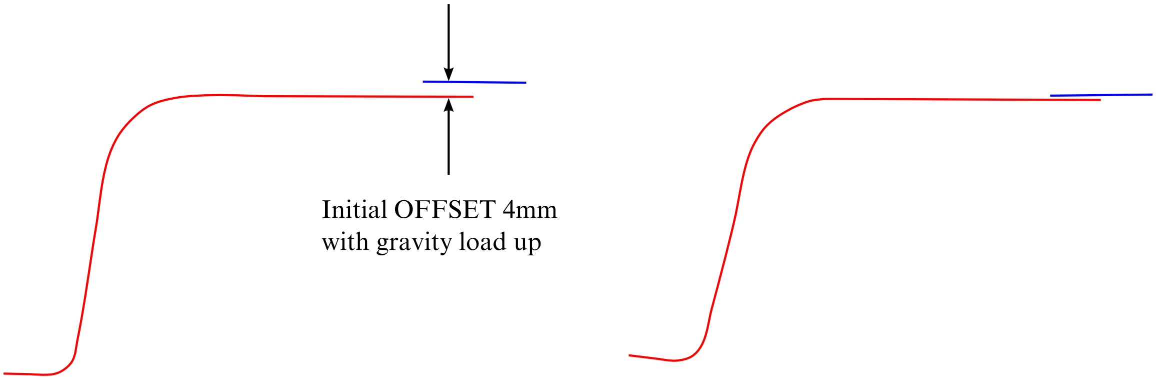


Initial OFFSET 4mm
with gravity load down

Before springback

After springback

Figure 12-17. Springback and contact with nets - gravity down.



Initial OFFSET 4mm
with gravity load up

Before springback

After springback

Figure 12-18. Springback and contact with nets – gravity up.

CONTROL_FORMING_AUTOPOSITION_PARAMETER_*{OPTION}

Available options include:

<BLANK>

SET

Purpose: The purpose of this keyword is to *calculate* the minimum required separation distances among forming tools for initial tool and blank positioning in metal forming simulation. It is applicable to shell elements only. It does not, actually, move the part; for that, see *PART_MOVE.

NOTE: This keyword requires that model begin in its home position. While processing this card, LS-DYNA moves the parts to match the auto-position results so that auto-position operations correctly compose. Upon completion of the auto-positioning phase, the parts are returned to their home positions.

Auto-Position Part Cards. Add one card for each part to be auto-positioned. The next keyword ("*") card terminates this is keyword.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	CID	DIR	MPID	POSITION	PREMOVE	THICK	PORDER
Type	I	I	I	I	I	F	F	I/A
Default	none	0	none	none	0	0.0	0.0	none

VARIABLE	DESCRIPTION
PID	Part ID. This part will be moved based on the following controlling parameters. When the option SET is activated, PID becomes part set ID, defined by *SET_PART_LIST. This is useful in defining tailor-welded blanks, where two pieces of the blank must be moved simultaneously.
CID	Coordinate ID (Default is global coordinate system).

VARIABLE **DESCRIPTION**

DIR

Direction in which the part will be moved:

EQ.1: x direction,

EQ.2: y direction,

EQ.3: z direction.

MPID

Master part ID, whose position is to be referenced by PID for positioning. When the option SET is activated, MPID becomes part set ID, defined by *SET_PART_LIST.

POSITION

Definition of relative position between PID and MPID:

EQ.1: PID is above MPID;

EQ.-1: PID is below MPID.

Definition of "above" is determined by the defined coordinate system. If PID is above MPID, it means PID has a larger z-coordinate. This definition is helpful in line die simulation where local coordinate system may be used.

PREMOVE

Move PID through distance PREMOVE *prior* to processing the other *CONTROL_FORMING_AUTOPOSITION cards. See Remark 5.

THICK

Thickness of the blank. The same value must be used in all defined move operations under this keyword.

PORDER

The name of the parameter without the ampersand "&", as defined in *PARAMETER, or the position or order of the parameter defined in the *PARAMETER list.

Background:

In line-die (multi-stage) simulation, initial positioning of the tools and blank is one of the major issues preventing several die processes from being run automatically from a single job submission. The most basic method for running a line die simulation is to chain a series of calculations together using the previous calculation's partially formed blank, written to a dynain file, as a part of the input for the next calculation.

Since the partial results are not known until the preceding calculation completes, the tools need to reposition before the next calculation. Without this card the repositioning step must be done by-hand using a preprocessor. With the combination of this card and the LS-DYNA case driver (see *CASE card) the repositioning can be fully automated, enabling a complete line-die simulation to be completed with a single job submission.

Workflow:

This card requires that all parts start in their home (tool closed) position. It calculates how far the parts need to be moved to prevent initial penetration. The results are stored into the parameter listed in the PORDER field to be used for a part move operation.

1. For each defined move operation a *PARAMETER card *must* initialize the parameter referred to in the PORDER field.

2. All tools must start in home position including *desired final gaps*.

3. The required distance between each contact pair is calculated and stored in the initialized parameter named in the PORDER field.

4. The parts are repositioned through a distance based on the value written to the parameter PORDER using the *PART_MOVE card.

5. The *PARAMETER_EXPRESSION can be used to evaluate expressions depending on the move distances, such as times and tool move speeds.

6. The *CASE feature, is used to chain together the sub-processes in the line-dime simulation.

Remarks:

1. **Order Dependence.** Input associated with this keyword is order sensitive. The following order should be observed:

   a) All model information *including* all elements and node

   b) Part definitions (see *PART)

   c) Part set definitions (see *SET_PART_LIST)

   d) *PARAMETER initialization

   e) This keyword

   f) *PARAMETER_EXPRESSION

   g) *PART_MOVE

2. This keyword can also be used to generate a new keyword input (dynain) containing the fully positioned model (without actually running the entire simulation). This procedure is identical to a full calculation except that the *PARAMETER_EXPRESSION keyword, the *CONTROL_TERMINATION keyword, and tool kinematic definitions are omitted.

3. When working in local coordinate systems it is often the case that the sign of the computed parameter may not correspond to its intended use. In this case, the absolute value function, ABS, for the *PARAMETER_EXPRESSION keyword is especially useful.

4. Draw beads can be modeled with beam elements that are positioned and attached to a tool at home position. Draw beads with beam elements can also be moved in the keyword *PART_MOVE, and automatically positioned just like any other types of elements.

5. Cards with the PREMOVE field set are processed before *all* other *CONTROL_-FORMING_AUTOPOSITON cards, regardless of their location in the input deck. The PREMOVE field serves to modify the initial state on which the calculations of the other AUTOPOSITON cards are based.

   For instance, when a binder is moved downward with the PREMOVE feature, it will be in its post-PREMOVE position for *all other* AUTOPOSITION calculations. But, as is the case with the other AUTOPOSITON cards, the model will be returned to its home position upon completion of the AUTOPOSITION phase. Note that the master part, MPID, and the POSITION fields are *ignored* when the PREMOVE field is set, and that the PREMOVE value is copied into the PORDER parameter.

6. This feature is implemented in LS-PrePost4.0 eZSetup (http://ftp.lstc.com/-anonymous/ outgoing/lsprepost/4.0/metalforming/) for metal forming in both explicit and implicit application.

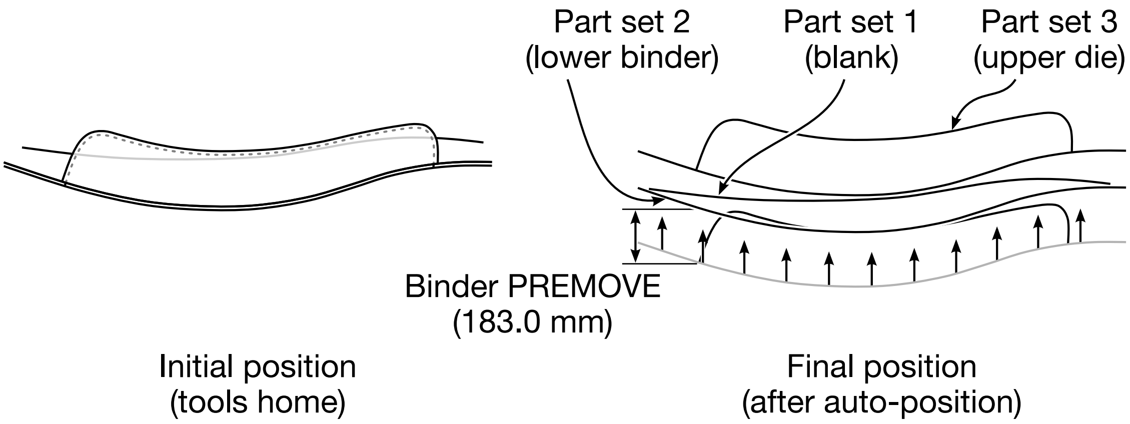


Figure 12-19. An example of using the variable PREMOVE

Example 1:

An air draw process like the one shown in Figure 12-19 provides a clear illustration of how this card, and, in particular, the PREMOVE field is used to specify the lower binder's travel distance.

1. The card with the PREMOVE field set, the *third* AUTOPOSITON card, is processed first. It moves lower binder 183 mm upward from its home position, and it will form the base configuration for other AUTOPOSITION cards. It will also store this move into &bindmv. Note that although the POSITION and MPID fields are set, they are ignored.

2. The *first* autoposition card, which will be the *second* one processed, calculates the minimum offset distance (&blankmv) necessary for the blank (part set 1) to clear part set 9999, which consists of the lower binder (PID = 2), which is in its post-PREMOVE location, and of the lower punch (PID = &lpunid).

3. The next card determines the minimum offset (&updiemv) necessary to bring the upper die (part set 3) as close to the blank as possible without penetrating. *This calculation proceeds under the assumption that the blank part set has been moved through &blankmv.*

```
*SET_PART_LIST
9999
&lpunpid,2
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*CONTROL_FORMING_AUTOPOSITION_PARAMETER_SET
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$      PID       CID       DIR      MPID  POSITION   PREMOVE     THICK    PORDER
$ blank move
         1                   3      9999         1             &bthick   blankmv
$ upper die move
         3                   3         1         1             &bthick   updiemv
$ lower binder move
         2                   3         1        -1     183.0   &bthick    bindmv
```

```
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*PART_MOVE
$    SID            XMOV            YMOV            ZMOV     CID   IFSET
$ blank move
       1             0.0             0.0        &blankmv              1
$ upper die move
       3             0.0             0.0        &updiemv              1
$ lower binder move
       2             0.0             0.0         &bindmv              1
```

The following examples demonstrates the *PARAMETER_EXPRESSION card, which is used to derive new parameters from the value calculated during auto-positioning. In this example, the auto-positioned distance for binder, which is stored in the parameter, &bindmv, is used to define an additional parameter,

$$\text{\&bindmv1} = \text{\&bindmv} - 30\ \text{mm}$$

The *PART_MOVE step uses &bindmv1 rather than &bindmv, to move both the lower binder and the draw beads.

```
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*CONTROL_FORMING_AUTOPOSION_PARAMETER_SET
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$      PID       CID       DIR      MPID  POSITION   PREMOVE     THICK  PORDER
$ blank move
   &blksid                   3      9999         1             &bthick   blankmv
$ upper die move
  &udiesid                   3   &blksid         1             &bthick   updiemv
$ lower binder move
  &bindsid                   3   &blksid        -1             &bthick   bindmv
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*PARAMETER_EXPRESSION
bindmv1   bindmv-30.0
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*PART_MOVE
$    SID            XMOV            YMOV            ZMOV     CID   IFSET
$ blank move
 &blksid             0.0             0.0        &blankmv              1
$ upper die move
&udiesid             0.0             0.0        &updiemv              1
$ lower binder move
&bindsid             0.0             0.0        &bindmv1              1
$ draw beads move
     909             0.0             0.0        &bindmv1              1
```

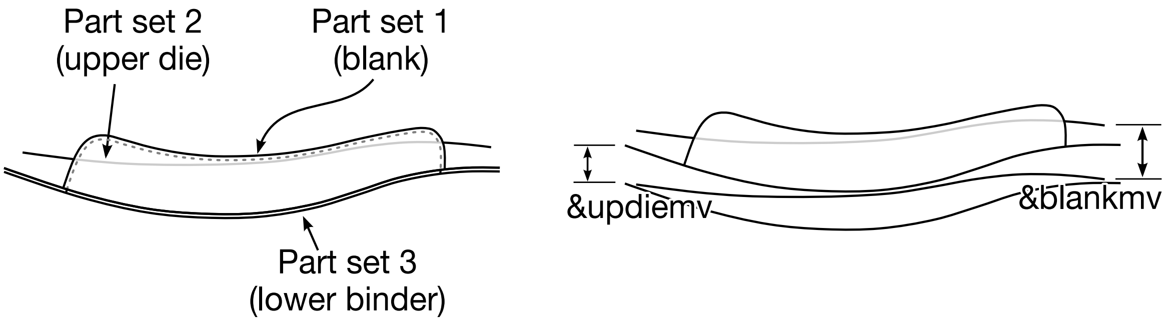


Figure 12-20. An example of binder closing in air draw

Example 2:

Figure 12-20 schematically shows the binder closing in the global Z-direction. A partial keyword details follow.

```
*INCLUDE
$blank from previous case
case5.dynain
*INCLUDE
closing_tool.k
*INCLUDE
beads_home.k
*SET_PART_LIST
$ blank
1
1
*SET_PART_LIST
$ upper die
2
2
*SET_PART_LIST
$ lower binder
3
3
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*parameter
$$$$$$$$$$$$$$$$$$$$$$$$$$ Tool move variables
R blankmv        0.0
R updiemv        0.0
R bindmv         0.0
$$$$$$$$$$$$$$$$$$$$$$$$$$ Tool speed and ramp up definition
R tclsup       0.001
R vcls        1000.0
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*CONTROL_FORMING_AUTOPOSION_PARAMETER_SET
$      PID       CID       DIR      MPID  POSITION   PREMOVE     THICK  PORDER
$ positioning blank on top of lower binder
         1                   3         3         1                 0.7   blankmv
$ positioning upper die on top of blank
         2                   3         1         1                 0.7   updiemv
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*PARAMETER_EXPRESSION
$    PRMR1    EXPRESSION
R clstime     (abs(updiemv)-vcls*tclsup)/vcls+2.0*tclsup
R endtime     &clstime
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*PART_MOVE
$    PID            XMOV            YMOV            ZMOV      CID
```

```
       1            0.0             0.0        &blankmv
       2            0.0             0.0        &updiemv
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*CONTROL_TERMINATION
&endtime
```

Revision information:

This feature is available starting in LS-DYNA Revision 56080 in both explicit and implicit, SMP and MPP versions. Later revisions are also available with various improvements.

*CONTROL_FORMING_BESTFIT

Available options include:

<BLANK>

VECTOR

Purpose: This keyword rigidly moves a part to the target so that they maximally coincide. This feature can be used in sheet metal forming to translate and rotate a spring back part (source) to a scanned part (target) to assess spring back prediction accuracy. This keyword applies to shell elements only. The VECTOR option allows vector components of the normal distance from the target to the part node to be included in the output file bestfit.out under the keyword *NODE_TO_TARGET_VECTOR (see corresponding manual pages).

This feature is available now in *LS-PrePost 4.3* in *Metal Forming Application/eZ Setup* (http://ftp.lstc.com/anonymous/outgoing/lsprepost/4.3/win64/).

Card 1	1	2	3	4	5	6	7	8
Variable	IFIT	NSKIP	GAPONLY	IFAST	IFSET	NSETS	NSETT	
Type	I	I	I	I	I	I	I	
Default	0	-3	0	1	0	none	none	

Card 2	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	A80							
Default	none							

VARIABLE	DESCRIPTION
IFIT	Best fit program activation flag: IFIT.EQ.0: do not perform best-fit. IFIT.EQ.1: activate the best-fit program.

VARIABLE **DESCRIPTION**

NSKIP

Optional skipping scheme during bucket searching to aid the computational speed (zero is no skipping):

NSKIP.GT.0: Number of nodes to skip in bucket searching. NSKIP of "1" does not skip any nodes in searching therefore computing speed is the slowest but accuracy is the highest. Higher values of NSKIP speed up the calculation time with slightly deteriorating accuracies. Based on studies, a value of "5" is recommended with IFAST = 1, which balances the speed and accuracy. See Table 12-21 for the effect of NSKIP on the accuracy of the fitting.

NSKIP.LT.0: Absolute value is the distance to skip in bucket searching. This scheme is faster compared to the previous method and therefore **is recommended** for computational efficiency and accuracy. A value of "-5" is suggested. See Example 3.

NSKIP	IFAST = 0		IFAST = 1	
	CPU time	Max/Min (mm)	CPU time	Max/Min (mm)
2	10 min 38 sec	1.28/-1.59	4 min 3 sec	1.22/-1.59
5	4 min 49 sec	1.21/-1.59	1 min 59 sec	1.25/-1.61
10	2 min 46 sec	1.27/-1.59	1 min 18 sec	1.44/-1.53
20	1 min 24 sec	1.27/-1.59	59 sec	1.42/-1.64
50	50 sec	1.22/-1.61	40 sec	1.43/-1.67

Table 12-21. Computing speed and the max/min deviations from the springback mesh to the target scan for an automotive part, under various combinations of NSKIP and IFAST. All runs were made on a 1 CPU XEON E5520 machine, with 685132 elements on the target scan and 135635 elements on the springback mesh.

VARIABLE	DESCRIPTION
GAPONLY	Separation distance calculation flag: GAPONLY.EQ.0: perform best-fit, calculate separation distances between the two best-fitted mesh parts. GAPONLY.EQ.1: no best-fit, just calculate separation distances between the two existing mesh parts. GAPONLY.EQ.2: User is responsible to move the parts closer in distance and orientation, in situation where target and source are not similar in shape. Also see NSETS and NSETT (recommended method).
IFAST	Computing performance optimization flag: IFAST.EQ.0: no computing speed optimization. IFAST.EQ.1: activate computing speed optimization (default), and is recommended. See Table 12-21 for detailed speed performance data.
IFSET	Optional flag to define a node set to be included or excluded in the source mesh file for best fitting. The node set can be defined in a file together with the source mesh. A node set can be defined using *LS-PrePost* via menu options *Model→CreEnt→Set Data→*SET_NODE→Cre.* IFSET.EQ.0: all nodes in the source mesh file will be best fitted. IFSET.GT.0: the input value is a node set ID; only the nodes in

VARIABLE	DESCRIPTION
	the set will be best fitted. IFSET.LT.0: the absolute value is a node set ID; all nodes excluding those in the set will be best fitted. See Example 2.
NSETS	An optional node set ID of three nodes from the source mesh. The nodes should be selected based on distinctive geometry features, such as, the center of an arc, the center of a dart, or the end node of a take-up bead (see Example 3 and Figure 12-22). The three nodes must not be aligned in one straight line. Define NSETS if the orientation of the source mesh deviates from the target is large (>~30 degrees in any direction). This is the recommended method.
NSETT	An optional node set ID from the target mesh, consists of the corresponding three nodes from the same geometry features of the source mesh. The three nodes should be input in the same order as those from the source mesh. Approximate locations are acceptable. Define NSETT only if NSETS is defined. See Example 3 and Figure 12-22 for details. This is the recommended method.
FILENAME	Target mesh file in keyword format, where only *NODE and *ELEMENT_SHELL should be included. The target mesh is typically the scanned part converted from the *STL* format file. STL file format can be imported into *LS-PrePost* via *File→Import→STL File*, then a keyword format mesh file can be saved.

Remarks:

In springback prediction and compensation process simulation, there is always a need to assess the accuracy of the springback prediction using physical white-light scanned parts. Scanned parts are typically given in the *STL* format, which can be imported into *LS-PrePost* and written back out as a keyword mesh file.

The converted scanned keyword file can be used as FILENAME as a target mesh in an input file (see Example 1). The predicted springback mesh (source), consisting of *NODE, *ELEMENT_SHELL, *CONSTRAIN_ADAPTIVITY cards only, can be included in the input file using *INCLUDE. The best-fit program uses an iterative least-squares method to minimize the separation distances between the two parts, eventually transforming the springback mesh (source) into the position of the target mesh (scan). The normal distances between the two parts are calculated after the best-fitting, and stored as thickness values in a file bestfit.out, which is essentially a dynain file.

Both positive and negative distances are calculated and stored as the *Thickness*. Color contours of the normal distances between the two parts can then be plotted using *COMP→Thickness*. Positive distance means the source mesh is above the target mesh in a larger coordinates, and negative distance is below the target mesh in a smaller coordinates. For areas where no corresponding meshes can be found between the two parts, the distances are set to nearly zero. The fitting accuracy is within 0.02mm.

To reduce the computing time , the scan file (*STL*) mesh can be coarsened in a scan-processing software from a typically very dense mesh to a more reasonably sized mesh. In any case, the coarser mesh should be selected as the target mesh for optimal computational speed.

The fitted mesh bestfit.out and target mesh parts can both be imported into *LS-PrePost*. Using the *SPLANE* feature in *LS-PrePost*, multiple sections can be cut on both parts to assess springback deviations on a cut-section basis.

It is suggested that the orientation of the included file (source) should be within 30 degrees in any direction of the target file. In addition, the more rotations needed to re-orient the include file to align with the target file, the more CPU time will it take to complete the best fitting.

In case the source mesh orients more than 30 degrees in any directions of the target mesh, NSETS and NSETT can be used to initially align the source mesh to the target mesh before a full best-fit is performed. See Example 3 and Figure 12-22.

Example 1 – fitting with all nodes from the included file:

A complete input example is provided below to best fit a source mesh part spbk_NoSS.k to the target mesh part scan.k. NSKIP is set to "-5" and speed optimization is activated by setting IFAST to "1".

```
*KEYWORD
*CONTROL_FORMING_BESTFIT
$     IFIT     NSKIP   GAPONLY     IFAST     IFSET
         1        -5         0         1         0
scan.k
*INCLUDE
spbk_NoSS.k
*END
```

Example 2 – fitting with an excluded node set:

From the previous example, the included source file spbk_NoSS.k now consists of node set 128. The node set, which is defined in the file spbk_NoSS.k, which may feature geometry that are not a part of the target mesh, is being excluded (IFSET = -128) from participating in the best fitting. Alternatively, the unwanted nodes can be just deleted from the source file.

```
*KEYWORD
*CONTROL_FORMING_BESTFIT
$     IFIT     NSKIP   GAPONLY     IFAST     IFSET
         1        -5         0         1      -128
scan.k
*INCLUDE
spbk_NoSS.k
*END
```

Example 3 – fitting with NSETS and NSETT (recommended):

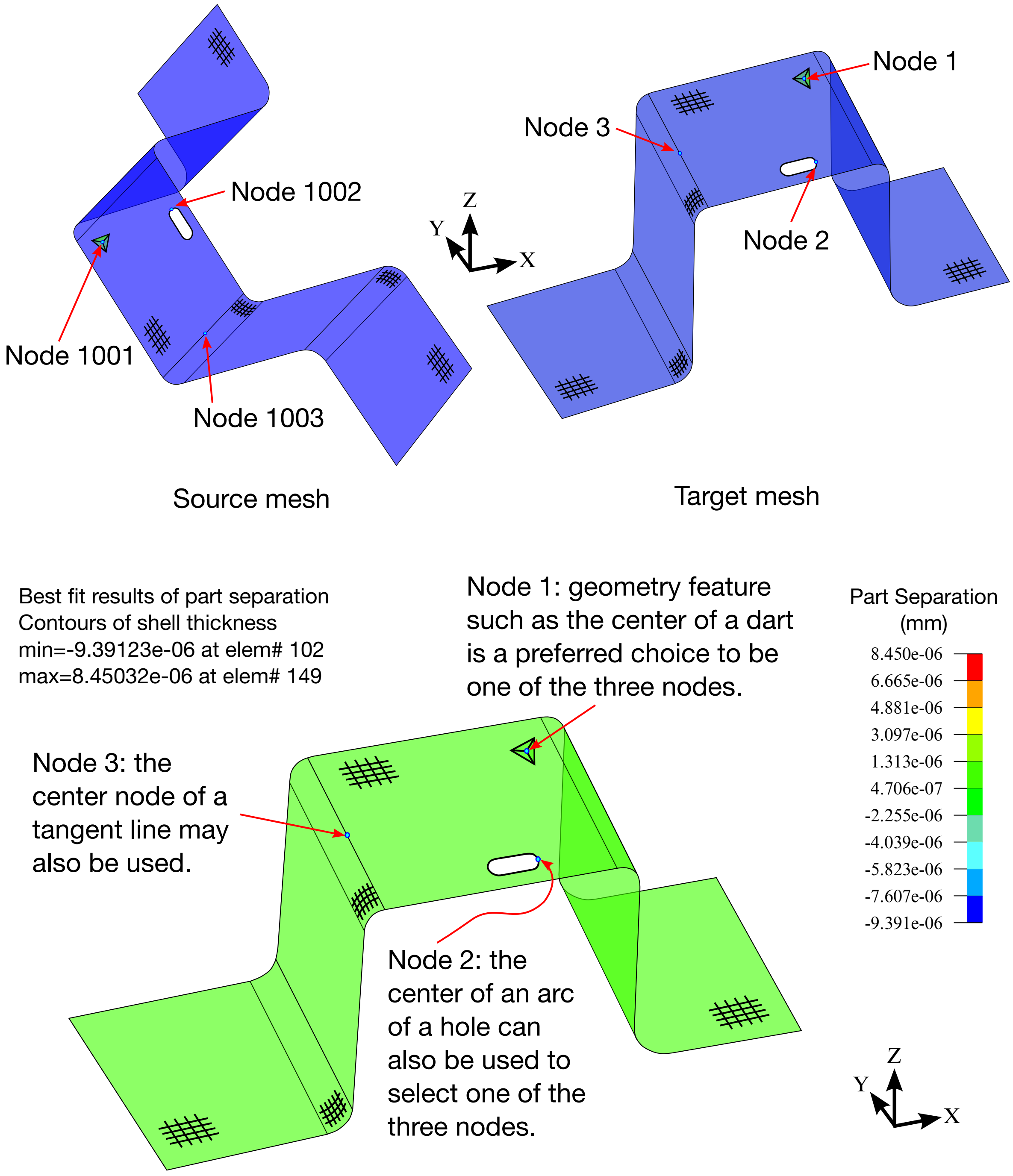
In the following partial keyword example (shown in Figure 12-22) a source mesh sourcemesh.k is being best fitted to a target mesh targetmesh.k. A node set with ID 1 on the source mesh is defined consisting of nodes 1001 1002 and 1003 and a corresponding node set with ID 2 on the target mesh is defined and consists of nodes 1, 2 and 3.

Node ID 1001 and 1 are both located at the center of a dart on the top surface of the hat-shaped part. Node ID 1002 and 2 are selected at the center of an arc of an cutout hole. Lastly, node ID 1003 and 3 are at the center of a tangent line of a radius. With the NSKIP set a "-5", the search will be done skipping every 5 mm of distance. In this example, since the source and target meshes are exactly the same, the normal distance, as displayed by "*thickness*" is nearly zero everywhere.

```
*CONTROL_FORMING_BESTFIT
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$#    IFIT     NSKIP   GAPONLY     IFAST     IFSET     NSETS     NSETT
         1        -5         0         1         0         1         2
$# FILENAME
targetmesh.k
*INCLUDE
sourcemesh.k
*SET_NODE_LIST
1
1,2,3
*SET_NODE_LIST
2
1001,1002,1003
```

Revision information:

This feature is available starting from LS-DYNA Revision 96427 double precision SMP. The variable IFSET is available starting from Revision 96696. The variables NSETS, NSETT are available starting from Revision 99369. The VECTOR option is available starting from Revision 112655.



Source mesh

Target mesh

Best fit results of part separation
Contours of shell thickness
min=-9.39123e-06 at elem# 102
max=8.45032e-06 at elem# 149

Node 1: geometry feature such as the center of a dart is a preferred choice to be one of the three nodes.

Node 3: the center node of a tangent line may also be used.

Node 2: the center of an arc of a hole can also be used to select one of the three nodes.

Best fit results - color contour of part separation plotted with "thickness" from the output file "Bestfit.out"

Figure 12-22. Best fit of two meshes with orientations greater than 30 degrees from each other.

*CONTROL_FORMING_INITIAL_THICKNESS

Purpose: This keyword is used to specify a varying thickness field in a specific direction on a sheet blank (shell elements only) as a result of a metal forming process such as a tailor-rolling, to be used for additional metal forming simulation. Another related keyword includes *ELEMENT_SHELL_THICKNESS.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	LCID	X0	Y0	Z0	VX	VY	VZ
Type	I	I	F	F	F/I	F	F	F
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
PID	Part ID of the sheet blank to be defined with varying thickness, as in *PART. Currently only 1 PID is allowed.
LCID	Load curve ID defining thickness (Y-values) vs. distance (X-values) starting from position coordinates (X0, Y0, Z0) and in the direction of a vector [VX, VY, VZ], as in *DEFINE_CURVE.
X0, Y0, Z0	Starting position coordinates.
VX, VY, VZ	Vector components defining the direction of the distance in the load curve.

Background:

Tailor-rolling is a process used to vary the thickness of the blank. A judiciously designed and manufactured tailor-rolled blank will reduce the number of parts (reinforcements) involved in the stamping process, as well as the number tools needed to make them. By reducing the number of spot welds, tailor-rolled pieces also possess superior structural integrity.

Remarks:

1. Beyond the last data point LS-DYNA extrapolates the load curve specified in LCID as being constant.

2. This card overrides thicknesses set with the *SECTION_SHELL keyword.

Application example:

An excerpt from an input deck containing a characteristic example of this card's application is given below. In this example the blank is part ID 1. The axis of the load curve starts at position $(-295, -607,\ -43)$ and the direction along which the load curve sets the thickness is given by $(524, 607, 0)$. For each of the load curve's abscissa values, t, the corresponding geometrical coordinate is given by:

$$\boldsymbol{r} = \begin{bmatrix} -295 \\ -607 \\ -43 \end{bmatrix} + \begin{bmatrix} 524 \\ 607 \\ 0 \end{bmatrix} t$$

For negative values along the load curve, $t < 0$, and values of $t > 101.0$, the thickness is extrapolated as a constant value of 0.8, and 0.9, respectively.

```
*CONTROL_FORMING_INITIAL_THICKNESS
$      PID       LCID        X0        Y0        Z0        VX        VY        VZ
         1       1012    -295.0    -607.0     -43.0     524.0     607.0       0.0
*DEFINE_CURVE
1012
0.0, 0.8
21.0, 0.9
43.0, 1.0
65.0, 1.1
82.0, 1.0
101.0, 0.9
```

In Figure 12-23, a sheet blank is defined with a varying thickness across its surface in a vector direction pointed from the start to end point. The thickness variation vs. the distance from starting point in section A-A is shown in Figure 12-24.

Revision information:

This feature is available in LS-DYNA starting in Revision 82990.

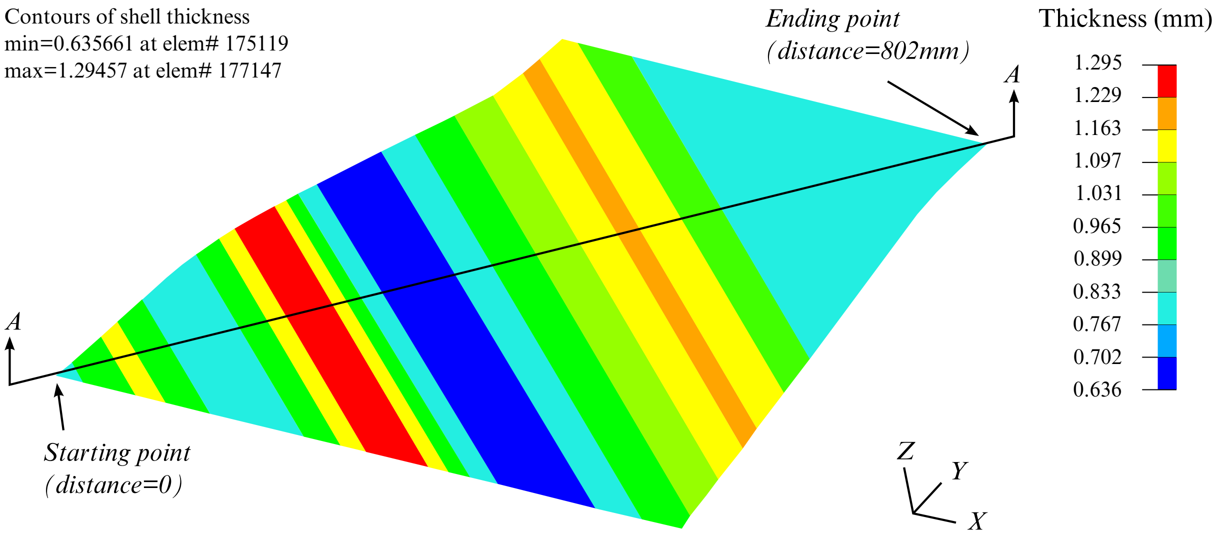


Figure 12-23. Define a varying thickness field across the sheet blank.

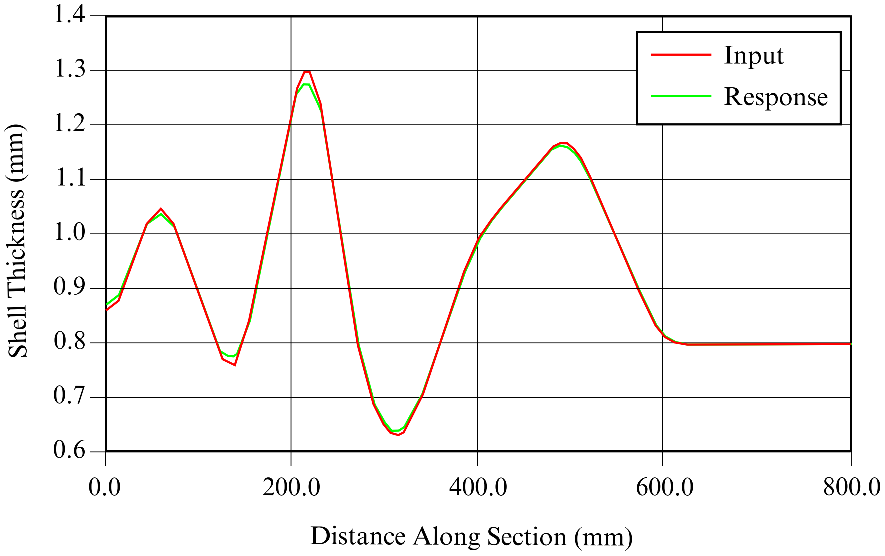


Figure 12-24. Thickness variation across section A-A

*CONTROL_FORMING_MAXID

Purpose: This card sets the node and element ID numbers for an adaptive sheet blank. The new node and element number of the adaptive mesh will start at the values specified on this card, typically greater than the last node and element number of all tools and blanks in the model. This keyword is often used in multi-stage sheet metal forming simulation. The *INCLUDE_AUTO_OFFSET keyword is related.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	MAXIDN	MAXIDE					
Type	I	I	I					
Default	none	none	none					

VARIABLE	DESCRIPTION
PID	Part ID of the sheet blank, as in *PART.
MAXIDN	Node ID number from which adaptive node ID numbers will be created.
MAXIDE	Element ID number from which adaptive element ID numbers will be created.

Remarks:

In a multi-stage automatic line die simulation the adaptivity feature may generate node and element IDs that collide with those of the tools used in the later stages of the process. Before the calculation begins, the set of IDs used by the tools is known. By setting MAXIDN to a value greater than the largest tool node ID and MAXIDE to a value greater than the largest tool element ID, it is guaranteed that refinement during the early stages will not lead to conflicts with tool IDs in the later stages.

The following example shows this feature applied in a 2D trimming simulation. Nodes and elements ID numbers generated from an adaptive trim simulation will be larger than the specified ID numbers of 5921980 and 8790292, respectively, for a sheet blank with part ID of 4.

```
*KEYWORD
*INCLUDE_TRIM
sim_trimming.dynain
⋮
```

```
*CONTROL_ADAPTIVE_CURVE
$    IDSET     ITYPE         N      SMIN
   &blksid         2         2       0.6
*CONTROL_CHECK_SHELL
$      PSID    IFAUTO    CONVEX      ADPT    ARATIO     ANGLE      SMIN
  &blksid1         1         1         1  0.250000150.000000  0.000000
*INCLUDE
EZtrim.k
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*DEFINE_CURVE_TRIM_NEW
$#    tcid    tctype      tflg      tdir     tctol      toln    nseed1    nseed2
     90914         2         0         1  1.250000  1.000000         0         0
sim_trimming_trimline_01.igs
*DEFINE_VECTOR
$#     vid        xt        yt        zt        xh        yh        zh       cid
         1     0.000     0.000     0.000     0.000     0.000  1.000000         0
*CONTROL_FORMING_MAXID
$       pid    maxidn    maxide
         4   5921980   8790292
*END
```

Revision Information:

This feature is available starting in LS-DYNA Revision 84159.

CONTROL_FORMING_ONESTEP_*{OPTION}

Purpose: This keyword activates a one-step solution using the *total strain theory* approximation to plasticity (also known as deformation theory) to implement an inverse method. Given the *final* geometry, the one-step method uses LS-DYNA's implicit statics solver to compute an approximate solution for (1) the stresses and strains in the formed part, (2) the thickness of the formed part, and (3) the size of the initial blank (unfolded flat blank). This method is useful for estimating the initial blank size with attendant material costs, and for augmenting crashworthiness models to account for metal forming effects, such as plastic strains and blank thickness in crash simulation.

NOTE: The input must contain only one "part", consisting entirely of shells, which is taken to be the *final* geometry.

1. This "part" may involve more than one PID to accommodate welded blanks,
2. it must be composed entirely of shells, and,
3. its external boundary must consist of a single closed loop.

Keywords associated with *CONTROL_FORMING_ONESTEP are:

*CONTROL_FORMING_UNFLANGING

*INTERFACE_BLANKSIZE_DEVELOPMENT.

Available options include:

<BLANK>

AUTO_CONSTRAINT

DRAWBEAD

FRICTION

TRIA

QUAD

QUAD2 (default)

Summary of keyword options:

1. The AUTO_CONSTRAINT option excludes rigid body motion from the implicit solution by automatically adding nodal constraints. A deck with a *CONTROL_FORMING_ONESTEP card should contain at most one *CONTROL_-FORMING_ONESTEP_AUTO_CONSTRIANT card. In addition, starting from Revision 91229, three nodes can be specified on the final part to position the unfolded blank for easier blank nesting, and for blank alignment in forming simulation.

2. The DRAWBEAD option is used to apply draw bead forces in addition to those provided by AUTOBD field in Card 1. A deck containing a *CONTROL_-FORMING_ONESTEP card may contain as many *CONTROL_FORMING_ONESTEP_DRAWBEAD cards as there are draw beads to be defined.

3. The FRICTION option applies friction along the edge of the part based on the binder tonnage input by the user in the DBTON field of card 1. A deck containing a *CONTROL_FORMING_ONESTEP card may contain as many *CONTROL_FORMING_ONESTEP_FRICTION cards as there are friction node sets to be defined.

4. Originally all quadrilateral elements in the model were split into two triangular elements internally for calculation. This original formulation is set as option TRIA as of Revision 112682. The option QUAD supports quadrilateral elements and implements some improved algorithm, which result in better results. In addition, this option greatly improves calculation speed under multiple CPUs in SMP mode, and is available starting in Revision 112071. The option QUAD2 is yet another improvement over the option QUAD with enhanced element formulation, which further improves results in terms of thinning and plastic strain with slightly longer CPU times. Calculation speed comparisons among the three options can be found in Performance among options TRIA, QUAD. The option QUAD2 is set as a default as of Revision 112682 and is the recommended option.

Card 1 for no option, <BLANK>.

Card 1	1	2	3	4	5	6	7	8
Variable	OPTION	TSCLMAX	AUTOBD	TSCLMIN	EPSMAX		LCSDG	DMGEXP
Type	I	F	F	F	F		I	F
Default	6	1.0	0.0	1.0	1.0		none	none

Card 1 for option AUTO_CONSTRAINT.

Card 1	1	2	3	4	5	6	7	8
Variable	ICON	NODE1	NODE2	NODE3				
Type	I	I	I	I				
Default	none	none	none	none				

Card 1 for option DRAWBEAD.

Card 1	1	2	3	4	5	6	7	8
Variable	NDSET	LCID	TH	PERCNT				
Type	I	I	F	F				
Default	none	none	0.0	0.0				

Card 1 for option FRICTION.

Card 1	1	2	3	4	5	6	7	8
Variable	NDSET	BDTON	FRICT					
Type	I	F	F					
Default	none	0.0	0.12					

Card 2 for no option <BLANK>.

Card 2	1	2	3	4	5	6	7	8
Variable	FLATNAME							
Type	A							
Default	none							

VARIABLE **DESCRIPTION**

OPTION

Options to invoke the one-step solution methods which account for undercut conditions in the formed part:

EQ.6: One-step solution with unfolded blank (flat) provided by LS-PrePost (see remarks). Card #2 is required.

EQ.7: One-step solution with blank automatically unfolded in LS-DYNA. Card #2 is a blank line. This option is recommended.

L.T.0: If a negative sign precedes any of the above OPTIONs, the stress and strain output in the file onestepresult will be in a large format (E20.0), which leads to more accurate stress results.

TSCLMAX

If not zero, it defines a thickness scale factor limiting the maximum thickness in the part.

For example, if the maximum thickness allowed is 0.8mm for a blank with initial thickness of 0.75mm TSCLMAX can be set to 1.0667. All thicknesses that are computed as more than 0.8mm in the sheet blank will be reset to 0.8mm. The scale factor is useful in advance feasibility analysis where part design and stamping process have not been finalized and could potentially cause large splits or severe wrinkles during unfolding, rendering the forming results unusable for crash/safety simulation.

VARIABLE **DESCRIPTION**

AUTOBD

Apply a fraction of a fully locked bead force along the entire periphery of the blank. The fully locked bead force is automatically calculated based on a material hardening curve input. AUTOBD can be increased to easily introduce more thinning and effective plastic strain in the part.

LT.0.0: Turns off the "auto-bead" feature.

EQ.0.0: Automatically applies 30% of fully locked force.

GT.0.0: Fraction input will be used to scale the fully locked force.

TSCLMIN

If not zero, it defines a thickness scale factor limiting the maximum thickness reduction.

For example, if the minimum thickness allowed is 0.6mm for a blank with initial thickness of 0.75mm TSCLMIN can be set to 0.8. All thicknesses that are computed as less than 0.6mm in the sheet blank will be reset to 0.6mm. The scale factor is useful in advance feasibility analysis where part design and stamping process have not been finalized and could potentially cause large splits or severe wrinkles during unfolding, rendering the forming results unusable for crash/safety simulation.

EPSMAX

If not zero, it defines the maximum effective plastic strain allowed. All computed effective plastic strains that are greater than this value in the blank will be set to this value.

LCSDG

Load curve ID defining equivalent plastic strain to failure vs. stress triaxiality, see *MAT_ADD_EROSION.

DMGEXP

Exponent for nonlinear damage accumulation, see *MAT_ADD_EROSION. Damage accumulation is written as history variable #6 in the file onestepresult.

ICON

Automatic nodal constraining option to eliminate the rigid body motion:

EQ.1: Apply.

VARIABLE DESCRIPTION

NODE[1,2,3] Node IDs for which the position is fixed during the unfolding. The position of these nodes in the calculated unfolded piece will coincide with the corresponding nodes in the input. The transformed and unfolded blank will be written in a keyword file "repositioned.k". *When these fields are undefined the orientation of the unfolded blank is arbitrary.*

NDSET Node set ID along the periphery of the part, as defined by keyword *SET_NODE_LIST.

LCID Load curve ID that defines the material hardening curve.

TH Thickness of the unformed sheet blank.

PERCNT Draw bead lock force fraction of the fully locked bead force.

BDTON Binder tonnage used to calculate friction force.

FRICT Coefficient of friction.

FLATNAME File name of the initial unfolded blank by LS-PrePost (see remarks). This is needed only for the OPTION = 6. Leave a blank line for OPTION = 7.

About One-Step forming solution:

One-step solution employs the total strain (or deformation) theory of plasticity in place of the more realistic incremental strain (or flow) theory. The total deformation theory expresses stress as a function of total strain; whereas the incremental strain theory requires that LS-DYNA compute a stress update at each time step (strain increment) from the deformation that occurred *during that time step*. In deformation theory, the results, therefore, do not depend on strain path, forming history, or the details of the stamping process.

When this card is included, the input must contain the *final geometry* from which LS-DYNA calculates the initial flat state using the inverse method. The one-step solution results can get close to the incremental results only when the forming process involves a linear strain path for which the deformation is either monotonically increasing or decreasing. In most cases total strain theory does not match incremental forming.

Path independence leads to several key simplifications:

1. Binder and addendum geometry are not required. There is no need to measure or model these geometries.

2. The solution is independent of stamping die processes (including part tipping).

3. There is no need for contact treatment since there are no tools and dies involved.

The one-step solution is mostly used for advance formability studies in which the user needs to quickly compare a wide range of different design alternatives. With this method the user can evaluate blank size, estimate material cost, and generate a first guess for blank size development (see also *CONTROL_FORMING_UNFLANGING, and *INTERFACE_BLANKSIZE_DEVELOPMENT). This method is also widely used to initialize forming stresses and strains in crash and occupant safety analysis.

Input details:

1. **Mesh.** In addition to the usual material and physical property definitions, this method requires that the final part be fully meshed using shell elements. This mesh must satisfy a different set of requirements than the tooling mesh. In particular, along the part bend radius, there is no need to build six elements along the arc length as one would do for the punch/die radius; two elements may be enough. A mesh consisting of uniformly distributed quadrilateral shell elements is ideal. *All elements in the mesh must also have normal consistency*.

   With *LS-PrePost 4.0*, this kind of mesh can be generated using *Mesh* → *AutoM* → *Size*. Since this method uses an implicit static solution scheme, the computational cost is controlled by the number of elements; element size has no effect. Furthermore, it is important to note that if one wants to obtain forming results that are closer to the incremental forming results, the part in the one-step input should be similar in size to the final formed blank shape in the incremental forming (before trimming).

2. **Holes.** Any trimmed-out holes can be filled (but not necessary). The filling can be done semi-automatically using *LS-PrePost 4.0* by selecting *Mesh* → *EleGen* → *Shell* → *Shell by Fill_Holes* → *Auto Fill*. The filled area of the part can be saved in a different part, as multiple parts (PID) are allowed. The forming results may depend on whether or not the holes are filled.

3. **Unfolding.** For OPTION = 6, the unfolded blank can be obtained from *LS-PrePost* via *EleTol* → *Morph* → *Type* = *Mesh_Unfolding* → *Unfold*. The unfolded mesh can be saved as a keyword file and used as input (see the FLATNAME field in Card 2). With OPTION = 7, LS-DYNA unfolds the mesh itself.

4. **Element Formulation.** Shell element of type 2 and 16 are supported. Since this feature uses the implicit method, type 16 is more convergent, computationally efficient, and, therefore, strongly recommended. Results are output on all integration points, as seen in the ELFORM and NIP variables in *SECTION_SHELL.

5. **Supported Materials.** Currently, *MAT_024, and *MAT_037 are supported. The user *must* provide a material hardening curve either in the LCSS field of *MAT_024 or in the HLCID field of *MAT_037. For *MAT_024 tables are supported. Future releases will add support for bilinear hardening with the ETAN feature. Additionally, in *MAT_024, strain rate is ignored, even when the variables C, and P are set.

6. **Boundary Conditions.** The primary "boundary/loading condition" for the one-step solution is the draw bead forces, which are set with the AUTOBD field or with the DRAWBEAD keyword option.

   a) With the DRAWBEAD option, draw bead forces are applied on a user defined node set (see NDSET). A fraction of the full lock force, determined by the tensile strength and sheet thickness, can be specified. The larger the fraction, the less the metal will flow into the die resulting in more stretching and thinning.

   b) Boundary conditions may also be set using the "Auto Beads" feature (see the AUTOBD field) with which draw bead forces are automatically applied to all nodes along the part boundary. The users must specify the fraction of the fully locked bead force to be applied. The default value of 30% is sufficient for crash/occupant safety applications.

   The last important, but often overlooked, "boundary condition" is the part's shape. For example, an oil pan with a larger flange area will experience greater thinning in the part wall, whereas having a smaller flange area will have the reverse effect. To obtain results that are closer to the incremental strain theory, additional materials may need to be added to the final part geometry in cases where the sheet blank is not "fully developed," meaning no trimming is required to finish the part.

7. **Friction.** Friction effects can be included with the FRICTION option. The frictional force is based on an expected binder tonnage, and is a percentage of the input force. Note that the binder tonnage value (see BDTON) is used *exclusively* in calculating friction forces. The binder tonnage is not actually applied on the binder as a boundary condition.

8. **Rigid Body Motion.** LS-DYNA will automatically add nodal constraints to prevent rigid body motion when the AUTO_CONSTRAINT option is used and ICON is set to 1.

9. **Implicit Solver Options.** All other implicit cards, such as *CONTROL_IMPLICIT_GENERAL, *CONTROL_IMPLICIT_SOLUTION, *CONTROL_IMPLICIT_SOLVER, *CONTROL_IMPLICIT_AUTO, *CONTROL_IMPLICIT_-TERMINATION, etc., are used to set the convergence tolerance, termination criterion, etc. The two most important variables controlling the solution con-

vergence are DELTAU from *CONTROL_IMPLICIT_TERMINATION, and DCTOL from *CONTROL_IMPLICIT_SOULTION. Experience has shown that they should be set to 0.001 and 0.01, respectively, to obtain the most efficient solution with the best results. Typically, four implicit steps are sufficient, and DT0 in *CONTROL_IMPLICIT_GENERAL and ENDTIM in *CONTROL_TERMINATION should be set accordingly. For difficult parts, more steps maybe needed. For some parts, ILIMIT in *CONTROL_IMPLICIT_SOLUTION may need to be set to "1" for the full Newton iteration.

10. **Blank Card.** Card #2 for no option <BLANK> is a blank card, but it *must* be present.

Output:

Results are stored in an ASCII file named "onestepresult" using the dynain format. This file contains the forming thickness, the stress and the strain fields on the final part. It can be plotted with *LS-PrePost*. One quick and useful *LS-PrePost* plotting feature is the "formability contour map", which colors the model to highlight various forming characteristics including cracks, severe thinning, wrinkles, and good surfaces. The formability map feature is located in *Post → FLD → Formability*.

Additionally, the final estimated blank size in its initial, flat state is stored in the d3plot files. The d3plot files also contain intermediate shapes from each implicit step. The final blank *mesh* in its flat state can be written to a keyword file using *LS-PrePost* by the following steps:

1. Go to *Post → Output → Keyword*,

2. check the box to include *Element* and *Nodal Coordinates*

3. move the animation bar to the last state, and,

4. click on *Curr* and *Write.*

In addition, blank outlines can be created by:

1. menu option *Curve → Spline → From Mesh (Method)*,

2. checking *Piecewise → byPart*,

3. select the blank,

4. click on *Apply*, and,

5. finally, save the curves in IGES format using the *File* menu at the upper left corner.

Effect of TSCLMAX, TSCLMIN and EPSMAX:

During the early stage of product design, the initial product specifications may lead to large strains and excessive thinning on the formed panel. The ensuing one-step results would not be suitable to be used in a crashworthiness simulation. However, these kinds of forming issues are certain to be fixed as a natural part of the design and stamping engineering process. The variables TSCLMAX, TSCLMIN and EPSMAX are thus created to impose artificial limits on the thinning and plastic strains. The variables provide convenient way to run a crash simulation with approximate and reasonable forming effects before the design is finalized. In the keyword below (which is a part of the firewall model with original thickness of 0.75mm), TSCLMIN and EPSMAX are set to 0.8 and 0.3, respectively.

```
*CONTROL_FORMING_ONESTEP
$   OPTION   TSCLMAX    AUTODB   TSCLMIN    EPSMAX
         7                 0.5       0.8       0.3
```

The thickness and effective plastic strain plots for the firewall model are shown in Figures 12-29 and 12-30, respectively. The minimum value in the thickness contour plot and maximum value in the plastic strain contour plot as shown in the upper left corner correspond to the values specified in TSCLMIN and EPSMAX, respectively.

Similarly, TSCLMAX can be set to 1.0667 to limit the max thickening in the part to 0.8mm:

```
*CONTROL_FORMING_ONESTEP
$   OPTION   TSCLMAX    AUTODB   TSCLMIN    EPSMAX
         7    1.0667       0.5       0.8       0.3
```

Reposition of unfolded flat blank:

Often times the input to one-step simulation is the final product part in the car axis system. However, after the simulation, the unfolded flat blank will be in a different orientation and position, requiring users to manually reposition the blank to its desired orientation and position. The variables NODE1, NODE2, NODE3 allow users to specify three nodes so that the blank is transformed onto the final part (the input), superimposing the exact same three nodes in both parts. In an example shown in Figure 12-32, the three nodes (Nodes 197, 210 and 171) are defined near the edges of two holes. The transformed and unfolded flat blank (written in a keyword file "repositioned.k") is seen superimposed onto the final part according to the three nodes specified (Figure 12-32 bottom). If these nodes are not defined, the simulation will result in the unfolded flat blank in a state shown in Figure 12-32 (top), undesirable to most users.

Damage accumulation:

Damage accumulation D is calculated based on (refer to manual section *MAT_ADD_EROSION):

$$D = \left(\frac{\varepsilon_p}{\varepsilon_f}\right)^{\text{DMGEXP}}$$

In the example below, load curve #500 provides plastic failure strain vs. stress triaxiality and DMGEXP is assumed to be 1.254. Since the damage accumulation is written into the file onestepresult as history variable #6, the variable NEIP in *DATABASE_EXTENT_BINARY should be set to at least '6".

```
*CONTROL_FORMING_ONESTEP
$   OPTION              AUTODB   TSCLMIN    EPSMAX                LCSDG    DMGEXP
         7                           0.8       0.3                  500     1.254
*DEFINE_CURVE
500
-0.3,0.6
-0.2,0.3
0.0,0.2
0.2,0.25
0.4,0.46
0.65,0.28
0.9,0.18
*DATABASE_EXTENT_BINARY
$    NEIPH     NEIPS    MAXINT    STRFLG    SIGFLG    EPSFLG    RLTFLG    ENGFLG
                   6         7         1
$   CMPFLG    IEVERP    BEAMIP     DCOMP      SHGE     STSSZ
                   1                   2
```

The damage accumulation contour map from the file onestepresult can be plotted in LS-PrePost.

Effect of hole-cut on the forming results:

In Figure 12-31, a thickness contour plot of a one-step calculation on the NCAC Taurus firewall model with its holes unfilled is shown. The unfilled case will undergo slightly less thinning, since the holes will expand as material flows outward away from the hole. However, the thicknesses with holes filled are likely closer to reality, since the holes are mostly filled during forming on the draw panel and then trimmed off afterwards in a trim process. On the other hand, it is important to realize that not all the holes are filled in a draw panel. Some holes are cut inside the part in the scrap area (but not all the way to the trim line) during draw process to allow material to flow into areas that are difficult to form, so as to avoid splitting.

Application example:

The following example provides a partial input file with typical control cards. It will iterate for four steps, with auto beads of 0% lock force applied around the part boundary, and with automatic nodal constraints.

```
*CONTROL_TERMINATION
$   ENDTIM
       1.0
*CONTROL_IMPLICIT_GENERAL
$   IMFLAG       DT0
         1      0.25
*CONTROL_FORMING_ONESTEP
$   OPTION              AUTODB
         7

*CONTROL_FORMING_ONESTEP_AUTO_CONSTRAINT
$     ICON
         1
*CONTROL_IMPLICIT_TERMINATION
$   DELTAU
     0.001
*CONTROL_IMPLICIT_SOLUTION
$  NSOLVR    ILIMIT    MAXREF     DCTOL     ECTOL
         2        11      1200      0.01      1.00
*CONTROL_IMPLICIT_SOLVER
$   LSOLVR
         5
*CONTROL_IMPLICIT_AUTO
$    IAUTO    ITEOPT    ITEWIN     DTMIN     DTMAX
         0         0         0       0.0       0.0
```

Additional cards below specify extra bead forces of 45% and 30% applied to node sets 22 and 23 along the part periphery, respectively. Also, the resulting friction forces with friction coefficient of 0.1 and binder tonnage of 10000.0 N used for friction force are applied on the same node sets.

```
*CONTROL_FORMING_ONESTEP_DRAWBEAD
$   NDSET      LCID        TH    PERCNT
       22       200       1.6      0.45
*CONTROL_FORMING_ONESTEP_DRAWBEAD
       23       200       1.6      0.30
*CONTROL_FORMING_ONESTEP_FRICTION
$   NDSET     BDTON     FRICT
       22   10000.0       0.1
*CONTROL_FORMING_ONESTEP_FRICTION
$   NDSET     BDTON     FRICT
       23   10000.0       0.1
```

The one-step forming results for the NCAC Taurus model's firewall are shown in Figure 12-25. The average element size across the blank is 8mm, and the trimmed part (with holes filled) consists of 15490 elements. *MAT_24 was used with BH210 material properties. On a 1 CPU Xeon E5520 Linux machine, it took 4 minutes to complete the run with a total of four steps. The thickness, the plastic strain, and the blank size prediction were reasonable, as shown in Figures 12-26, 12-27 and 12-28.

Performance among options TRIA, QUAD and QUAD2:

The following partial keyword input is an example of using the option QUAD. Note the draw bead force parameter AUTOBD is set at 0.5. Calculation speed comparison among options QUAD, QUAD2 and TRIA can be found in Table 12-1.

```
*KEYWORD
*include
model.k
*CONTROL_TERMINATION
1.0
*CONTROL_FORMING_ONESTEP_QUAD
$#  option  maxthick    autobd   thinmin    epsmax
         7                 0.5

*CONTROL_FORMING_ONESTEP_AUTO_CONSTRAINT
         1
*CONTROL_IMPLICIT_GENERAL
$#  imflag       dt0    imform      nsbs       igs     cnstn      form    zero_v
         1    0.2500         2         1         0         0         0         0
*CONTROL_IMPLICIT_TERMINATION
$#  deltau    delta1     ketol     ietol     tetol     nstep
  0.001000     0.000     0.000     0.000     0.000         0
*CONTROL_IMPLICIT_NONLINEAR
$#  nsolvr    ilimit    maxref     dctol     ectol  not used     lstol      rssf
        12        11       200  0.010000  0.100000     0.000     0.000     0.000
$#   dnorm    diverg     istif   nlprint
         0         0         0         2
$#  arcctl    arcdir    arclen    arcmth    arcdmp
         0         0     0.000         1         2
*CONTROL_IMPLICIT_SOLVER
5
*PART

   5000000   5000000   5000000
*SECTION_SHELL
   5000000        16        1.        5.        1.
      0.72      0.72      0.72      0.72
...
```

Table 12-1 Calculation speed improvement with and without option _QUAD.

	Number of elements	Calculation speed (D.P. SMP Rev.112720, 8 CPUs)		
		Option TRIA	Option QUAD	Option QUAD2
A hat shape part	71000	21.0 min	14.1 min	16.6 min
A upper dash panel	61700	24.5 min	11.5 min	17.2 min

Revision information:

This feature is available starting in Revision 67778 SMP and double precision only. Over time, improvements are made to improve accuracy and speed up the calculation time; revision 110117 or later is recommended. Historic revisions are listed as follows:

1)Revision 73442: output of stress tensors
2)Revision 75156: output of strain tensors.
3)Revision 75854: variables THINPCT and EPXMAX are available.
4)Revision 76709: holes are allowed.
5)Revision 91229: variables NODE1, NODE2, NODE3 are available.
6)Revision 108229: variables LCSDG and DMGEXP are available.
7)Revision 111311: variable TSCLMAX is available.
8)Revision 112071: option QUAD is available.
9)Revision 112682: original formulation is designated as option TRIA. A new option QUAD2 is activated.
10)Revision 109680: negative value of OPTION for a large format stress and strain output.

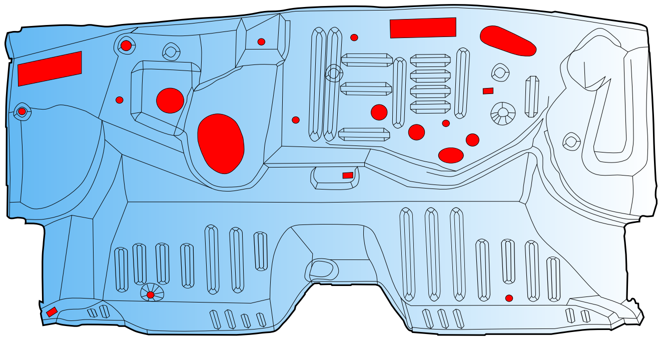


Figure 12-25. A trimmed dash panel (firewall) with holes auto-filled using LS-PrePost 4.0 (original model courtesy of NCAC Taurus crash model).

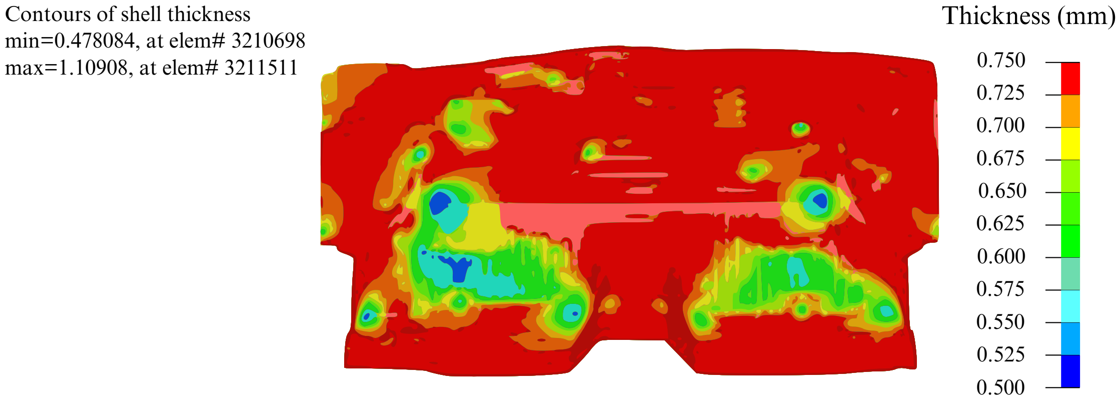


Figure 12-26. Shell thickness prediction (t_0 = 0.75mm).

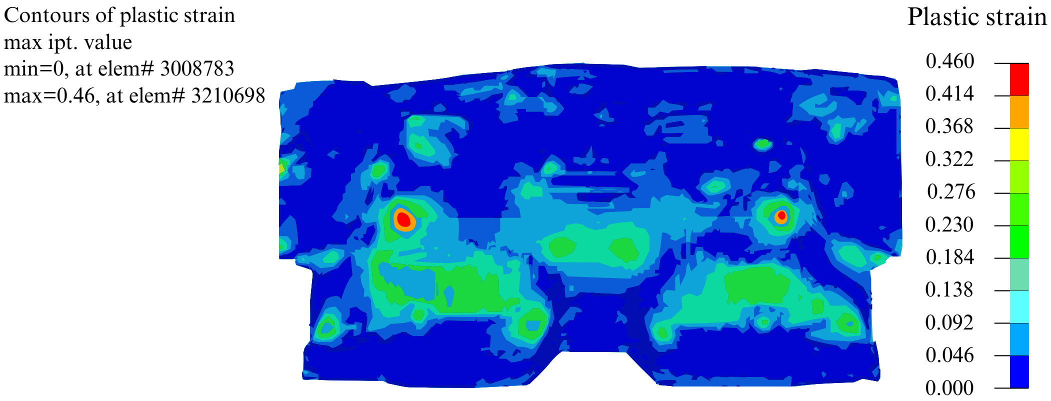


Figure 12-27. Effective plastic strain Prediction.

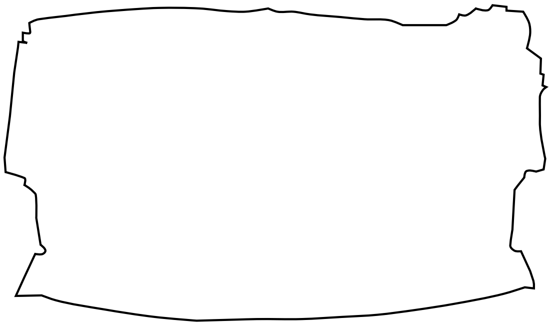


Figure 12-28. Initial blank size prediction (flat, not to scale).

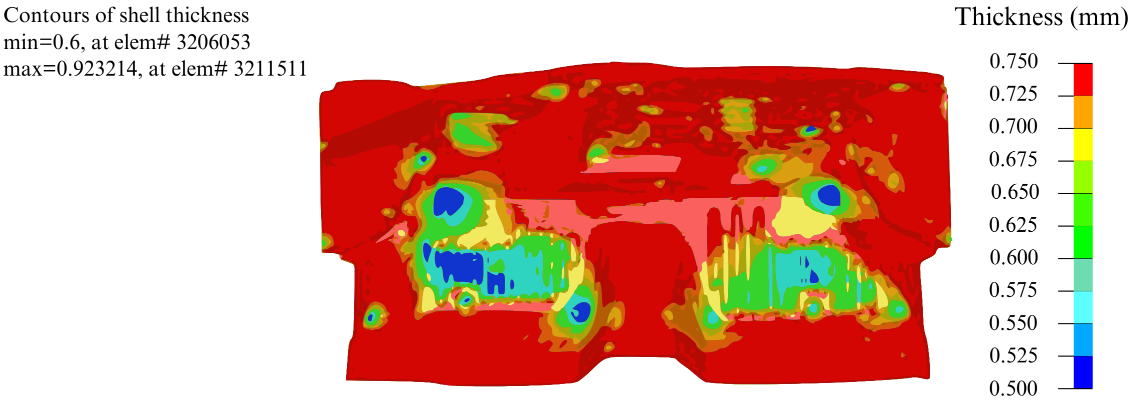


Figure 12-29. Blank thickness prediction with TSCLMIN = 0.8.

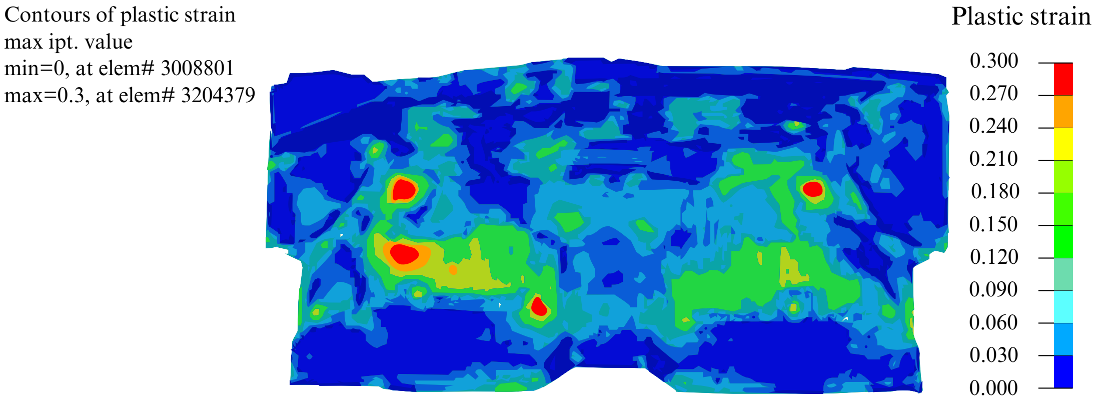


Figure 12-30. Effective plastic strain with EPSMAX = 0.3.

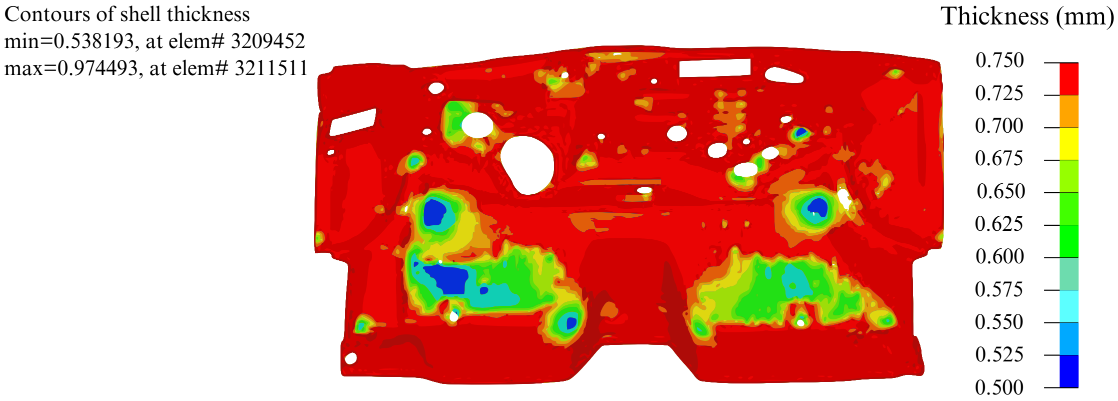


Figure 12-31. Blank thickness with trimmed holes ($t_0 = 0.75$mm).

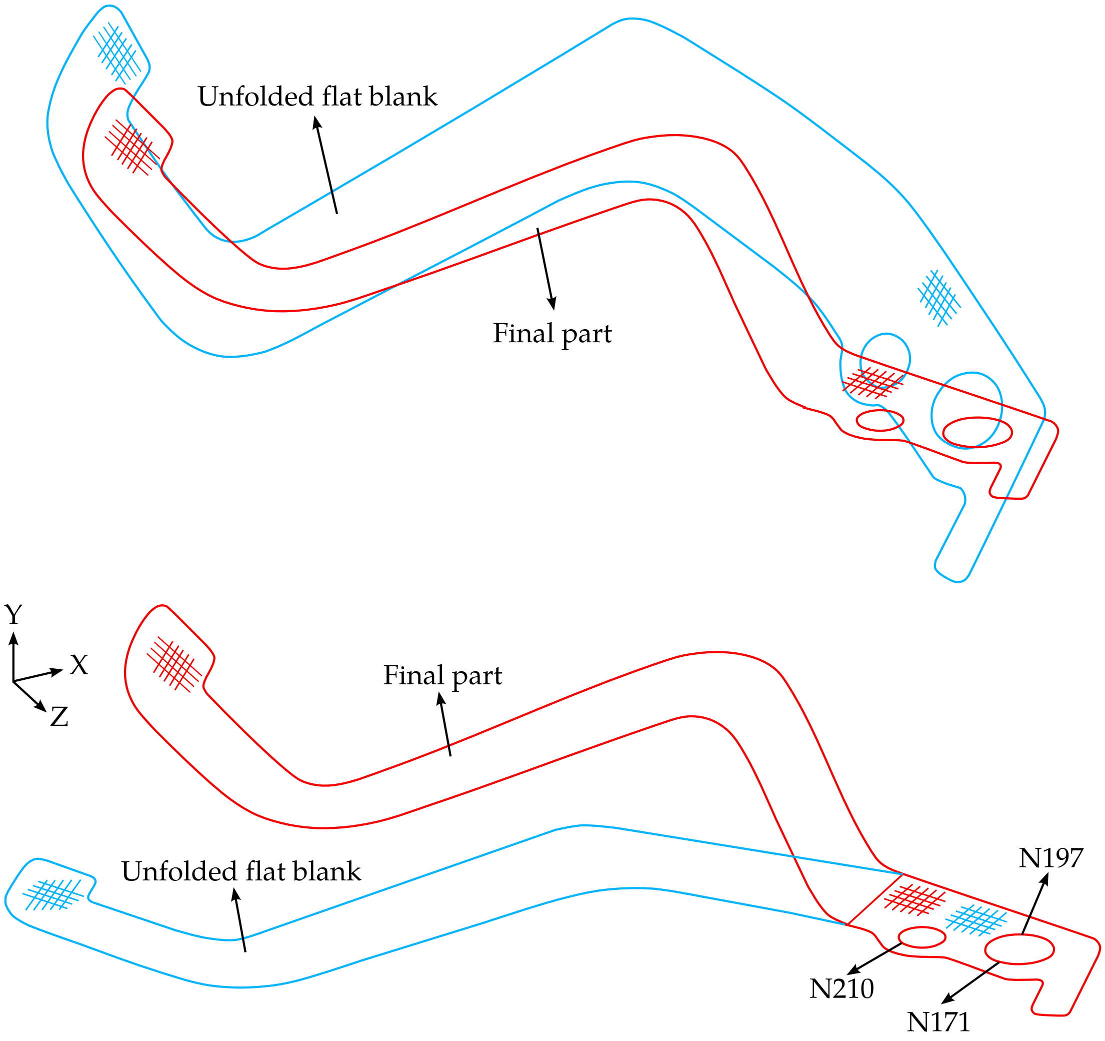


Figure 12-32. An example of the results when using the NODE1, NODE2, and NODE3 feature (bottom) and without using the feature (top), courtesy of Kaizenet Technologies Pvt Ltd, India.

CONTROL_FORMING_OUTPUT_*{OPTION}

Available options include:

<BLANK>

INTFOR

Purpose: This card defines the times at which states are written to the d3plot and intfor files based on the tooling's distances from the home (final) position. When the INTFOR option is set this keyword card controls when states are written to the intfor file, otherwise it controls the d3plot file. This feature may be combined with parameterized input and/or automatic positioning of the stamping tools using the *CONTROL_FORMING_AUTOPOSITION_PARAMETER card.

> NOTE: When this card is present no states are written except for those specified on this card. This card supersedes the *DATABASE_BINARY_D3PLOT card.

Forming Output Cards. Repeat as many times as needed to define additional outputs in separate tooling kinematics curves. The next keyword ("*") card terminates the input.

Card 1	1	2	3	4	5	6	7	8
Variable	CID	NOUT	TBEG	TEND	Y1/LCID	Y2/CIDT	Y3	Y4
Type	I	I	F	F	F/I	F/I	F	F
Default	none	0	0.0	none	none	none	none	none

VARIABLE	DESCRIPTION
CID	ID of a tooling kinematics curve. This curve is integrated so that the specified output distances can be mapped to times. For correct distance-to-time mapping CID must be applied to the tool of interest using a *BOUNDARY_PRESCRIBED_MOTION_RIGID card. The ordinate scale factor SFO in the *DEFINE_CURVE is supported in this keyword starting from Revision 82755.

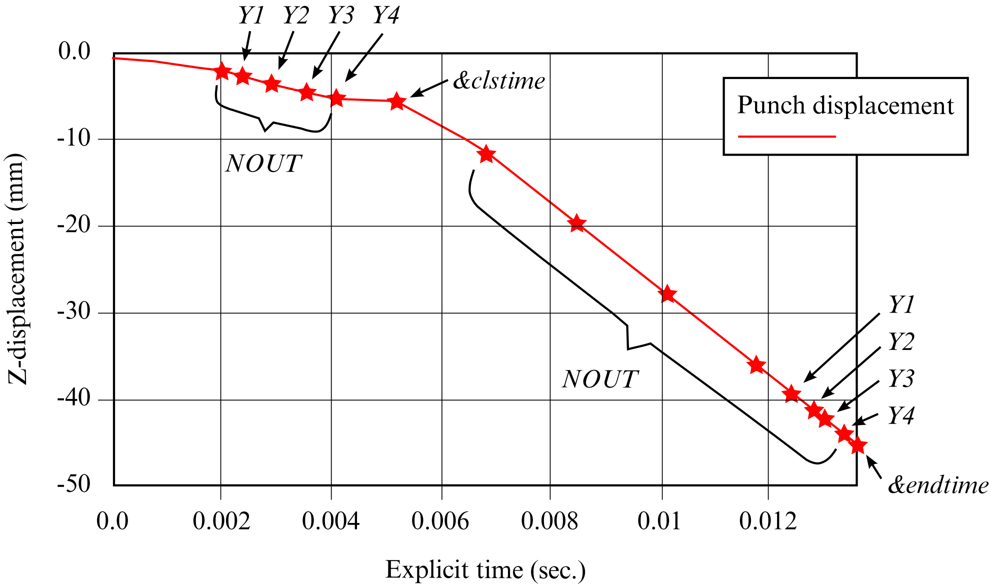


Figure 12-33. An output example for closing and drawing. See the example provided at the end of this section.

VARIABLE	DESCRIPTION
NOUT	Total number states written to the d3plot or intfor databases for the tooling kinematics curve, CID, excluding the beginning and final states. If NOUT is larger than the number of states specified by either LCID or Y*i* fields (5 through 8), the remaining states are evenly distributed between TBEG and the time corresponding to the biggest Y*i* from the home position, as shown in Figure 12-33. If NOUT is left as blank or as “0”, the total number of output states will be determined by either LCID or Y*i*’s.
TBEG	Start time of the curve. This time should be consistent with the BIRTH in *BOUNDARY_PRESCRIBED_MOTION_RIGID.
TEND	End time of the curve. This time should be consistent with the DEATH in *BOUNDARY_PRESCRIBED_MOTION_RIGID. This time is automatically reset backward removing any idling time if the tool finishes traveling early, so output distances can start from the reset time. A state is written at TEND.

VARIABLE	DESCRIPTION
Y1/LCID, Y2, Y3, Y4	Y1/LCID.GT.0: All four variables (Y1, Y2, Y3, Y4) are taken to be the distances from the punch home, where d3plot files will be output. Y1/LCID.LT.0: The absolute value of Y1/LCID (must be an integer) is taken as a load curve ID (see *DEFINE_CURVE). Only the abscissas in the load curve, which are the distances to punch home, are used. These distances specify the states that are written to the d3plot files. Ordinates of the curve are ignored. This case accommodates more states than is possible with the four variables Y1, Y2, Y3, Y4. Furthermore, when Y1/LCID < 0, Y2, Y3, and Y4 are ignored. Available starting from Revision 112604, the output will be skipped for any negative abscissa in the load curve. Note all abscissas being negative are not allowed.
Y2/CIDT	Y2/CIDT.GT.0: The input is taken as the distance from the punch home, where a d3plot file will be output. Y2/CIDT.LT.0: The absolute value of Y2/CIDT (must be an integer) is taken as a load curve ID (see *DEFINE_CURVE). Only the abscissas in the load curve, which are the simulation times, are used. These times specify the states that are written to the d3plot files. Ordinates of the curve are ignored. Note this time-dependent load curve will output additional d3plot files on top of the d3plot files already written in case Y1/LCID < 0 (if specified). Furthermore, when Y2/CIDT < 0, Y3 and Y4 are ignored. See an example for usage.

Motivation:

In stamping simulations not all time steps are of equal interest to the analyst. This feature allows the user to save special states, usually those for which wrinkling and thinning conditions arise as the punch approaches its home position.

This feature is available in Application eZ-Setup in LS-PrePost4.0 (http://ftp.lstc.com/anonymous/outgoing/lsprepost/4.0/metalforming/).

Remarks:

1. Keywords *DATABASE_BINARY_D3PLOT and *DATABASE_BINARY_INTFOR are not required (ignored if present) to output D3PLOT and INTFOR files when this keyword is present;

2. *CONTROL_FORMING_OUTPUT and *CONTROL_FORMING_OUTPUT_INTFOR can share the same CIDs;

3. If columns 5 through 8 are left blank, output (NOUT) will be evenly distributed through the travel;

4. The variable NOUT has priority over the number of points on the LCID;

5. Distances input (in LCID) that are greater than the actual tool travel will be ignored;

6. Distance input (in LCID) does not necessarily have to be in a descending or ascending order.

Applicability:

This keyword is applicable to the parameter VAD of "0" (velocity) in *BOUNDARY_PRESCRIBED_MOTION_RIGID, and for explicit dynamics only. Tooling kinematics profiles of various trapezoids (including right trapezoid) are all supported. Local coordinate systems are supported.

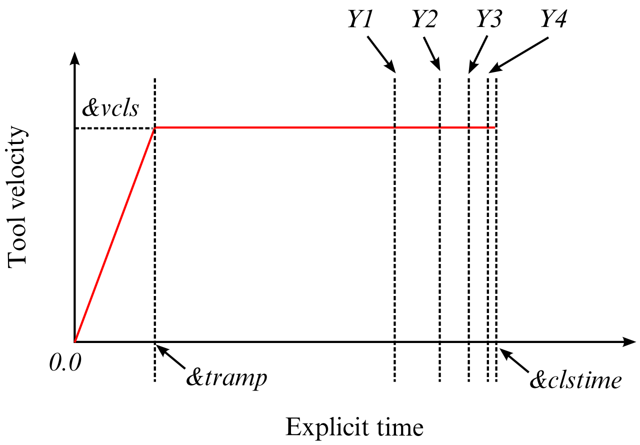


Figure 12-34. Specifying d3plot/intfor output at specific distances to punch home.

Application example for an air draw:

In a keyword example below (air draw, referring to Figures 12-33 and 12-35), a total of five states will be output during a binder closing. The kinematics are specified by the curve of ID 1113, which defines tooling kinematics starting time 0.0 and ending at time &clstime.

Curve 1113 is used to associate the specified distances to the appropriate time step. In this example NOUT is set to 5. Of these five outputs states the last four will be output at upper die distance to closing of 3.0, 2.0, 1.0, and 0.5 mm according to the values specified in the Y1, Y2, Y3, and Y4 fields.

Similarly, a total of eight states will be written to the d3plot file made during draw forming according curve ID 1115, which defines tooling kinematics starting at time *&clstime*, and ending at time *&endtime*. Of the eight states the last four will be output at punch distance to draw home of 6.0, 4.0, 3.0, and 1.0 mm; *the remaining four outputs will be evenly distributed between starting punch distance to home and punch distance of 6.0mm to home.*

Likewise, for intfor, 15 states will be written before closing and 18 states after the closing. The d3plot and intfor files will always be output for the first and last states as a default; and at where the two curves meet at &clstime, only one d3plot and intfor will be output.

To output intfor, "S=*filename*" needs to be specified on the command line, and SPR and MPR need to be set to "1" on the *CONTACT_… cards.

```
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*CONTROL_FORMING_OUTPUT
$       CID        NOUT        TBEG       TEND          y1          y2          y3          y4
```

```
      1113         5            &clstime       3.0       2.0       1.0       0.5
      1115         8  &clstime  &endtime       6.0       4.0       3.0       1.0
*CONTROL_FORMING_OUTPUT_INTFOR
$      CID      NOUT      TBEG      TEND        y1        y2        y3        y4
      1113        15            &clstime       3.5       2.1       1.3       0.7
      1115        18  &clstime  &endtime      16.0       4.4       2.1       1.3
*BOUNDARY_PRESCRIBED_MOTION_RIGID
$   typeID       DOF       VAD      LCID        SF       VID     DEATH     BIRTH
&udiepid           3         0      1113      -1.0         0  &clstime       0.0
&bindpid           3         0      1114       1.0         0  &clstime       0.0
&udiepid           3         0      1115      -1.0         0  &endtime  &clstime
&bindpid           3         0      1115      -1.0         0  &endtime  &clstime
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*DEFINE_CURVE
1113
0.0,0.0
&clsramp,&vcls
&clstime,0.0
*DEFINE_CURVE
1114
0.0,0.0
10.0,0.0
*DEFINE_CURVE
1115
0.0,0.0
&drwramp,&vdraw
&drwtime,&vdraw
```

The keyword example below illustrates the use of load curves 3213 and 3124 to specify the states written to the d3plot and intfor files respectively. In addition to the eight states specified by curve 3213, five additional outputs will be generated. Similarly, in addition to the 10 intfor states defined by curve 3214, eight additional states will be output.

```
*CONTROL_FORMING_OUTPUT
$      CID      NOUT      TBEG      TEND        y1        y2        y3        y4
      1113        13            &clstime     -3213
*CONTROL_FORMING_OUTPUT_INTFOR
$      CID      NOUT      TBEG      TEND        y1        y2        y3        y4
      1113        18            &clstime     -3214
*DEFINE_CURVE
3213
88.0
63.0
42.0
21.5
9.8
5.2
3.1
1.0
*DEFINE_CURVE
3214
74.0
68.0
53.0
32.0
25.5
7.8
4.2
2.1
1.4
0.7
```

Application example for a multiple flanging process:

Referring to Figure 12-36 and a partial keyword example listed below, flanging steels #1 through #4 are defined as parameters &flg1pid through &flg4pid, respectively, which are moving in their own local coordinate systems. The termination time &endtime is defined as pad closing time &clstime plus the maximum travel time of all four flanging steels. A total of ten d3plot states and ten intfor states are defined for each flanging steel using curve IDs 980 and 981, respectively. Curve values outside of the last 10 states (distances) are ignored; and reversed points are automatically adjusted.

In Figure 12-37, locations of d3plot states are indicated by "x" markers for each flanging steel move. Note that for flanging steels with longer travel distances, there may be additional d3plot states between the defined points, controlled by distance output defined for other flanging steels with shorter travels. The total number of d3plot (and intfor) states is the sum of all nout defined for each flanging steel so care should be taken to limit the total d3plot (and intfor) states, especially if large number of flanging steels are present.

```
*KEYWORD
$  -------------------------closing
*BOUNDARY_PRESCRIBED_MOTION_RIGID
$   typeID       DOF       VAD      LCID        SF       VID     DEATH     BIRTH
    &upid1         3         0      1113  &padvdir         0  &clstime
*BOUNDARY_PRESCRIBED_MOTION_RIGID_local
  &flg1pid         3         0      1114       1.0         0  &clstime
  &flg2pid         3         0      1114       1.0         0  &clstime
  &flg3pid         3         0      1114       1.0         0  &clstime
  &flg4pid         3         0      1114       1.0         0  &clstime
$  -------------------------flanging
*BOUNDARY_PRESCRIBED_MOTION_RIGID
$   typeID       DOF       VAD      LCID        SF       VID     DEATH     BIRTH
    &upid1         3         0      1115  &padvdir         0            &clstime
*BOUNDARY_PRESCRIBED_MOTION_RIGID_local
  &flg1pid         3         0      1116       1.0         0            &clstime
  &flg2pid         3         0      1117       1.0         0            &clstime
  &flg3pid         3         0      1118       1.0         0            &clstime
  &flg4pid         3         0      1119       1.0         0            &clstime
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*DEFINE_CURVE
      1116
                 0.0                 0.0
             &tdrwup               &vdrw
             &tdown1               &vdrw
            &drw1tim                 0.0
             1.0E+20                 0.0
*DEFINE_CURVE
      1117
                 0.0                 0.0
             &tdrwup               &vdrw
             &tdown2               &vdrw
            &drw2tim                 0.0
             1.0E+20                 0.0
*DEFINE_CURVE
      1118
                 0.0                 0.0
             &tdrwup               &vdrw
             &tdown3               &vdrw
            &drw3tim                 0.0
```

```
             1.0E+20                 0.0
*DEFINE_CURVE
      1119
                 0.0                 0.0
             &tdrwup               &vdrw
             &tdown4               &vdrw
            &drw4tim                 0.0
             1.0E+20                 0.0
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*DEFINE_CURVE
980
60.0
55.0
42.0
40.0
38.0
31.0
23.0
19.0
15.0
13.0
13.5
5.0
3.0
2.0
2.5
1.0
*DEFINE_CURVE
981
23.0
19.0
15.0
13.0
13.5
  ⋮
*CONTROL_FORMING_OUTPUT
$ -------1---------2---------3---------4---------5---------6---------7---------8
$      CID      NOUT      TBEG      TEND   Y1/LCID
      1116        10  &clstime  &endtime      -980
      1117        10  &clstime  &endtime      -980
      1118        10  &clstime  &endtime      -980
      1119        10  &clstime  &endtime      -980
*CONTROL_FORMING_OUTPUT_INTFOR
$ -------1---------2---------3---------4---------5---------6---------7---------8
$      CID      NOUT      TBEG      TEND   Y1/LCID
      1116        10  &clstime  &endtime      -981
      1117        10  &clstime  &endtime      -981
      1118        10  &clstime  &endtime      -981
      1119        10  &clstime  &endtime      -981
        ⋮          ⋮      ⋮         ⋮           ⋮
```

An example of using CIDT:

The example below shows in addition to the 7 states output based on various distances from punch home, defined by load curve 980, 4 more states are output based on simulation time, defined by load curve 999.

```
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*DEFINE_CURVE
999
1.0e-03
2.0e-03
```

```
3.0e-03
4.0e-03
*DEFINE_CURVE
980
13.5,0.0
13.0,0.0
5.0,0.0
3.0,0.0
2.5,0.0
2.0,0.0
1.0,0.0
*CONTROL_FORMING_OUTPUT
$ -------1---------2---------3---------4---------5---------6---------7---------8
$      CID      NOUT      TBEG      TEND   Y1/LCID   Y2/CIDT
      1116         0  &clstime  &endtime      -980      -999
      1117         0  &clstime  &endtime      -980      -999
      1118         0  &clstime  &endtime      -980      -999
      1119         0  &clstime  &endtime      -980      -999
```

Revision information:

This feature is available starting from LS-DYNA Revision 74957. Other options' availabilities are as follows:

7. Y1/LCID is available from Revision 81403.

8. The scale factor SFO for ordinate values in *DEFINE_CURVE is supported from Revision 82755.

9. Output for multiple tools is available from Revision 83090.

10. Support for arbitrary BIRTH and DEATH in *BOUNDARY_PRESCRIBED_MOTION_RIGID is available from Revision 83090.

11. The INTFOR option is available from Revision 83757.

12. Y2/CIDT is available from Revision 110091.

13. Negative abscissa in the LCID is available starting from Revision 112604.

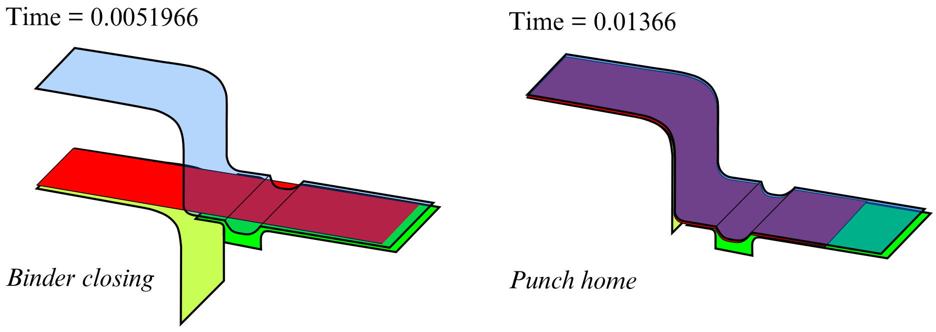


Figure 12-35. An air draw example with closing and drawing.

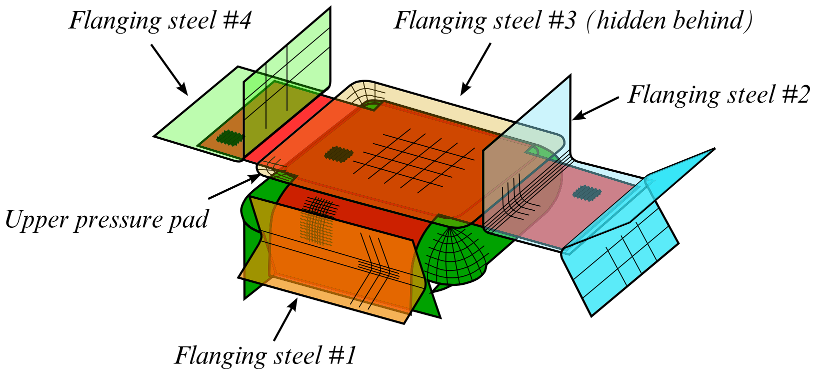


Figure 12-36. An example of multiple flanging process.

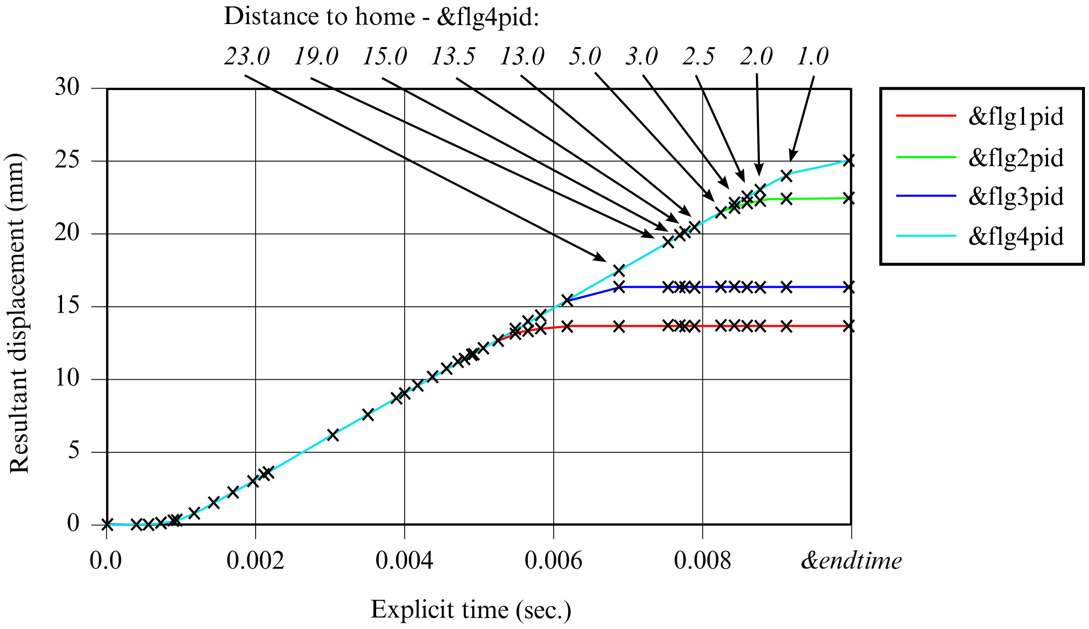


Figure 12-37. D3PLOT/INTFOR output in case of multiple flanging process.

*CONTROL_FORMING_PARAMETER_READ

Purpose: This feature allows for reading of a numerical number from an existing file and store in a defined parameter. The parameter can be used and referred in the current simulation. The file to be read may be a result from a previous simulation. The file may also simply contain a list of numbers defined beforehand and to be used for the current simulation.

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	C							

Parameter Cards. Include one card for each parameter. The next "*" card terminates the input.

Card 2	1	2	3	4	5	6	7	8
Variable	PARNAME	METHOD	LINE #	BEGIN	END			
Type	C	I	I	I	I			
Default	none	0	0	0	0			

VARIABLE	DESCRIPTION
FILENAME	Name of the file to be read.
PARNAME	Parameter name. Maximum character length: 7.
METHOD	Read instruction: EQ.1: read, follow definition by LINE#, BEGIN and END definition
LINE #	Line number in the file.
BEGIN	Beginning column number in the line number defined above.
END	Ending column number in the line number defined above.

Remarks:

1. Keyword input order is sensitive. Recommended order is to define variables in *PARAMETER first, followed with this keyword, using the defined variables.

2. Multiple variables can be defined with one such keyword, with the file name needed to be defined only once. If there are variables located in multiple files, the keyword needs to be repeated for each file.

3. An example provided below shows that multiple PIDs for individual tools and blank are defined in files “data.k” and “data1.k”. In the main input file “sim.dyn” used for LS-DYNA execution, variables (integer) are first initialized for PIDS of all tools and blank with *PARAMETER. These variables are updated with integers read from files “data.k” and “data1.k” from respective line number and column number through the use of this keyword. In the *SET_PART_LIST definition, these PIDs are used to define the part set.

   Below is file “data.k”, to be read into “sim.dyn::

```
$$$$$$$$$$$$$$$$$$$$$$$$$
$$$ define PIDs
$$$$$$$$$$$$$$$$$$$$$$$$$
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+--
upper die  pid:              3
lower post pid:              2
Below is file “data1.k”, also to be read into “sim.dyn”:
$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$
$$$ define PIDs
$$$$$$$$$$$$$$$$$$$$$$$$$
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+--
lower binder pid:            4
blank pid:                   1
```

   Below is partial input for the main input file “sim.dyn”:

```
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+--
*INCLUDE
blank.k
*INCLUDE
tool.k
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+--
*PARAMETER
Iblankp,0
Iupdiep,0
Ipunchp,0
Ilbindp,0
Rblankmv,0.0
Rpunchmv,0.0
Rupdiemv,0.0
Rbindmv,0.0
Rbthick,1.6
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+--
*CONTROL_FORMING_PARAMETER_READ
data.k
updiep,1,5,30,30
punchp,1,6,30,30
*CONTROL_FORMING_PARAMETER_READ
```

```
data1.k
lbindp,1,7,30,30
blankp,1,8,30,30
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+--
*SET_PART_LIST
1
&blankp
*SET_PART_LIST
2
&punchp
*SET_PART_LIST
3
&updiep
*SET_PART_LIST
4
&lbindp
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+--
*CONTROL_FORMING_AUTOPOSITION_PARAMETER_SET
$#    psid       cid       dir     mpsid  position   premove     thick parname
         1         0         3         2         1     0.000   &bthick blankmv
         3         0         3         1         1     0.000           updiemv
         4         0         3         1        -1     0.000            bindmv
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+--
*PART_MOVE
$pid,xmov,ymov,zmov,cid,ifset
1,0.0,0.0,&blankmv,,1
3,0.0,0.0,&updiemv,,1
4,0.0,0.0,&bindmv,,1
```

4. This feature is available in LS-DYNA R5 Revision 55035 and later releases.

*CONTROL_FORMING_POSITION

Purpose: This keyword allows user to position tools and a blank in setting up a stamping process simulation. All tools must be pre-positioned at their home positions. For tools that are positioned above the sheet blank (or below the blank) and ready for forming, *CONTROL_FORMING_TRAVEL should be used. This keyword is used together with *CONTROL_FORMING_USER. One *CONTROL_FORMING_POSITION card may be needed for each part.

> **NOTE:** This option has been deprecated in favor of *CONTROL_FORMING_AUTOPOSITION_PARAMETER).

Positioning Cards. For each part to be positioned include an additional card. The next "*" card terminates the input.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	PREMOVE	TARGET					
Type	I	F	I					
Default	none	none	I					

VARIABLE	DESCRIPTION
PID	Part ID of a tool to be moved, as in *PART
PREMOV	The distance to pre-move the tool, in the reverse direction of forming.
TARGET	Target tool PID, as in *PART. The tool (PID) will be moved in the reverse direction of the forming and positioned to clear the interference with the blank, then traveled to its home position with a distance GAP (*CONTROL_FORMING_USER) away from the TARGET tool to complete the forming.

Remarks:

When this keyword is used, all stamping tools must be in their respective home positions, which is also the position of each tool at its maximum stroke. From the home position each tool will be moved to its start position, clearing interference between the blank and tool yet maintaining the minimum separation needed to avoid initial penetration. Currently the tools can only be moved and travels in the direction of the global Z-axis.

A partial keyword example is provided in manual pages under *CONTROL_FORMING_USER.

Revision information:

This feature is available starting in Revision 24641.

*CONTROL_FORMING_PRE_BENDING

Purpose: This keyword allows for a pre-bending of an initially flat sheet metal blank, typically used in controlling its gravity loaded shape during sheet metal forming.

Card 1	1	2	3	4	5	6	7	8
Variable	PSET	RADIUS	VX	VY	VZ	XC	YC	ZC
Type	I	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
PSET	Part set ID to be included in the pre-bending.
RADIUS	Radius of the pre-bending. GT.0.0: bending center is on the same side as the element normals LT.0.0: bending center is on the reverse side of the element normals. See figure below for more information.
VX, VY, VZ	Vector components of an axis about which the flat blank will be bent.
XC, YC, ZC	X, Y, Z coordinates of the center of most-bent location. If undefined, center of gravity of the blank will be used as a default.

About pre-bending for gravity:

In some situation, a flat blank upon gravity loading will result in a “concave” shape in a die. This mostly happens in cases where there is little or no punch support in the middle of the die cavity and in large stamping dies. Although the gravity loaded blank shape is correct the end result is undesirable. In these conditions, buckles may result during the ensuing closing and forming simulation. In reality, a true flat blank rarely exists. Typically, the blank is either manipulated (shaking or bending) by die makers in the tryout stage, or by suction cups in a stamping press, to get an initial convex shape prior to the binder closing and punch forming. This keyword allows this bending to be performed.

Application example:

A partial keyword example (*NUMISHEET2022 fender outer*) is provided below, where blank part set ID variable &BLKSID is defined previously, is to be bent in a radius value of -10000.0mm, with the bending axis of Z, located on the reverse side of the blank positive normal (Figure 12-38). The bending is off gravity center at x = 234.0, y = 161.0, z = 81.6 (to the right along positive X-axis). Only a slight pre-bending on the blank is needed to ensure a convex gravity-loaded shape.

```
*KEYWORD
⋮
*CONTROL_IMPLICIT_FORMING
1
*CONTROL_FORMING_PRE_BENDING
$     PSET    RADIUS        VX        VY        VZ        XC        YC        ZC
   &BLKSID   -10000.      0.00      0.00       1.0   234.000   161.000     81.60
...
*END
```

In Figures 12-39, initial blank shape without pre-bending is shown. Without pre-bending, the gravity loaded blank sags in the middle of the die cavity, Figure 12-40, which is likely unrealistic, and would lead to predictions of surface quality issues. With pre-bending applied, Figure 12-41, blank bends slight and in convex shape before loading. This shape results in an overall convex shape after gravity completes loading (Figure 12-42), leading to a much shorter binder closing distance, and a more realistic surface quality assessment.

Revision information:

This feature is available in double precision LS-DYNA Revision 66094 and later releases. It is also available in LS-PrePost4.0 eZ-Setup for metal forming application (http://ftp.lstc.com/anonymous/outgoing/lsprepost/4.0/metalforming/).

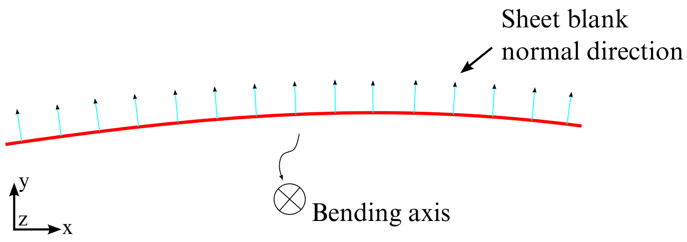


Figure 12-38. Negative "R" puts center of bending on the opposite side of the positive blank normal.

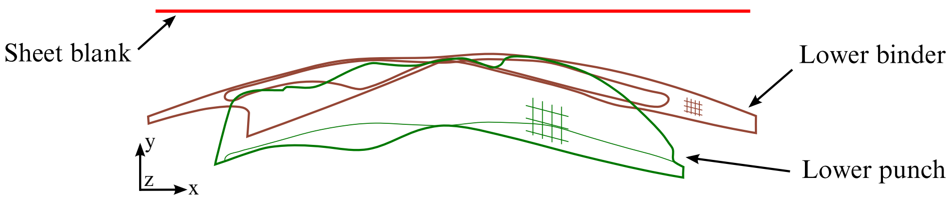


Figure 12-39. Initial model before auto-positioning.

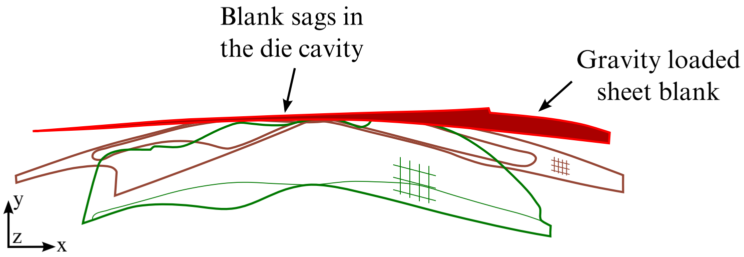


Figure 12-40. Gravity loaded blank without using this keyword.

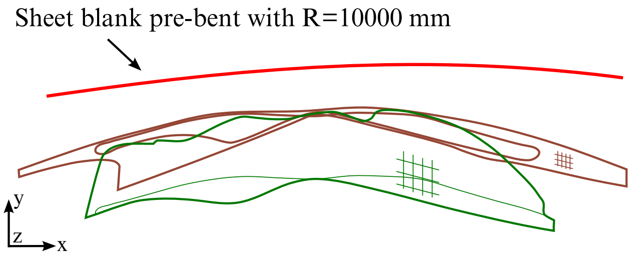


Figure 12-41. Pre-bending using this keyword (1st state of D3plots).

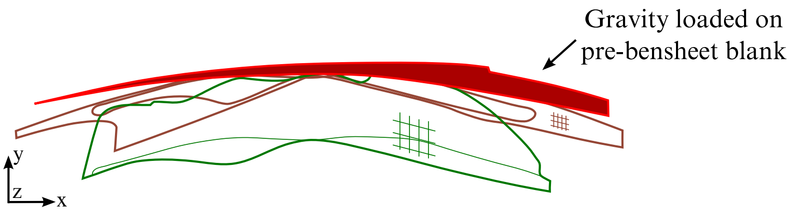


Figure 12-42. Gravity loaded shape (last state of D3plots) with convex shape.

*CONTROL_FORMING_PROJECTION

Purpose: To remove initial penetrations between the blank and the tooling (shell elements only) by projecting the penetrated blank (slave) nodes along a normal direction to the surface of the blank with the specified gap between the node and the tooling surface. This is useful for line die simulation of the previously formed panel to reduce tool travel therefore saving simulation time.

Define Projection Card. This card may not be repeated.

Card 1	1	2	3	4	5	6	7	8
Variable	IDPS	IDPM	GAP	NRSST	NRMST			
Type	I	I	F	I	I			
Default								

VARIABLE	DESCRIPTION
IDPS	Part ID of the blank (slave side).
IDPM	Part ID for the tool (master side).
GAP	A distance, which defines the minimum gap required.
NRSST	Normal direction of the blank: EQ.0: the normal to the surface of the blank is pointing towards the tool, EQ.1: the normal to the surface of the blank is pointing away from the tool.
NRMST	Normal direction of the tool: EQ.0: the normal to the surface of the tool is pointing towards the blank, EQ.1: the normal to the surface of the tool is pointing away from blank.

Remarks:

This feature requires consistent normal vectors for both the rigid tooling surface and the blank surface.

Revision information:

This feature is available starting in Revision 25588.

*CONTROL_FORMING_REMOVE_ADAPTIVE_CONSTRAINTS

Purpose: This keyword converts an adaptive mesh into a fully connected mesh. Adaptive constraints are removed and triangular elements are used to connect the mesh.

Card 1	1	2	3	4	5	6	7	8
Variable	PID							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
PID	Part ID (as in *PART) of the part whose adaptive mesh constraints is to be removed and its mesh converted into connected meshes.

Remarks:

In some application in sheet metal forming, such as stoning or spirngback simulation, adaptive refinement on the sheet blank may affect the accuracy of the calculation. To avoid this problem, non-adapted mesh is required. However, adaptively refined mesh has the optimal mesh density that is tailored to the tooling geometry; the resulting mesh, in its initial shape, either flat or deformed, has fewer elements (than a blank with non-adapted and uniformly-sized elements) and thus is the most efficient for simulation. If the parameter IOFLAG in *CONTROL_ADAPTIVE is turned on, such a mesh adapt.msh will be generated at the end of each simulation, with its shape conforming to the initial input blank shape.

This keyword takes the adapted mesh, removes the adaptive constraints, and use triangular elements to connect the otherwise disconnected mesh. The resulting mesh is a fully connected mesh, with the optimal mesh density, to be used to rerun the simulation (without mesh adaptivity) for a better accuracy.

Note that the original adapt.msh file from a LS-DYNA run will include not only the blank but the tooling mesh as well. In order to be used for this keyword, the original file can be read into *LS-PrePost*, with blank shown in active display only, and menu option *File → Save As → Save Active Keyword As* can be used to write out the adapted blank mesh only.

Example:

The following complete input file converts an adaptive mesh file blankadaptmsh.k (Figure 12-43 left) with the PID of 1 into a connected mesh (Figure 12-43 right). The resulting mesh will be in the dynain file.

```
*KEYWORD
*INCLUDE
blankadaptmsh.k
*PARAMETER
I blkpid           1
$--------1---------2-------
*CONTROL_TERMINATION
0.0
*CONTROL_FORMING_REMOVE_ADAPTIVE_CONSTRAINTS
$   PID
&blkpid
*set_part_list
1
&blkpid
*INTERFACE_SPRINGBACK_LSDYNA_NOTHICKNESS
1
*INTERFACE_SPRINGBACK_EXCLUDE
INITIAL_STRAIN_SHELL
INITIAL_STRESS_SHELL
*PART

$      PID       SID       MID
   &blkpid         1         1
*MAT_037
...
*SECTION_SHELL
$    SECID    ELFORM      SHRF       NIP
         1         2 0.000E+00         3
1.0,1.0,1.0,1.0
*END
```

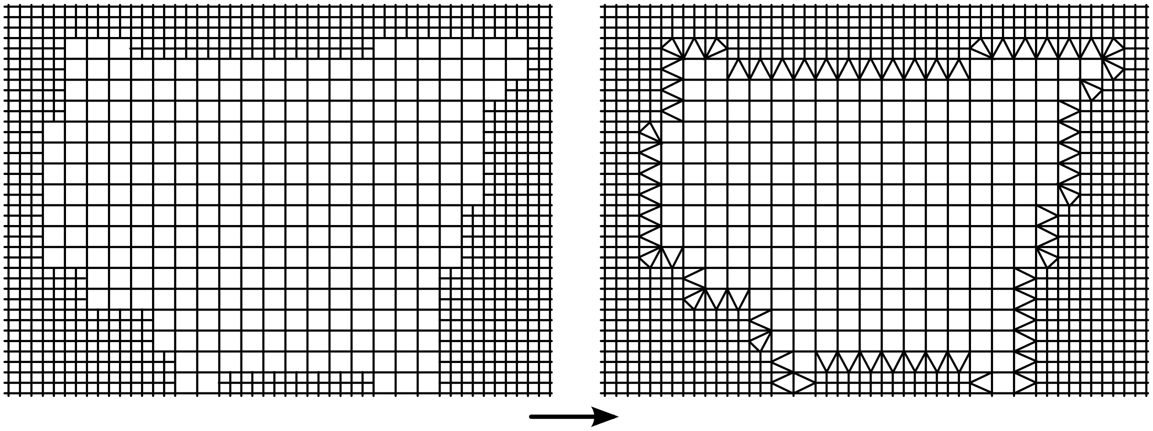


Figure 12-43. Converting an adaptive mesh to a fully connected mesh.

Revision information:

This feature is available starting from LS-DYNA Revision 108157, in both SMP, MPP, single and double precision.

*CONTROL_FORMING_SCRAP_FALL

Purpose: This keyword allows for direct and aerial trimming of a sheet metal part by trim steels in a trim die. According to the trim steels and trim vectors defined, the sheet metal part will be trimmed into a parent piece and multiple scrap pieces. The parent piece is defined as a fixed rigid body. Trimmed scraps (deformable shells) are constrained along trim edges until they come into contact with the trim steel; the edge constraints are gradually released as the trim steel's edge contacts the scrap piece, allowing for contact-based scrap fall simulation. This keyword applies to shell elements only.

Include Card 1 columns 1-6 only per each scarp piece for the *constraint release method* (see Remarks). For the *scrap trimming method* include one set of Cards 1, 2 and 3 per trim steel. The next "*" card terminates the input.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	VECTID	NDSET	LCID	DEPTH	DIST	IDRGD	IFSEED
Type	I	I	I	I	F	F	I	I
Default	none	none	none	none	none	none	none	none

Card 2	1	2	3	4	5	6	7	8
Variable	NOBEAD	SEEDX	SEEDY	SEEDZ	EFFSET	GAP	IPSET	EXTEND
Type	I	F	F	F	F	F	I	F
Default	none	none	none	none	none	none	none	none

Card 3	1	2	3	4	5	6	7	8
Variable	NEWID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
PID	Part ID of a scrap piece. This part ID becomes a dummy ID if all trimmed scrap pieces are defined by NEWID. See definition for NEWID and Figure 12-46.
VECTID	Vector ID for a trim steel movement, as defined by *DEFINE_VECTOR. If left undefined (blank), global *z*-direction is assumed.
NDSET	A node set consists of all nodes along the cutting edge of the trim steel. Note that prior to Revision 90339 the nodes in the set must be defined in consecutive order. See Remarks (LS-PrePost) below on how to define a node set along a path in LS-PrePost. This node set, together with VECTID, is projected to the sheet metal to form a trim curve. To trim a scrap out of a parent piece involving a neighboring trim steel, which also serves as a scrap cutter, the node set needs to be defined for the scrap cutter portion only for the scrap, see Figure 12-46.
LCID	Load curve ID governing the trim steel kinematics, as defined by *DEFINE_CURVE. GT.0: velocity-controlled kinematics LT.0: displacement-controlled kinematics An example input deck is provided below.
DEPTH	A small penetrating distance between the cutting edge of the trim steel and the scrap piece, as shown in Figure 12-45. Nodes along the scrap edge are released from automatically added constraints at the simulation start and are free to move after this distance is reached.
DIST	A distance tolerance measured in the plane normal to the trim steel moving direction, between nodes along the cutting edge of the trim steel defined by NDSET and nodes along an edge of the scrap, as shown in Figure 12-44. This tolerance is used to determine if the constraints need to be added at the simulation start to the nodes along the trim edge of the scrap piece.
IDRGD	Part ID of a parent piece, which is the remaining sheet metal after the scrap is successfully trimmed out of a large sheet metal. Note the usual *PART needs to be defined somewhere in the input deck, along with *MAT_20 and totally fixed translational and rotational DOFs. See Figure 12-46.

VARIABLE	DESCRIPTION
IFSEED	A flag to indicate the location of the scrap piece. EQ.0: automatically determined. The trim steel defined will be responsible to trim as well as to push (have contact with) the scrap piece. EQ.1: automatically determined, however, the trim steel in definition will only be used to trim out the scrap, not to push (have contact with) the scrap piece. EQ.-1: user specified by defining SEEDX, SEEDY, and SEEDZ
NDBEAD	A node set to be excluded from initially imposed constraints after trimming. This node set typically consists of nodes in the scrap draw bead region where due to modeling problems the beads on the scrap initially interfere with the beads on the rigid tooling; it causes scrap to get stuck later in the simulation if left as is. See Figure 12-47.
SEEDX, SEEDY, SEEDZ	x, y, z coordinates of the seed node on the scrap side; define only when IFSEED is set to "-1". See Figure 12-46.
EFFSET	Scrap edge offset amount away from the trim steel edge, towards the scrap seed node side. This is useful to remove initial interference between the trimmed scrap (because of poorly modeled trim steel) and coarsely modeled lower trim post. See Figure 12-46.
GAP	Scrap piece offset amount from the part set defined by IPSET (e.g. top surfaces of the scrap cutters), in the direction of the element normals of the IPSET. This parameter makes it easier to remove initial interference between the scrap and other die components. See Figure 12-48.
IPSET	A part set ID from which the scrap will be offset to remove the initial interference, works together only with GAP. The part set ID should only include portions of tool parts that are directly underneath the scrap (top surface portion of the tools). The normals of the IPSET must point toward the scrap. The parts that should belong to IPSET are typically of those elements on the top surface of the scrap cutter, see Figure 12-48.

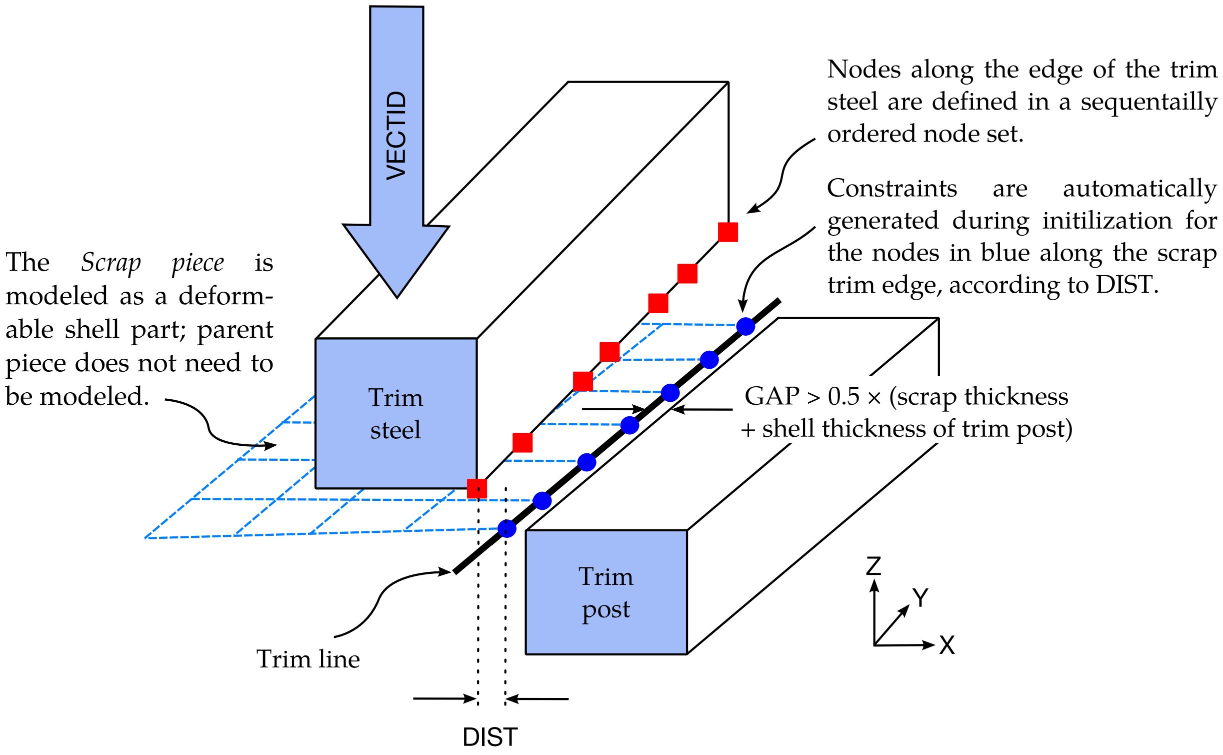

VECTID
Nodes along the edge of the trim steel are defined in a sequentailly ordered node set.
Constraints are automatically generated during initilization for the nodes in blue along the scrap trim edge, according to DIST.
The *Scrap piece* is modeled as a deformable shell part; parent piece does not need to be modeled.
Trim steel
GAP > 0.5 × (scrap thickness + shell thickness of trim post)
Trim post
Trim line
DIST
Z
Y
X

Figure 12-44. Modeling details of the constraint release method. *Drawing modified from the original sketches courtesy of the Ford Motor Company.*

VARIABLE	DESCRIPTION
EXTEND	An amount to extend a trim steel's edge based on the NDSET defined, so it can form a continuous trim line together with a neighboring trim steel, whose edge may also be extended, to trim out the scrap piece. See Figure 12-46.
NEWID	New part ID of a scrap piece for the scrap area defined by the seed location. If this is not defined (left blank) or input as "0", the scrap piece will retain original PID as its part ID. See Figure 12-46. This is useful in case where one original scrap is trimmed into multiple smaller pieces, and contacts between these smaller pieces need to be defined.

Background:

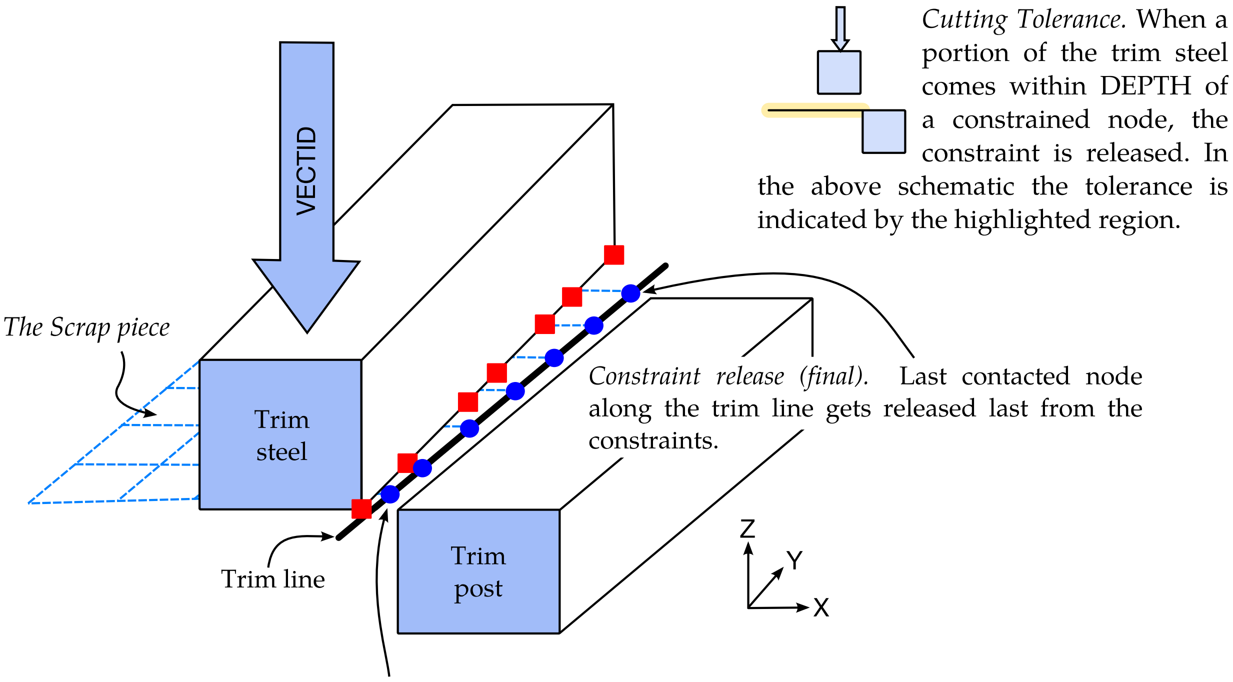
Sheet metal trimming and the resulting scrap fall are top factors in affecting the efficiency of stamping plants worldwide. Difficult trimming conditions, such as those multiple direct trims, a mixture of direct and cam trims, and multiple cam trims involving bypass condition, can cause trimmed scraps to get stuck around and never separate from the trim edge of the upper trim steels or lower trim post. Inappropriate design of die structure and scrap chute can slow down or prevent scraps from tumbling out to the scrap collectors. Smaller scrap pieces (especially aluminum) can sometimes shoot straight up, and get stuck and gather in areas of the die structure. All these problems result in shutdowns of stamping presses, reducing stroke-per-minute (SPM) and causing hundreds of thousands of dollars in lost productivity.

With this keyword, engineers can consider the trimming details, manage the scrap trim and the drop energy, study different trimming sequences, explore better die structure and scrap chutes design and layout before a trim die is even built. This feature is developed in conjunction with the *Ford Motor Company*.

The constraint release method:

Prior to Revision 91471 (see Revision), simulating the scrap trim and fall uses the "constraint release" method, where only the scrap piece is modeled and defined.

As shown in Figure 12-44, the scrap piece is modeled as a deformable body and the trim steel and trim post as rigid shell elements, while the parent piece does not need to be modeled at all. Between the trim edge of the scrap piece and the post there should be a gap (indicated by GAP in the figure). The gap ensures that the contact interface (to be explained later) correctly accounts for the shell thickness along the edge. A gap that is too small may cause initial penetration between the scrap and the post which may manifest as unphysical adhesion between the scrap and the post.



Cutting Tolerance. When a portion of the trim steel comes within DEPTH of a constrained node, the constraint is released. In the above schematic the tolerance is indicated by the highlighted region.

Constraint release (final). Last contacted node along the trim line gets released last from the constraints.

Constraint release (start). First node in contact is the first released from constraints. The motion of the trim steel is carried onto the scrap piece by the contact interface.

Figure 12-45. Contact-based separation and contact-driven kinematics and dynamics in the constraint release method. *Drawing modified from the original sketches courtesy of the Ford Motor Company.*

The edge of the scrap piece should initially be flush with that of the trim post (perpendicular to the trim direction), just as exactly what happens in the production environment. If the scrap is unrealistically positioned above the trim post edge, the scrap may be permanently caught between the trim steel and the post under a combination of uncertain trimming forces as the trim steel moves down.

During initialization, constraints are added automatically on the nodes along the scrap trim edge corresponding to the node set (NDSET) along the trim steel, based on the supplied tolerance variable DIST and trim vector VECTID. Although the direction of the path is not important, prior to Revision 90339, the NDSET must be arranged so that the nodes are in a sequential order (*LS-PrePost 4.0* creating node set by *path*). As the edge of the trim steel comes within DEPTH distance of the trim line, the constraints are removed. The contact interfaces serve to project the motion of the trim steel onto the scrap piece, see Figure 12-45.

The scrap trimming method:

The original simplified method has the following drawbacks:

1. No scrap trimming – the scrap piece cannot be trimmed directly from a parent piece; an exact scrap piece after trimming must be modeled.

2. Poorly (or coarsely) modeled draw beads in the scrap piece do not fit properly in badly modeled draw beads on the tooling, resulting in initial interferences between the two and therefore affecting the simulation results.

3. For poorly (or coarsely) modeled scrap edges and trim posts, users have to manually modify the scrap trim edges to clear the initial interference with the trim posts.

4. Users must clear all other initial interferences (e.g. between scrap and scrap cutter) manually.

Based on users' feedback, a new method "scrap trimming" (after Revision 91471) has been developed to address the above issues and to, furthermore, reduce the effort involved in preparing the model. The new method (Figure 12-46) involves trimming scrap from an initially large piece of sheet metal, leaving the parent piece as a fixed rigid body. The trim lines are obtained from the trim steel edge node set NDSET and the trim vector VECTID.

Parameters related to the constraint release method:

1. The value of DEPTH is typically set to one-half of the scrap thickness.

2. The initial gap separating the scrap from the post must be greater than the average of the scrap and post thickness values, see Figure 12-44.

3. The input parameter DIST should be set larger than the maximum distance between nodes along the trim steel edge and scrap edge in the view along the trim direction, see Figure 12-44.

Parameters related to the scrap trimming method:

4. Similar to DEPTH, EFFSET should be typically set to one-half of the scrap thickness, although it may be larger for some poorly modeled trim steels and trim posts.

Contact:

Only *CONTACT_FORMING contact interfaces are allowed for contact between the scrap piece and the trim steel. In particular, *CONTACT_FORMING_SURFACE_TO_SURFACE is recommended. A negative contact offset must be used; this is done typically by setting the variable MST in *CONTACT_FORMING_SURFACE_TO_SURFACE to the negative thickness value of the scrap piece.

For contact between the scrap piece and the shell elements in all the other die structures, *CONTACT_AUTOMATIC_GENERAL should be used for the edge-to-edge contact frequently encountered during the fall of the scrap piece. All friction coefficients should be small. The explicit time integrator is recommended for the modeling of scrap trim and fall. Mass scaling is not recommended.

LS-PrePost:

The node set (NDSET) defined along the trim steel edge can be created with *LS-PrePost 4.0*, via *Model/CreEnt/Cre, Set Data, *SET_NODE, ByPath*, then select nodes along the trim edge continuously until finish and then hit *Apply.*

Keyword examples – the constraint release method:

A partial example of using the keyword below includes a node set ID 9991 along the trim steel (PID 2) edge used to release the constraints between the scrap piece with PID 1, and the parent piece. The LCID for the trim steel kinematics is (+)33 (load curve is controlled by velocity) moving in –Z direction. The trimming velocity is defined as 1000 mm/s and the retracting velocity is 4000 mm/s. The variables DEPTH and DIST are set to 0.01 and 2.5, respectively. The contact interface between the trim steel and scrap piece is defined using *CONTACT_FORMING_SURFACE_TO_SURFACE and contact between the scrap and all other die structures are defined using *CONTACT_-AUTOMATIC_GENERAL.

```
*KEYWORD
*CONTROL_TERMINATION
&endtime
*CONTROL_FORMING_SCRAP_FALL
$       PID    VECTID     NDSET      LCID     DEPTH      DIST
         1                 9991        33      0.75      2.0
*SET_NODE_LIST
      9991
     24592     24591     24590     24589     24593     24594     24595     24596
*BOUNDARY_PRESCRIBED_MOTION_rigid
$pid,dof,vad,lcid,sf,vid,dt,bt
2,3,0,33,-1.0
*DEFINE_CURVE
33
0.0,0.0
0.216,1000.0
0.31,-4000.0
0.32,0.0
0.5,0.0
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*CONTACT_forming_surface_to_surface_ID
         1
         1         2         3         3         0         0         0         0
      0.02       0.0       0.0       0.0      20.0         0       0.01.0000E+20
$#     sfs       sfm       sst       mst      sfst      sfmt       fsf       vsf
```

```
       0.0       0.0       0.0      &mst       1.0       1.0       1.0       1.0
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*CONTACT_AUTOMATIC_GENERAL_ID
         2

*END
```

For the negative option of LCID, displacement will be used as input to control the tool kinematics. A partial example is provided below, where LCID is defined as a negative integer of a load curve, controlling the trim steel kinematics. The trim steel is moving down for 27.6075 mm in 0.2 sec to trim, and moving up for the same distance to its original position in 0.3 sec to retract. Although this option is easier to use, the corresponding velocity from the input time and displacement must be realistic for a realistic simulation.

```
*CONTROL_FORMING_SCRAP_FALL
$ LCID<0: trimming steel kinematics is controlled by displacement.
$      PID    VECTID     NDSET      LCID     DEPTH      DIST
         1        44         1    -33332      0.70      2.00
*DEFINE_VECTOR
44,587.5,422.093,733.083,471.104,380.456,681.412
*BOUNDARY_PRESCRIBED_MOTION_rigid_LOCAL
$pid,dof,vad,lcid,sf,vid,dt,bt
11,3,2,33332,1.0,44
*DEFINE_CURVE
33332
0.0,0.0
0.2,-27.6075
0.5,0.0
```

A keyword example – the scrap trimming method:

The keyword example below shows three scrap pieces, with original PID &SPID1, new PIDs 1001 and 1002, being trimmed out of a larger scrap &SPID1; the remaining parent piece is defined as a fixed rigid body with PID 110. A different seed location is defined separately for each scrap. The scraps &SPID1, 1001 and 1002 are each offset by 0.60mm in the area of the seed location defined, in the direction normal to the elements defined by IPSET 887, 888, and 889, respectively. The trim edge offset is 0.90mm for all scraps. The draw bead node sets to be released are, 987, 988, 989 for each scrap as defined by the corresponding seed locations.

```
*CONTROL_FORMING_SCRAP_FALL
$      PID    VECTID     NDSET     VLCID     DEPTH      DIST     IDRGD    IFSEED
    &spid1    &cord1    &nset1      1800   &depth1      2.00       110        -1
$   NDBEAD     seedx     seedy     seedz    effset       GAP     IPSET    EXTEND
       987  -528.046    373.40   710.000      0.90      0.60       887       8.0
         0
    &spid1    &cord2    &nset2      1801   &depth1      2.00       110        -1
       987  -528.046    373.40   710.000      0.90      0.60       887       8.0
         0
    &spid1    &cord3    &nset3      1802   &depth1      2.00       110        -1
       987  -528.046    373.40   710.000      0.90      0.60       887       8.0
         0
$
    &spid1    &cord3   &nset33      1802   &depth1      2.00       110        -1
       988  -252.452   383.322   799.974      0.90      0.60       888       8.0
```

```
      1001
    &spid1    &cord4     &nset4       1803    &depth1       2.00        110         -1
       988  -252.452    383.322    799.974       0.90       0.60        888        8.0
      1001
    &spid1    &cord5     &nset5       1804    &depth1       2.00        110         -1
       988  -252.452    383.322    799.974       0.90       0.60        888        8.0
      1001
$
    &spid1    &cord5    &nset55       1804    &depth1       2.00        110         -1
       989    74.452    404.522    857.974       0.90       0.60        889        8.0
      1002
    &spid1    &cord6     &nset6       1805    &depth1       2.00        110         -1
       989    74.452    404.522    857.974       0.90       0.60        889        8.0
      1002
    &spid1    &cord7     &nset7       1806    &depth1       2.00        110         -1
       989    74.452    404.522    857.974       0.90       0.60        889        8.0
      1002
```

Revision/Other information:

A graphical user interface capable of setting up a complete input deck for the original simplified method is now available in *LS-PrePost 4.0* under *APPLICATION/Scrap Trim* (http://ftp.lstc.com/anonymous/outgoing/lsprepost/4.1/). A reference paper regarding the development and application of this keyword for the constraint release method can be found in the *proceedings of the 12th International LS-DYNA User's Conference*. The following provides a list of revision history for the keyword:

1. The *constraint release method* is available between LS-DYNA Revision 63618 and 91471.

2. The *scrap trimming method* is available starting in Revision 91471.

3. The parameter NEWID is available starting in Revision 92578.

4. The restriction of NDSET must be defined in a consecutive order is lifted starting in Revision 90339.

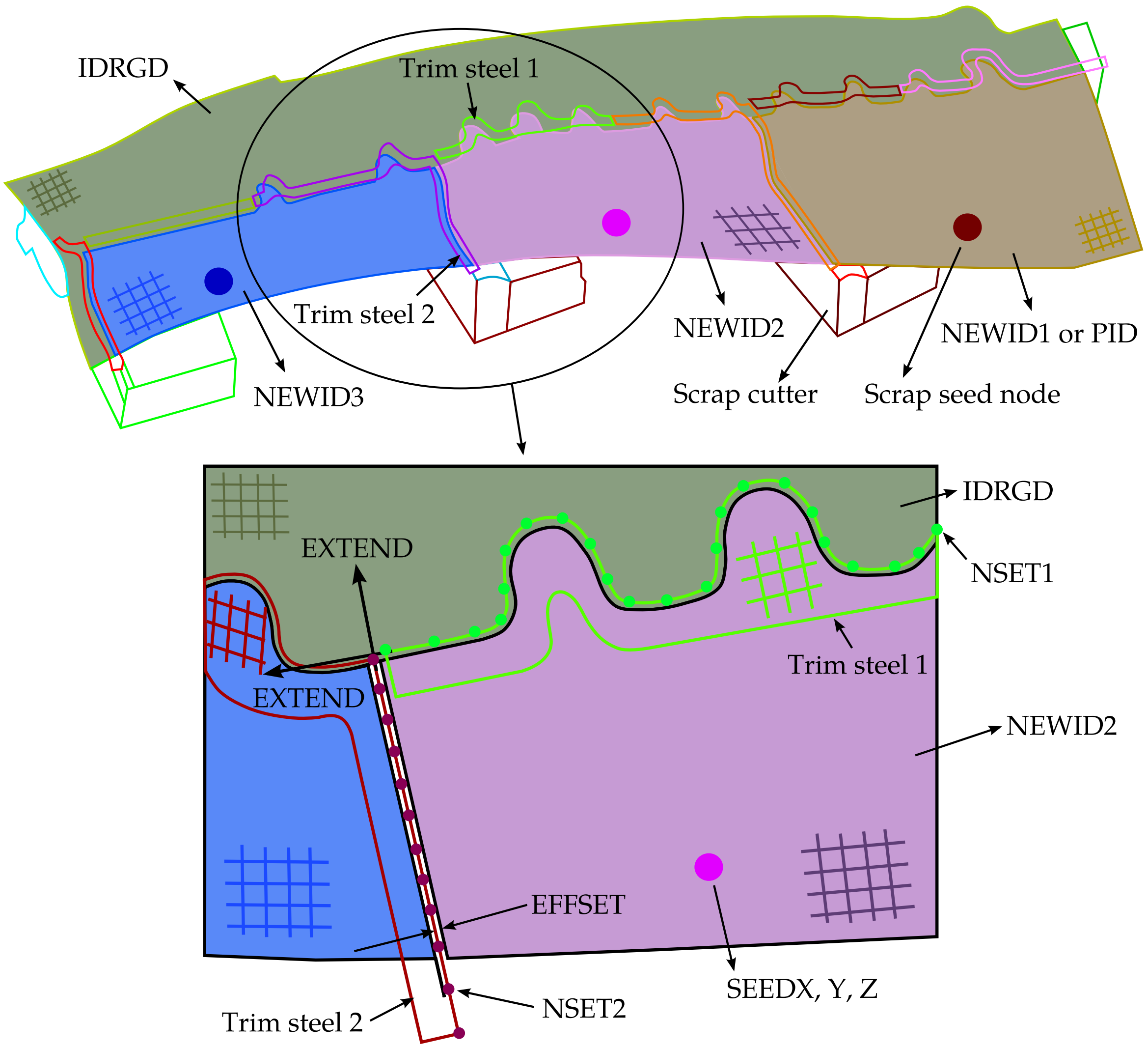


Figure 12-46. Trimming of multiple scraps and parameter definitions in the scrap trimming method. *Model courtesy of the Ford Motor Company.*

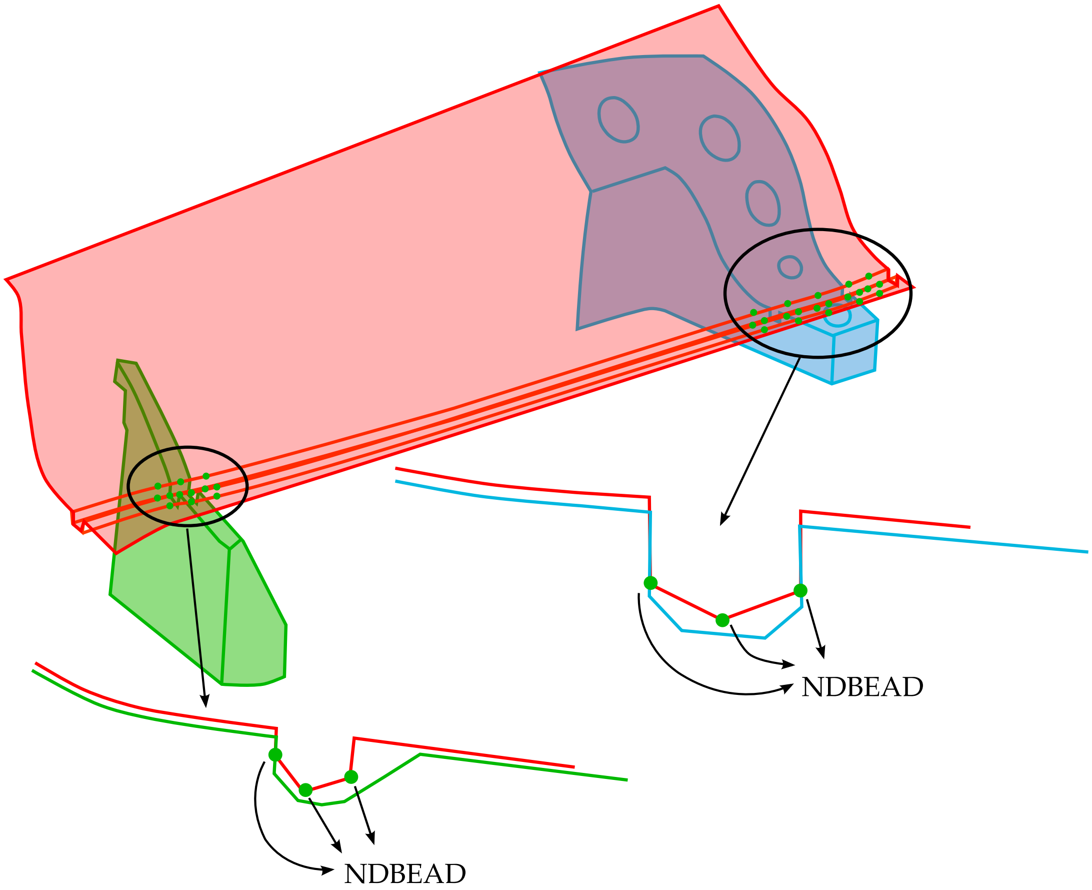
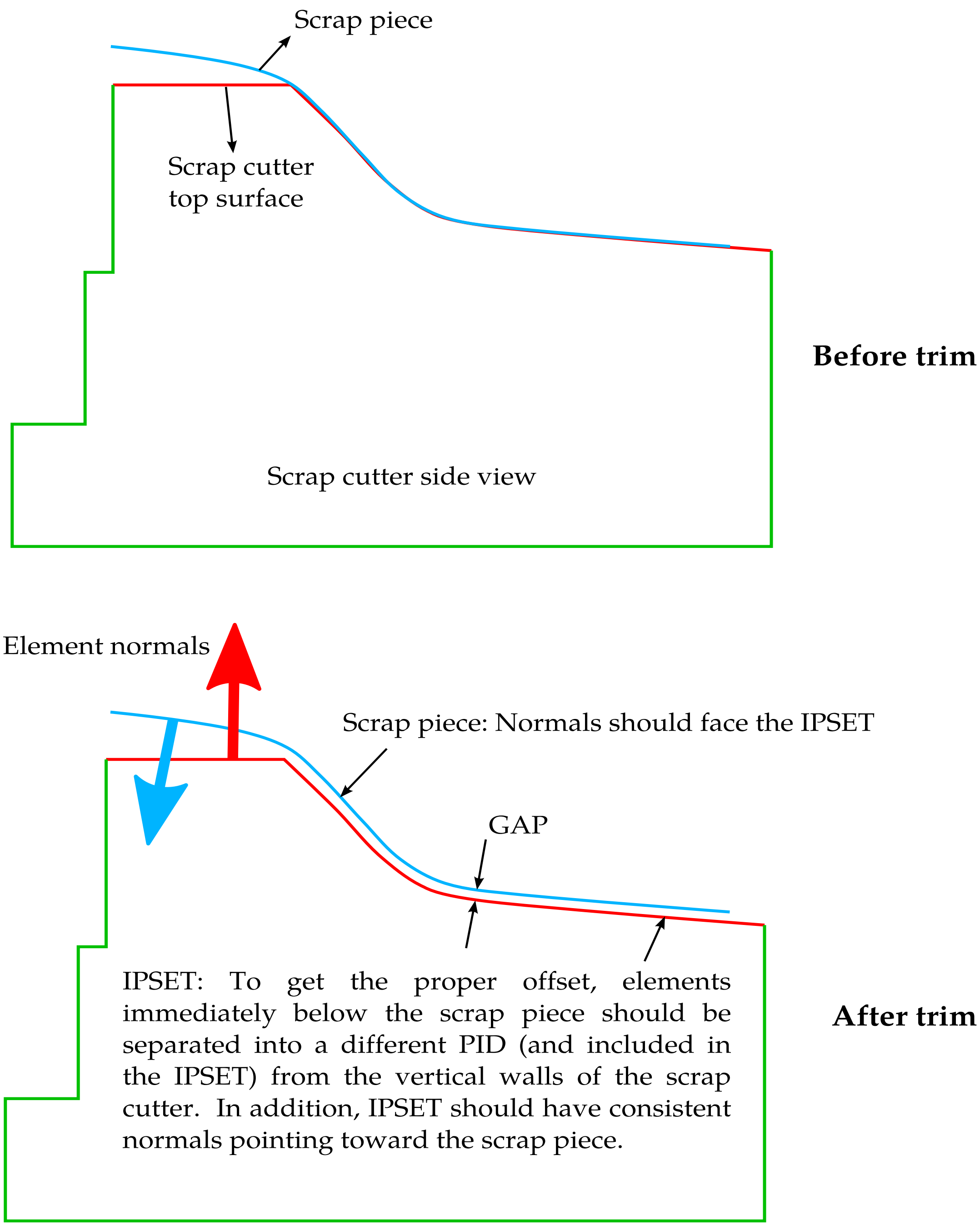


Figure 12-47. Definition of NDBEAD in the scrap trimming method. *Model courtesy of the Ford Motor Company.*



IPSET: To get the proper offset, elements immediately below the scrap piece should be separated into a different PID (and included in the IPSET) from the vertical walls of the scrap cutter. In addition, IPSET should have consistent normals pointing toward the scrap piece.

Figure 12-48. Element normal of the IPSET in the scrap trimming method. *Model courtesy of the Ford Motor Company.*

*CONTROL_FORMING_SHELL_TO_TSHELL

Purpose: This keyword is created to allow users to easily change the element type from thin shell elements (*SECTION_SHELL) to thick shell elements (*SECTION_TSHELL), and to generate segments on both top and bottom side of the thick shells. Note that mesh adaptivity is also supported.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	THICK	MIDSF	IDSEGB	IDSEGT			
Type	I	I	I	I	I			
Default	none	none	none	none	none			

VARIABLE	DESCRIPTION

PID — Part ID of the thin shell elements.

THICK — Thickness of the thick shell elements (Tshell).

MIDSF — Tshell mid-plane position definition (see Figure 12-49):

MIDSF.EQ.0: Mid-plane is at thin shell surface.

MIDSF.EQ.1: Mid-plane is at one half of THICK above thin shell surface.

MIDSF.EQ.-1: Mid-plane is at one half of THICK below thin shell surface.

IDSEGB — Set ID of the segments to be generated on the bottom layer of the Tshells, which can be used for segment-based contact. The bottom side of Tshells is the opposite side of the positive normal side of the thin shells, see Figure 12-49.

Note the default normal of the generated segments are consistent with the thin shells' normal. To reverse this default normal, set the IDSEGB to a negative number.

Under the FORMING type of contact, if the generated segments are used as a slave member in contact with a master member of a rigid body, the rigid body's normals must be consistent and facing the slave segments. Note that the slave segments normals are not required to point at the rigid bodies, although they should be made consistent.

VARIABLE **DESCRIPTION**

IDSEGT

Set ID of the segments to be generated on the top layer of the Tshells, which can be used for segment-based contact. The top side of a Tshells is the same side of the positive normal side of the thin shells, see Figure 12-49.

Note the normal of the generated segments are consistent with the thin shells' normal. To reverse this default normal, set the IDSEGT to a negative number.

Remarks:

This keyword will convert thin shell elements to thick shell elements. The position of the thick shells' mid-plane in reference to the thin shell's surface is dependent on MIDSF (Figure 12-49). Node IDs of the thick shell elements will be the same as those for the thin shells. Element IDs of the thick shell elements will start at 2 (so renumber element IDs of other PID accordingly). Only one layer of thick shells will be created.

New nodes generated adaptively from their parent nodes with *BOUNDARY_SPC are automatically constrained accordingly.

This feature is developed as requested by JSOL Corporation.

Examples:

1)A standalone part of thin shell elements can be changed to thick shell elements with a simplified small input deck. The following will convert shell elements with PID 100 of thickness 1.5mm to thick shell elements of PID 100 with thickness of 2.0mm, with thick shell meshes stored in a file "dynain.geo". Note that MIDSF, IDSEGB and IDSEGT cannot be used in this case.

```
*KEYWORD
*CONTROL_TERMINATION
0.0
*INCLUDE
shellupr.k
*SET_PART_LIST
1,
100
*PART
Sheet blank
100,100,100
*SECTION_SHELL
$       SID    ELFORM      SHRF       NIP     PROPT
        100         2     0.833         3       1.0
$        T1        T2        T3        T4      NLOC
1.5,1.5,1.5,1.5
*MAT_037
$#      MID        RO         E        PR      SIGY      ETAN         R     HLCID
100        7.8500E-9 2.1000E+5  0.333000                           1.00     90905
```

```
*DEFINE_CURVE
     90905
     0.000000000E+00     0.380000000E+03
     0.300000003E-02     0.392489226E+03
     0.600000005E-02     0.403294737E+03
     0.899999961E-02     0.412847886E+03
     0.120000001E-01     0.421429900E+03
     0.150000006E-01     0.429234916E+03
     0.179999992E-01     0.436402911E+03
     0.209999997E-01     0.443038343E+03
*INTERFACE_SPRINGBACK_LSDYNA
1
OPTCARD,,,1
*CONTROL_FORMING_SHELL_TO_TSHELL
$      PID     THICK
       100       2.0
*END
```

2)The conversion can also be done in an input deck set up for a complete metal forming simulation with thin shell elements as a sheet blank. The conversion happens in the beginning of the simulation, as shown in an example below. Only the keywords needed change are listed, commented out with $ signs and replaced with appropriate cards for thick shells. The thin shell sheet blank with PID 1 is to be converted to a thick shell sheet blank with thickness of 1.6mm, noting that the *SECTION_TSHELL should be defined instead of *SECTION_SHELL for the sheet blank. Corresponding material type for the sheet blank (*MAT_037) also needs to be changed to a type that supports solid element simulation (*MAT_024). The mid-plane of the thick shells will be one half of 1.6 mm below the thin shells' surface, with segment IDs 10 (IDSEGB) and 11 (IDSEGT) created on the bottom and top side of the thick shells, respectively, as shown in Figure 12-49. IDSEGB 10 with SSTYP 0 is defined to contact with the lower punch (part set ID 2) with MSTYP 2, and IDSEGT 11 with SSTYP 0 is used for contact with the upper die cavity with part set ID of 3, of MSTYP 2.

```
*KEYWORD
...
*PART
Sheet blank
1,1,1
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$                    Blank property
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$*SECTION_SHELL
$      SID    elform      SHRF       nip     PROPT   QR/IRID     ICOMP     SETYP
$        1         2     0.833         3       1.0
$       T1        T2        T3        T4      NLOC
$&b1thick,&b1thick,&b1thick,&b1thick
*SECTION_TSHELL
$      SID    elform      SHRF       nip     PROPT   QR/IRID     ICOMP     SETYP
         1         1     0.833      &nip       1.0
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$*MAT_037
$      MID        RO         E        PR      SIGY      ETAN         R     HLCID
$        1  7.85E-09  2.07E+05      0.28                                      92
*MAT_024
$      MID        RO         E        PR      SIGY      ETAN      FAIL      TDEL
         1  7.85E-09  2.07E+05      0.28     382.8       0.0       0.0       0.0
$        C         P      LCSS      LCSR        VP
```

```
       0.0       0.0        92         0       0.0
$     EPS1      EPS2      EPS3      EPS4      EPS5      EPS6      EPS7      EPS8

$      ES1       ES2       ES3       ES4       ES5       ES6       ES7       ES8

*DEFINE_CURVE
92,,,0.5
    0.0000000000E+00    3.8276000000E+02
    4.0000000000E-03    3.9616000000E+02
    8.0000000000E-03    4.0695000000E+02
           ⋮                    ⋮
*INTERFACE_SPRINGBACK_LSDYNA
1
OPTCARD,,,1
*CONTROL_FORMING_SHELL_TO_TSHELL
$      PID     THICK     MIDSF    IDSEGB    IDSEGT
       100       1.6        -1        10        11
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE
$     SSID      MSID     SSTYP     MSTYP
        11         2         0         2
         ⋮         ⋮         ⋮
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE
$     SSID      MSID     SSTYP     MSTYP
        10         3         0         2
         ⋮         ⋮         ⋮
...
*END
```

Revision information:

This feature is available in LS-DYNA starting in Revision 104903. The following revisions indicate the Revision history of additional features:

1) Revision 106116: negative option of IDSEGB and IDSEGT.
2) Revision 106162: MIDSF, IDSETGB and IDSEGT becomes available.
3) Revision: 106217: automatically add *BOUNDARY_SPC for newly generated nodes whose parent nodes are assigned with *BOUNDARY_SPC.

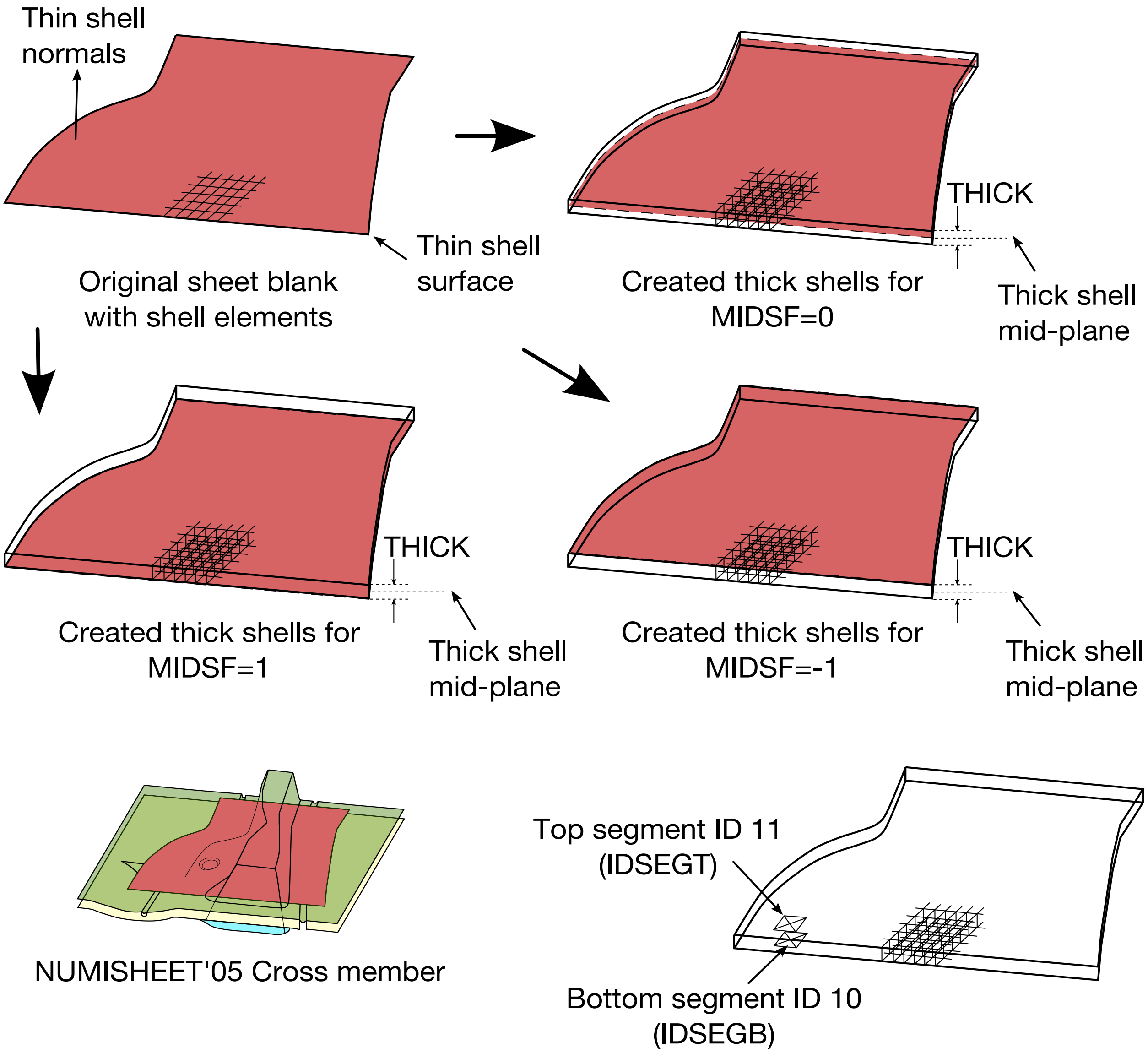


Figure 12-49. Converting thin shells to thick shells in sheet metal forming

*CONTROL_FORMING_STONING

Purpose: This feature is developed to detect surface lows or surface defects formed during metal stamping. This calculation is typically performed after a springback simulation. A curvature-based method is implemented with the feature. Users have the option to check an entire part or just a few local areas, defined by node set or shell element set. In each area, direction of the stoning action can be specified by two nodes (see **Remarks** below) or simply allow the program to automatically determine the stoning direction.

Card 1	1	2	3	4	5	6	7	8
Variable	ISTONE	LENGTH	WIDTH	STEP	DIRECT	REVERSE	METHOD	
Type	I	F	F	F	F	I	I	
Default	none	0.0	0.0	0.0	0.0	0	0	

Card 2	1	2	3	4	5	6	7	8
Variable	NODE1	NODE2	SID	ITYPE	V1	V2	V3	
Type	I	I	I	I	F	F	F	
Default	0	0	0	0	0.0	0.0	0.0	

VARIABLE	DESCRIPTION
ISTONE	Stoning calculation option. EQ.1: calculate panel surface quality using stoning method.
LENGTH	Length of the stone.
WIDTH	Width of the stone.
STEP	Stepping size of the moving stone.
DIRECT	Number of automatically determined stoning direction(s).

VARIABLE	DESCRIPTION
REVERSE	Surface normal reversing option: EQ.0: do not reverse surface normals. EQ.1: reverse surface normals.
METHOD	Stoning method. EQ.0: curvature-based method.
NODE1	Tail node defining stoning moving direction.
NODE2	Head node defining stoning moving direction.
SID	Node or shell set ID.
ITYPE	Set type designation: EQ.1: node set EQ.2: element set
V1, V2, V3	Vector components defining stoning direction (optional).

About stoning:

Stoning is a quality checking process on class-A exterior stamping panels. Typically the long and wider surfaces of an oil stone of a brick shape are used to slide and scratch in a given direction against a localized area of concern on a stamped panel. Surface “lows” are shown where scratch marks are not visible and “highs” are shown in a form of scratch marks. This keyword is capable of predicting both the surface “lows” and “highs”. Since stoning process is carried out after the stamping (either drawn or trimmed) panels are removed from the stamping dies, a springback simulation needs to be performed prior to conducting a stoning analysis.

Modeling guidelines:

As a reference, typical stone length and width can be set at 150.0 and 30.0 mm, respectively. The step size of the moving stone is typically set about the same order of magnitude of the element length. The smallest element length can be selected as the step size.

The variable DIRECT allows for the automatic definition of the stoning directions. Any number can be selected but typically 2 is used. Although CPU time required for the stoning calculation is trivial, a larger DIRECT consumes more CPU time.

Stoning is performed on the outward normal side of the mesh. Element normals must be consistent and oriented accordingly. Element normal can be automatically made consistent in *LS-PrePost4.0* under *EleTol/Normal* menu. Alternatively, the variable REVERSE provides in the solver an easy way to reverse a part with consistent element normals before the computation.

The variables NODE1 and NODE2 are used to define a specific stoning direction. The stone is moved in the direction defined by NODE1 to NODE2. Alternatively, one can leave NODE1 and NODE2 blank and define the number of automatically determined stoning directions by using the variable DIRECT. Furthermore, stoning direction can also be defined using a vector by defining the variables V1, V2, and V3.

The blank model intended for analysis can be included using keyword *INCLUDE. If nothing is defined for SID and ITYPE then the entire blank model included will be used for stoning analysis.

A large area mesh can be included in the input file. An ELSET must also be included, which defines a local area that requires stoning computation. Alternatively, an ELSET can define several local areas to be used for the computation. Furthermore, an ELSET should not include meshes that have reversed curvatures. An ELSET can be easily generated using *LS-PrePost4.0*, under *Model/CreEnt/Cre/Set_Data/*SET_SHELL*.

Since stoning requires high level of accuracy in springback prediction, it is recommended that the SMOOTH option in keyword *CONTACT_FORMING_ONE_-WAY_SURFACE_TO_SURFACE to be used during the draw forming simulation. Not all areas require SMOOTH contact, only areas of interest may apply. In addition, meshes in the areas of concern need to be very fine, with average element size of 1 to 2 mm. Mesh adaptivity is not recommended in the SMOOTH/stoning areas. Also, mass scaling with DT2MS needs to be sufficiently small to reduce the dynamic effect during forming. For binder closing of large exterior panels, implicit static method using *CONTROL_IMPLICIT_FORMING type 2 is recommended, to further reduce potential buckles caused by the inertia effect.

Stoning results/output:

It is recommended that double precision version of LS-DYNA be used for this application. The output of the stoning simulation results is in a file named “filename.output”, where “filename” is the name of the LS-DYNA stoning input file containing this keyword, without the file extension. The stoning results can be viewed using *LS-PrePost4.0*, under MFPost/FCOMP/Shell_Thickness.

Application example:

An example of a stoning analysis on a Ford Econoline door outer panel is provided for reference. The original part model comes from *National Crash Analysis Center* at *The George Washington University*. The original part was modified heavily in *LS-PrePost4.0* to fit the needs of the demonstration purpose. Binder and addendum were created and sheet blank size was assumed. The blank is assigned 0.65mm thickness and a BH210 properties with *MAT_037. Shell thickness contour plots for the drawn and trimmed panels are shown in Figures 12-50 and 12-51, respectively. Springback amount in Z is plotted in Figure 12-52. The complete input deck used for the stoning simulation is provided below for reference; where, a local area mesh of the door handle after springback simulation "Doorhandle.k" and an element set "elset1.k" are included in the deck. Locations of the ELSETs are defined for the upper right (Figure 12-53 left) and lower right corners (Figure 12-54 left) of the door handle, where "mouse ear" are expected.

```
*KEYWORD
*TITLE
Stoning Analysis
*INCLUDE
Doorhandle.k
*INCLUDE
elset1.k
*CONTROL_FORMING_STONING
$   ISTONE    LENGTH     WIDTH      STEP    DIRECT   REVERSE    METHOD
         1     150.0       4.0       1.0         9         0         0
$    NODE1     NODE2       SID     ITYPE
                             1         2
*END
```

Stoning results are shown in Figures 12-53 (right) and 12-54 (right) for the upper right and lower right corners, respectively. "Mouse ears" are predicted where anticipated.

Revision information:

The stoning feature is available in LS-DYNA Revision 54398 and later releases. Vector component option is available in Revision 60829 and later releases.

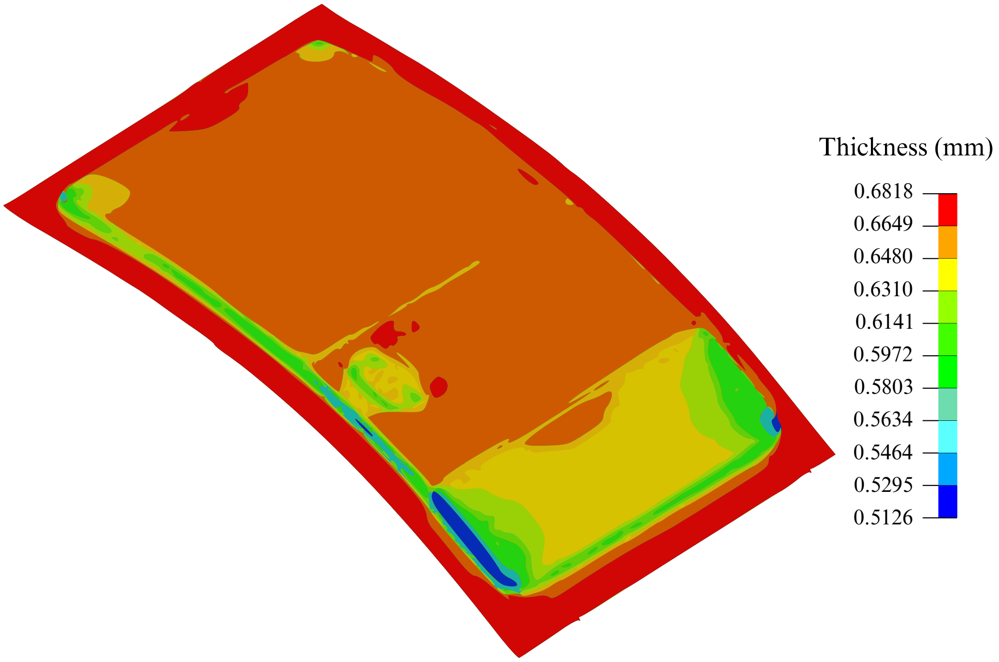


Figure 12-50. Thickness contour of the panel after draw simulation.

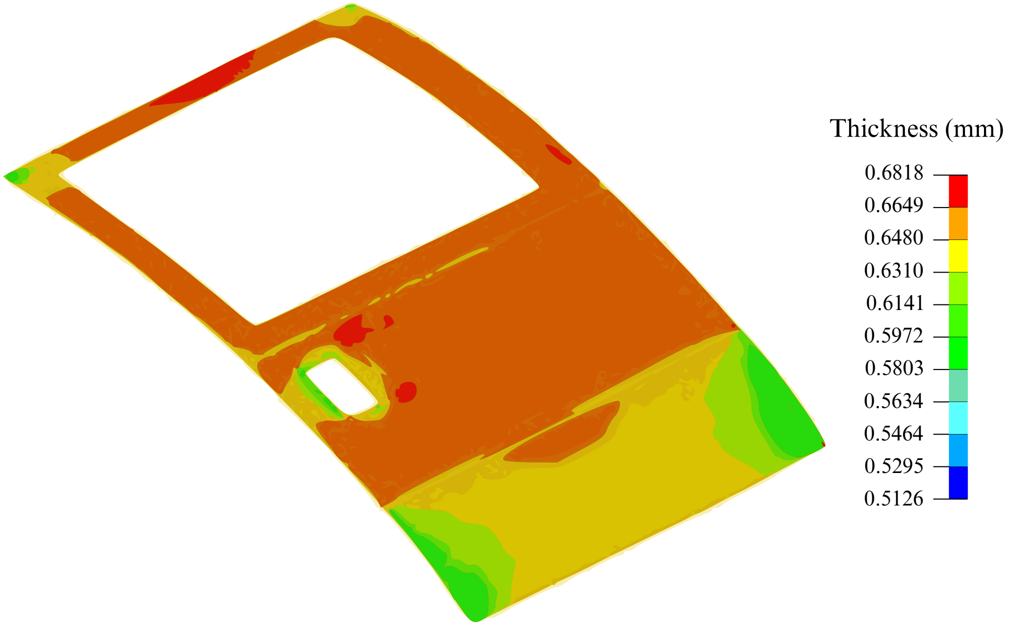


Figure 12-51. Thickness contour of the panel after trimming.

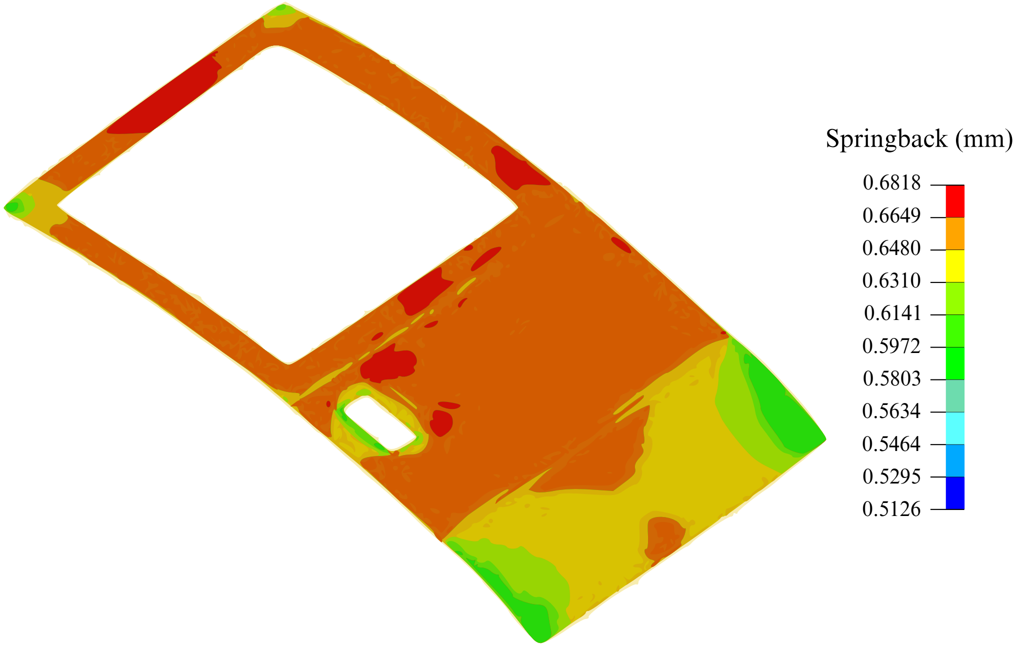


Figure 12-52. Springback amount (mm).

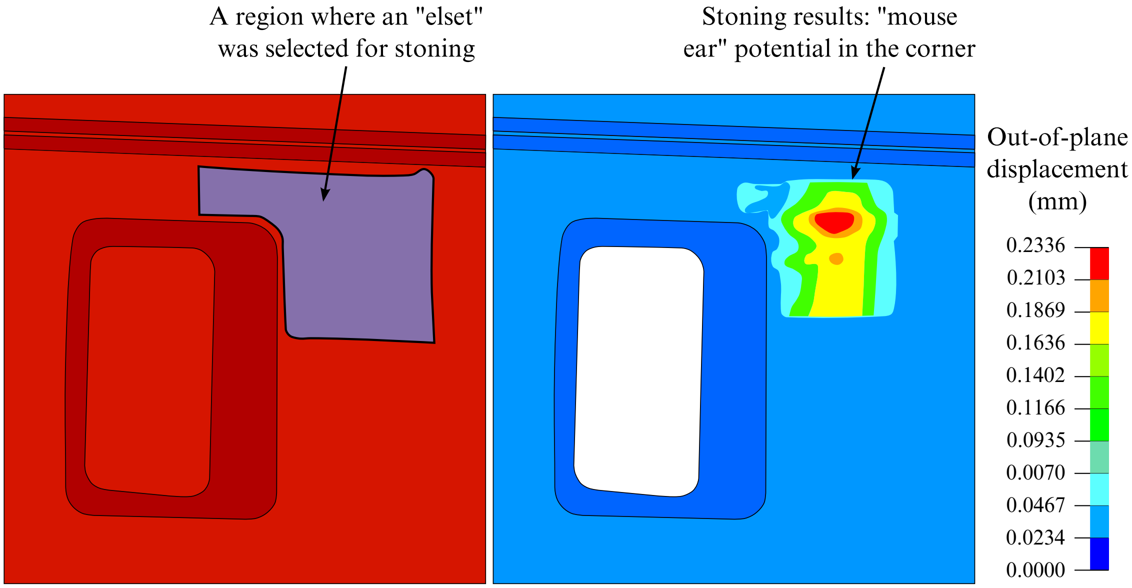

A region where an "elset" was selected for stoning
Stoning results: "mouse ear" potential in the corner
Out-of-plane displacement (mm)
0.2336
0.2103
0.1869
0.1636
0.1402
0.1166
0.0935
0.0070
0.0467
0.0234
0.0000

Figure 12-53. Stoning simulation for the upper right door corner.

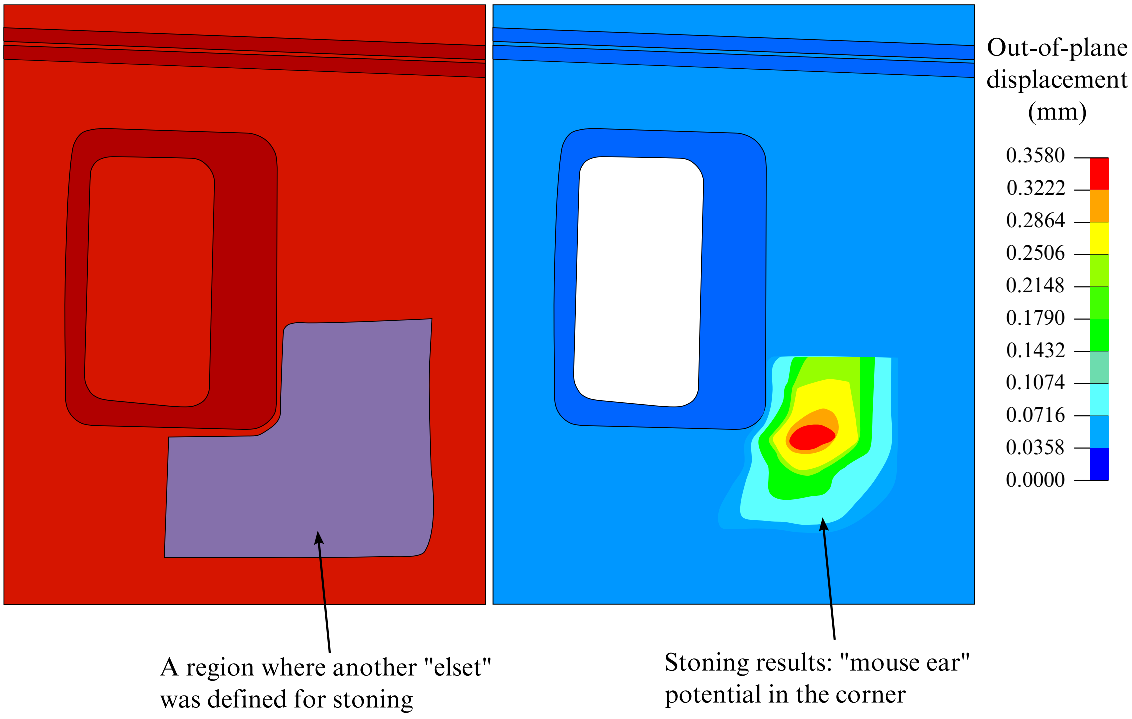


Figure 12-54. Stoning simulation for the lower right door corner.

*CONTROL_FORMING_TEMPLATE

Purpose: This keyword is used to simplify the required input for sheet metal stamping simulations. With this keyword, five templates are given: three-piece air draw, three-piece toggle draw, four-piece stretch draw, trimming, and springback.

> **NOTE:** This option has been deprecated in favor of *CONTROL_FORMING_AUTOPOSITION_PARAMETER.

Card 1	1	2	3	4	5	6	7	8
Variable	IDTEMP	BLKID	DIEID	PNCH	BNDU	BNDL	TYPE	PREBD
Type	I	I	I	I	I	I	I	F
Default	none	none	none	none	none	none	0	0.0
Remarks	1	2						

Card 2	1	2	3	4	5	6	7	8
Variable	LCSS	AL/FE	R00	R45	R90	E	DENSITY	PR
Type	I	C	F	F	F	F	F	F
Default	none	Fe	1.0	R00	R00	none	none	none

Card 3	1	2	3	4	5	6	7	8
Variable	K	N	MTYP	UNIT	THICK	GAP	FS	
Type	F	F	I	I	F	F	F	
Default	none	none	37	1	none	1.1t	0.1	

Card 4	1	2	3	4	5	6	7	8
Variable	PATERN	VMAX	VX	VY	VZ	VID	AMAX	
Type	I	F	F	F	F	I	F	
Default	1	1000	0	0	-1	none	1.0e+6	

Card 5	1	2	3	4	5	6	7	8
Variable	LVLADA	SIZEADA	TIMSADA	D3PLT				
Type	I	F	I	I				
Default	1	none	20	10				

VARIABLE	DESCRIPTION
IDTEMP	Type of forming process: EQ.1: 3-piece air-draw (Figure 12-55) EQ.2: 3-piece Toggle-draw (Figure 12-56) EQ.3: 4-piece stretch draw (Figure 12-57) EQ.4: Springback EQ.5: Trimming
BLKID	Part or part set ID (see TYPE) that defines the blank.
DIEID	Part or part set ID that defines the die. See Figures 12-55, 12-56 and 12-57 for more information
PNCHID	Part or part set ID that defines the punch.
BNDUID	Part or part set ID that defines the upper binder.
BNDLID	Part or part set ID that defines the lower binder.

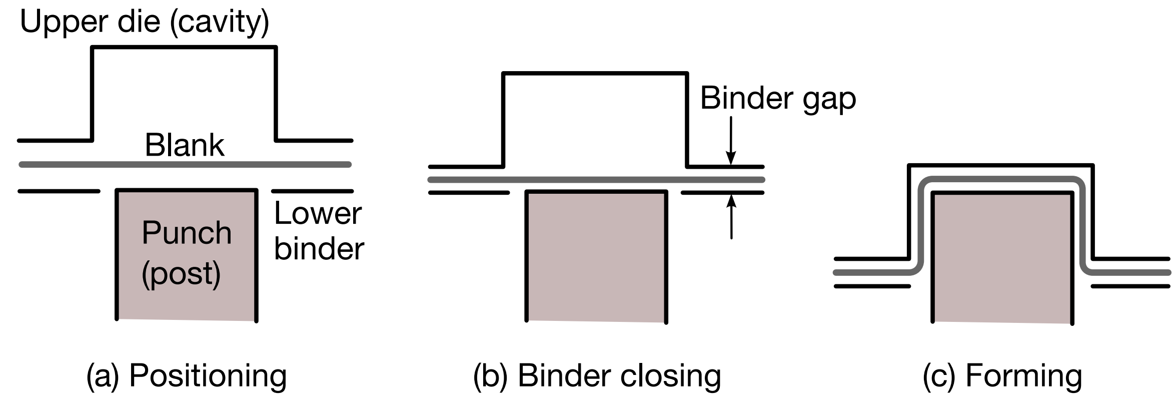


Figure 12-55. IDTEMP = 1: forming in 3-piece air draw.

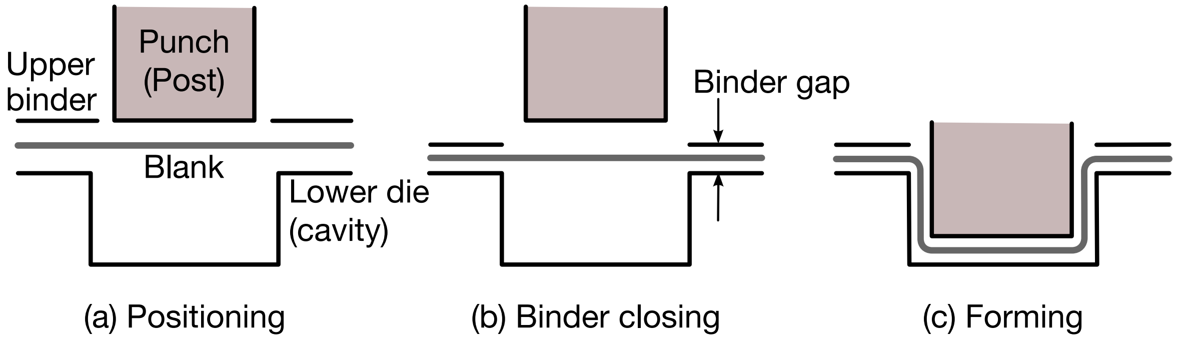
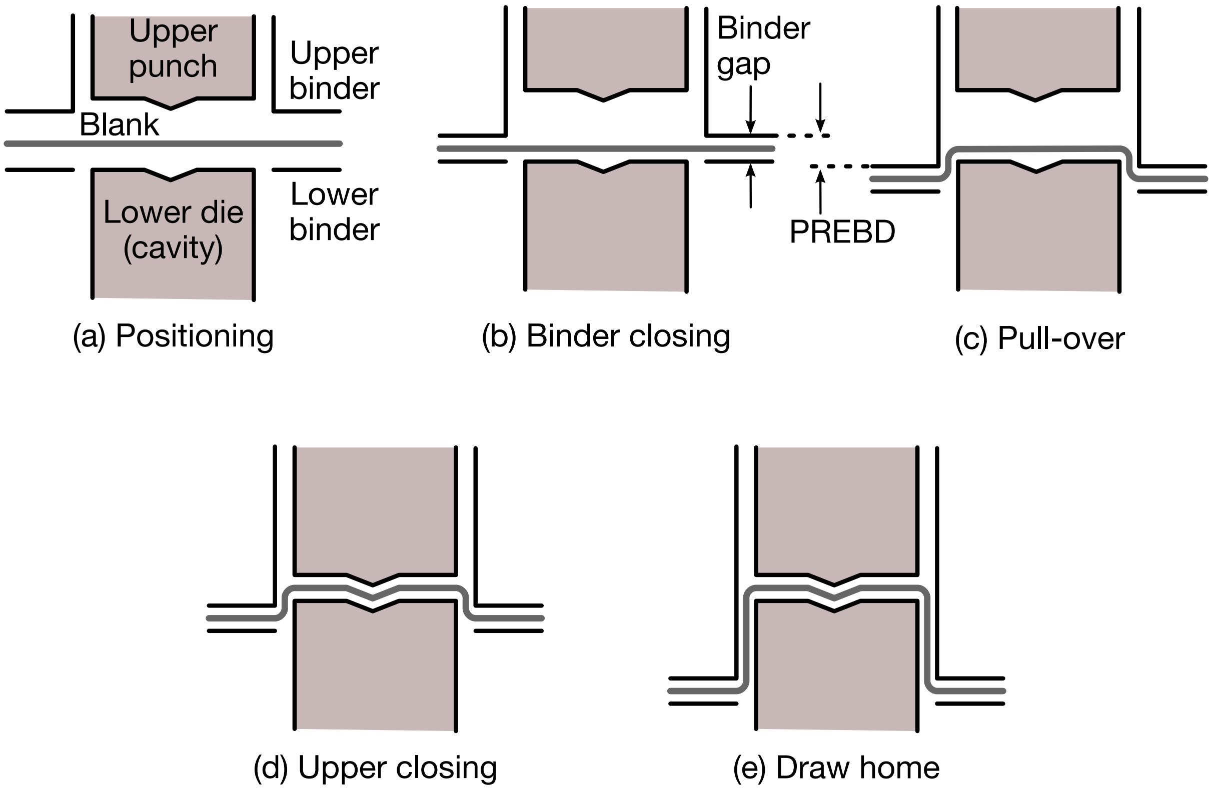


Figure 12-56. IDTEMP = 2: forming in 3-piece toggle draw.

VARIABLE	DESCRIPTION
TYPE	Flag for part or part set ID used in the definition of BLKID, DIEID, PNCHID, BNDUID, and BNDLID: EQ.0: Part ID EQ.1: Part set ID
PREBD	"Pull-over" distance, for 4 piece stretch draw only. This is the travel distance of both upper and lower binder together after they are fully closed. Typically this distance is below 50mm. See Figure 12-57 for more information.
LCSS	If the material (*MAT_XXX) for the blank is not defined, this curve ID will define the stress-strain relationship; otherwise, this curve is ignored.



(a) Positioning (b) Binder closing (c) Pull-over

(d) Upper closing (e) Draw home

Figure 12-57. IDTEMP = 3: forming in 4-piece stretch draw.

VARIABLE	DESCRIPTION
AL/FE	This parameter is used to define the Young's Modulus and density of the blank. If this parameter is defined, E and DENSITY will be defined in the units given by Table 12-58. EQ.A: the blank is aluminum EQ.F: the blank is steel (default)
R00, R45, R90	Material anisotropic parameters. For transversely anisotropy the R value is set to the average value of R00, R45, and R90.
E	Young's Modulus. If AL/FE is user defined, E is unnecessary
DENSITY	Material density of blank. If AL/FE is user defined, this parameter is unnecessary
PR	Poisson's ratio.
K	Strength coefficient for exponential hardening ($\bar{\sigma} = k\bar{\varepsilon}^n$). If LCSS is defined, or if a blank material is user defined by *MAT_XXX, this parameter is ignored.

VARIABLE **DESCRIPTION**

N

Exponent for exponential hardening ($\overline{\sigma} = k\bar{\varepsilon}^{n}$). If LCSS is defined, or if a blank material user defined, this parameter is ignored.

MTYP

Only material models *MAT_036 and *MAT_037 are supported.

UNIT

Define a number between 1 and 10 (Table 12-58) to indicate the UNIT used in this simulation. This unit is used to obtain proper punch velocity, acceleration, time step, and material properties.

THICK

Blank thickness. If the blank thickness is already defined with *SECTION_SHELL, this parameter is ignored.

GAP

The gap between rigid tools at their home position. If *BOUNDARY_PRESCRIBED_RIGID_BODY is user defined, this parameter is ignored. The default is 1.1 x blank thickness.

FS

Friction coefficient (default = 0.10). If the contact (*CONTACT) is user defined, this parameter is ignored.

PATERN

Velocity profile of moving tool. If the velocity is user defined by *BOUNDARY_PRESCRIBED_RIGID_BODY, PATERN is ignored.

EQ.1: Ramped velocity profile

EQ.2: Smooth velocity curve

VX, VY, VZ

Vector components defining the direction of the punch movement. The default direction is defined by VID.

VID

Vector ID defining the direction of the punch movement. This variable overrides the vector components (VX, VY, VZ). If VID and (VX, VY, VZ) are undefined, the punch is assumed to move in the negative z-direction.

AMAX

The maximum allowable acceleration.

LVLADA

Maximum adaptive level.

SIZEADA

Minimum element size permitted in the adaptive mesh.

TIMSADA

Total number of adaptive steps during the forming simulation.

D3PLT

The total number of output states in the D3PLOT database.

Applicable units:

UNIT	1	2	3	4	5	6	7	8	9	10
Mass	Ton	Gm	Gm	Gm	Gm	Kg	Kg	Kg	Kg	Kg
Length	Mm	Mm	Mm	Cm	Cm	Mm	Cm	Cm	Cm	m
Time	S	Ms	S	Us	S	Ms	Us	Ms	S	S
Force	N	N	μN	1e7N	Dyne	KN	1e10N	1e4N	1e-2N	N

Table 12-58. Available units for metal stamping simulation.

About IDTEMP:

When the variable IDTEMP is set to 1, it represents a 3-piece draw in air, as shown in Figure 12-55. When IDTEMP is set to 2, a 3-piece toggle draw is assumed, Figure 12-56. For IDTEMP of 1 or 2, LS-DYNA will automatically position the tools and minimize the punch travel (step a), calculate the binder and punch travel based on the blank thickness and the home gap (step b), set the termination time based on step (a) and (b), define the rigid body motion of the tooling, establish all the contacts between the blank and rigid tools, and, select all necessary control parameters.

When IDTEMP is set to 3, a 4-piece stretch draw shown in Figure 12-57 will be followed. The die action goes as follows: after upper binder moves down to fully close with lower binder, both pieces move together down a certain distance (usually ~50mm) to “pull” the blank “over” the lower die, then upper punch closes with the lower die, finally the binders move down together to their home position.

Both toggle draw and 4-piece stretch draw are called “double action” processes which suffer from a slower stamping speed. As the metric of “hits per minute” (or “parts per minute”) becomes a stamping industry benchmark for efficiency, these types of draw are becoming less popular (especially the 4-piece stretch draw). Nevertheless, they remain important stamping processes for controlling wrinkles in difficult-to-form panels such as lift gate inners, door inners and floor pans. These two processes are also used in situation where deep drawn panels require draw depth of over 250mm, the usual limit for automatic transfer presses.

For all the above IDTEMP values, users do not need to define additional keywords, such as *PART, *CONTROL, *SECTION, *MAT_…, *CONTACT_… (drawbead definition is an exception), and, *BOUNDARY_PRESCRIPTION_RIGID, etc. If any such keyword is defined, automatic default settings will be overridden.

When IDTEMP is set to 4, springback Simulation will be conducted. The only additional keyword, *BOUNDARY_SPC_… is needed to specify the constraints in the input deck.

When IDTEMP is set to 5, a trimming operation will be performed. The only additional keyword, *DEFINE_CURVE_TRIM, is needed to specify the trim curves in the input deck.

Revision information:

This feature is available starting in Revision 45901 and later releases.

*CONTROL_FORMING_TIPPING

Purpose: This keyword is developed to reorient or reposition a part between the stamping dies. In stamping line die simulation, panel tipping and translation between the die stations are frequently required. Typically such transformation involves only a small amount of rotations, e.g. < 15 degrees; and some large amounts of translation. For example, there could be a tipping angle of 10 degree along Y-axis and a translation of 2000 mm along the X-axis between the current trimming die and next flanging die.

Card Set. For each rotated or translated part or set add a Tipping Card plus NMOVE Move Cards. The data set for this keyword ends at the next keyword ("*") card.

Tipping Card. Specify a part or set ID to be tipped.

Card 1	1	2	3	4	5	6	7	8
Variable	PID/SID	ITYPE	ISTRAIN	IFSTRSS	NMOVE			
Type	I	I	I	I	I			
Default	none	none	0	0	0			

Move Card (Rot). Format when first entry, ROT/TRAN, is set to 1.

Card 2	1	2	3	4	5	6	7	8
Variable	ROT/TRAN	V11	V12	V13	X01	Y01	Z01	DISTA1
Type	I	F	F	F	F	F	F	F
Default	none	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Move Card (Trans). Format when first entry, ROT/TRAN, is set to 2.

Card 2	1	2	3	4	5	6	7	8
Variable	ROT/TRAN	DX	DY	DZ				
Type	I	F	F	F				
Default	none	0.0	0.0	0.0				

VARIABLE	DESCRIPTION
PID/SID	Part ID or part set ID of part(s) that requires tipping and/or translation.
ITYPE	Part ID or part set ID indicator: EQ.1: PID means part ID, EQ.2: PID/SID means part set ID.
ISTRAIN	Strain tensors inclusion option: EQ.1: include in tipping/translation.
ISTRESS	Stress tensors inclusion option: EQ.1: include in tipping/translation.
NMOVE	Total number of tipping and translation intended with this keyword.
ROT/TRAN	Transformation type: EQ.1: rotation, EQ.2: translation.
V11, V12, V13	Vector components of an axis about which tipping is performed.
X01, Y01, Z01	X, Y and Z coordinates of a point through which the tipping axis passes.
DSITA	Tipping angle in degree.
DX, DY, DZ	Translation distances along global X-axis, Y-axis and Z-axis.

Remarks:

1. Keyword *INCLUDE can be used to include the file to be tipped or translated.

2. Tipping angle DISTA1 is defined in degree. Signs of the tipping angles follow the "right hand rule".

3. An example of the keyword is included below, to tip a part +23.0 degrees, -31.0 degrees, and +8.0 degrees about X-, Y-, and Z-axis, respectively and passing through the origin; and to translate the part 12.0mm, -6.0mm and 91.0mm along X, Y, and Z axis, respectively.

```
*INCLUDE
trimmedpart.dynain
*CONTROL_FORMING_TIPPING
$ PID/PSID     ITYPE   ISTRAIN    ISTRSS     NMOVE
         1         0         1         1         4
$ ROT/TRAN       V11       V12       V13       X01       y01       z01    DSITA1
         1     1.000  0.000000     0.000     0.000     0.000     0.000      23.0
$ ROT/TRAN       V21       V22       V23       X21       y21       z21    DSITA2
         1     0.000  1.000000     0.000     0.000     0.000     0.000     -31.0
$ ROT/TRAN       V31       V32       V33       X31       y31       z31    DSITA3
         1  0.000000     0.000     1.000     0.000     0.000     0.000       8.0
$ ROT/TRAN        DX        DY        DZ
         2     12.0      -6.0      91.0
```

Revision Information:

This feature is available starting in LS-DYNA Revision 53448, with major updates from Revision 80261. It is also available in *LS-PrePost4.0* eZSetup for metal forming application (http://ftp.lstc.com/anonymous/outgoing/lsprepost/4.0/metalforming/).

*CONTROL_FORMING_TOLERANC

Purpose: This keyword utilizes a smoothing algorithm to reduce the output noise of the strain ratio β (minor strain/major strain) in calculating the Formability Index (F.I.), which predicts sheet metal failure under nonlinear strain paths frequently occurred in metal forming application. This keyword must be used together with the NLP option in and for *MAT_036 and *MAT_037 only; and applies to shell elements only. This feature is jointly developed with the Ford Motor Company.

Card 1	1	2	3	4	5	6	7	8
Variable	DT/CYCLE	WEIGHT	OUTPUT					
Type	F	F	I					
Default	none	none	0					

VARIABLE	DESCRIPTION
DT/CYCLE	Flag for output option (time interval or cycle number). DT/CYCLE.LT.0: The absolute value is the time interval between outputs. DT/CYCLE.GT.0: Cycle numbers between outputs.
WEIGHT	Coefficient α in equation below.
OUTPUT	Output flag. When OUTPUT is set to 1, information such as integration point, element ID, time, strain ratio β, major and minor strains will be output to the ".o" file (a scratch file from batch queue run).

Remarks:

The incremental change of in-plane major and minor strains are smoothed according to the following formula:

$$\Delta\epsilon 1_{(n-1)} * (1-\alpha) + d\epsilon 1_{(n)} * \alpha$$
$$\Delta\epsilon 2_{(n-1)} * (1-\alpha) + d\epsilon 2_{(n)} * \alpha$$

where, $d\epsilon 1_{(n)}$ and $d\epsilon 2_{(n)}$ are incremental changes of $\epsilon 1$ and $\epsilon 2$ in the current time step n, $\Delta\epsilon 1_{(n-1)}$ and $\Delta\epsilon 2_{(n-1)}$ are incremental changes of $\epsilon 1$ and $\epsilon 2$ in the previous time step *n-1*. The weighting coefficient α regulates the smoothness of the incremental changes in $\epsilon 1$ and $\epsilon 2$.

Strain ratio β results from smoothed incremental major and minor strains and stored in history variable #2 along with additional information (see Table 12-2 for details) in ".o" file if running in a batch queueing system, or directly dumped onto the screen if running in an interactive window.

$$\beta = \frac{\Delta\epsilon 2_{(n-1)} * (1-\alpha) + d\epsilon 2_{(n)} * \alpha}{\Delta\epsilon 1_{(n-1)} * (1-\alpha) + d\epsilon 1_{(n)} * \alpha}$$

The upper limit of β is set at 1.0 while the lower limit is:

$$-\frac{\bar{r}}{1+\bar{r}}$$

where $\bar{r}$ is the anisotropic parameter:

$$\bar{r} = \frac{r_0 + 2r_{45} + r_{90}}{4}$$

where r_0, r_{45} and r_{90} are Lankford parameter in the rolling, diagonal and transverse direction, respectively.

The keyword usage is shown in the following partial input deck, where *MAT_3-PARAMETER_BARLAT_NLP is used. Note NEIPS is set at 3 for output of 3 history variables that include formability index (F.I.), strain ratio β and effective plastic strain $\bar{\varepsilon}$.

```
*KEYWORD
*INCLUDE_TRIM
sim_trimming.dynain
⋮
*DATABASE_EXTENT_BINARY
$    NEIPH     NEIPS    MAXINT    STRFLG    SIGFLG    EPSFLG    RLTFLG    ENGFLG
                   3      &nip         1
⋮
*PARAMETER_EXPRESSION
R d3plot   endtime/1000.0
R nt       -1.0*d3plot
...
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*CONTROL_FORMING_TOLERANC
$ DT/CYCLE    WEIGHT    OUTPUT
       &nt      0.15         1
*MAT_3-PARAMETER_BARLAT_NLP
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$      MID        RO         E        PR        HR
        13   7.8E-09  2.07E+05      0.30     3.000
$        M       R00       R45       R90      LCID
     6.000     1.200     1.450     1.090        99
$     AOPT         C         P     VLCID                                  NLP
     2.000                                                                200
$                                      A1        A2        A3
                                    1.000     0.000     0.000
$       V1        V2        V3         D1        D2        D3
                                    0.000     1.000     0.000
*DEFINE_CURVE
      200
$FORM LIMIT DIAGRAM
             -0.7000              0.8309
             -0.4500              0.6805
             -0.2500              0.5081
```

```
              0.0000              0.2479
              0.2000              0.3487
              0.4000              0.3845
*DEFINE_CURVE
        99
     0.000000000E+00     0.166100006E+03
     0.579999993E-02     0.189110001E+03
     0.124000004E-01     0.204789993E+03
     0.190999992E-01     0.218520004E+03
     0.255999994E-01     0.230580002E+03
            ⋮                    ⋮
     0.100000000E+01     0.512053406E+03
...
*END
```

As shown in Figure 12-59, beta smoothed using smoothing algorithm is much better than unsmoothed one. Most importantly, a plot of strain path (Figure 12-60) in the traditional FLD space ($\epsilon1$ vs. $\epsilon2$) confirms the terminal beta is approximately 0.9, which is much closer to the smoothed beta value (Figure 12-59) at the end of the simulation.

Output items	IP #	Element ID	Time	β	$\epsilon1$	$\epsilon2$
Columns	1st to 8th	9th to18th	19th to 29th	30th to 40th	41th to 51th	52th to 62th

Table 12-2. ".o" file output information and positions. Note only the mid-IP information are output.

Revision Information:

Revision history information is listed below. Output information in ".o" file currently applies to SMP and 1 CPU MPP only.

1)LS-DYNA Revision 84159: β smoothing is enabled for *MAT_036.

2)Revision 110928: β smoothing is enabled for *MAT_037.

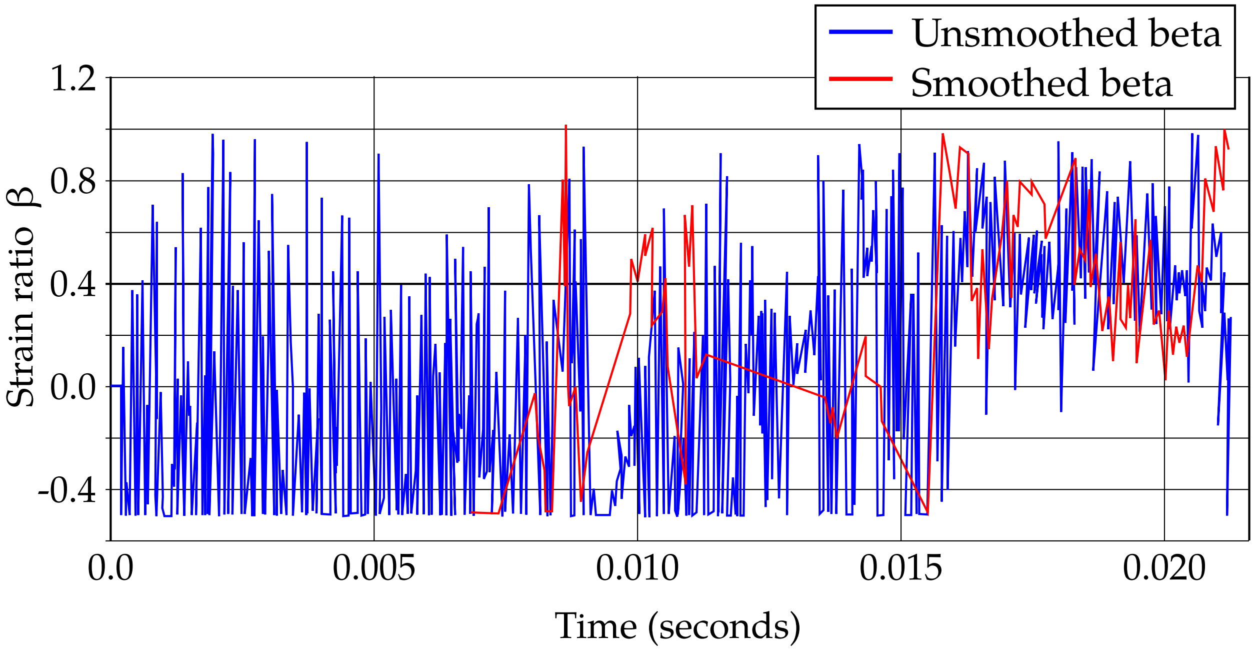


Figure 12-59. Effect of smoothing on strain ration β.

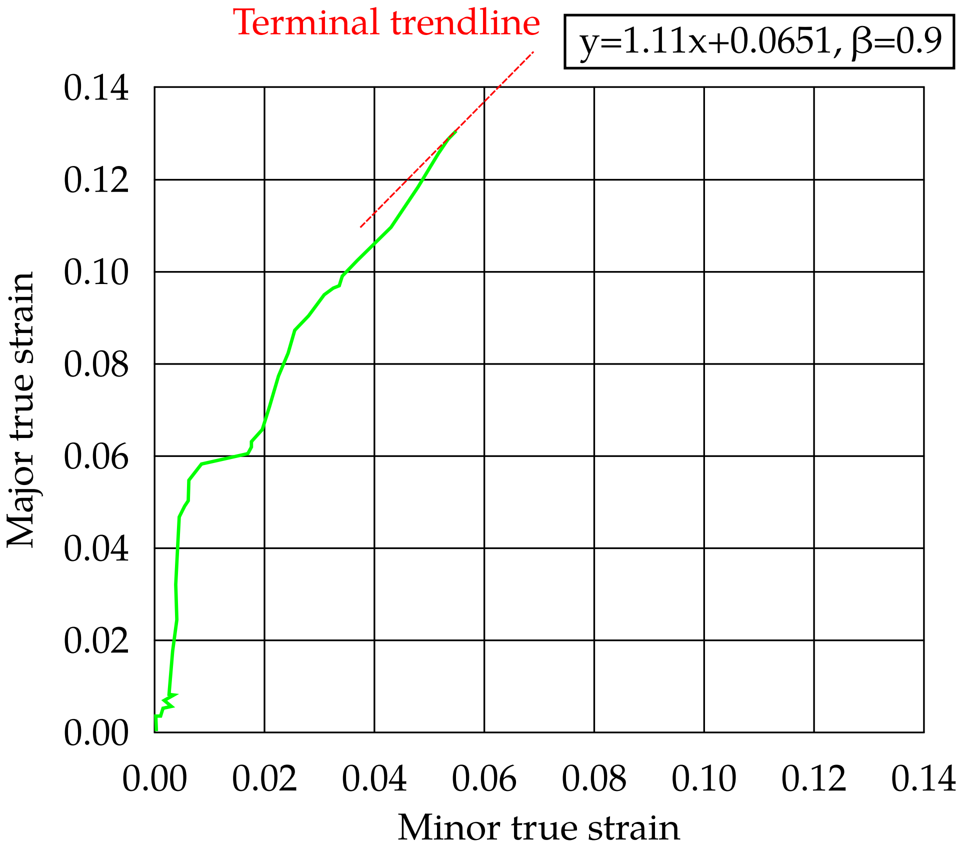


Figure 12-60. Strain path and terminal strain ratio β value.

*CONTROL_FORMING_TRAVEL

Purpose: This keyword allows user to define tool travel for each phase in a stamping process simulation. The entire simulation process can be divided into multiple phases corresponding to the steps of an actual metal forming process. This keyword is to be used for tools that are pre-positioned above the sheet blank (or below the blank) and ready for forming. For tools that are pre-positioned at their home positions, *CONTROL_FORMING_TRAVEL should be used. This keyword is used together with *CONTROL_FORMING_USER.

> **NOTE:** This option has been deprecated in favor of *CONTROL_FORMING_AUTOPOSITION_PARAMETER).

Define Travel Cards. Repeat Card as many times as needed to define travels in multiple phases. The next "*" card terminates the input.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	VID	TRAVEL	TARGET	GAP	PHASE	FOLLOW	
Type	I	I	F	I	F	I	I	
Default	none	none	none	none	1.0t	none	none	

VARIABLE	DESCRIPTION
PID	Part ID of a stamping tool, as defined in *PART.
VID	Vector ID defining the direction of travel for the tool defined by the PID.
TRAVEL	The distance in which the tool will be traveled to complete forming in the direction specified by the VID. If TRAVEL is defined, it is unnecessary to define TARGET.
TARGET	Target tool PID, as defined in *PART. The tool (defined by PID) will be traveled to where the TARGET tool is to complete forming.
GAP	The minimum distance between the tool (PID) and TARGET tool at the home position (forming complete). The GAP is by default the sheet blank thickness "t".

VARIABLE	DESCRIPTION
PHASE	Phase number, starting sequentially from 1. For example, phase 1 is the binder closing, and phase 2 is the drawing operation.
FOLLOW	Part ID of a stamping tool to be followed by the tool (PID). When this variable is defined, the distance between the tool (PID) and part ID defined by FOLLOW, will remain constant during the phase.

Remarks:

FOLLOW can be used to reduce total simulation time. For example, in a toggle draw, the upper punch travels together with the upper binder during binder closing phase, thus reducing the upper travel distance during the draw, shortening the overall termination time.

An example is provided in manual pages under *CONTROL_FORMING_USER.

*CONTROL_FORMING_TRIM_MERGE

Purpose: This feature allows for automatic close of any open trim loop curve for a successful trimming simulation. Previously, sheet metal trimming would fail if a trim curve does not form a closed loop. This keyword is used together with *DEFINE_-CURVE_TRIM, *ELEMENT_TRIM, *DEFINE_VECTOR, *CONTROL_ADAPTIVE_-CURVE, *CONTROL_CHECK_SHELL, and applies to shell elements only.

Card 1	1	2	3	4	5	6	7	8
Variable	IMERGE	GAPM						
Type	I	F						
Default	1	0.0						

VARIABLE	DESCRIPTION
IMERGE	Activation flag. Set to ‘1’ (default) to activate this feature.
GAPM	Gap distance between two open ends of a trim loop curve in the model. If the gap is smaller than GAPM, the two open ends of a trim curve will be closed automatically.

Remarks:

1. If multiple open trim loop curves exist, GAPM should be set to a value larger than any of the gap distances of any trim curves in the trim model.

2. An example provided below shows that for both 3D (#90905) and 2D trim curve (#90907), each with an open gap of 2.3 and 2.38mm, respectively. An automatic merge operation is being performed with the GAPM set at 2.39 mm. Since this set value is larger than both gaps in the model, trimming will automatically close the gap for both curves and to form two closed-loop curves for a successful trim. In Figure 12-61, two different 2D trimming results are illustrated with GAPM of 2.39 (successful) as well as 2.37 (fail).

```
*KEYWORD
*INCLUDE_TRIM
drawn2.dynain
⋮
*CONTROL_ADAPTIVE_CURVE
$    IDSET     ITYPE         N       SMIN
   &blksid         2         3        0.6
```

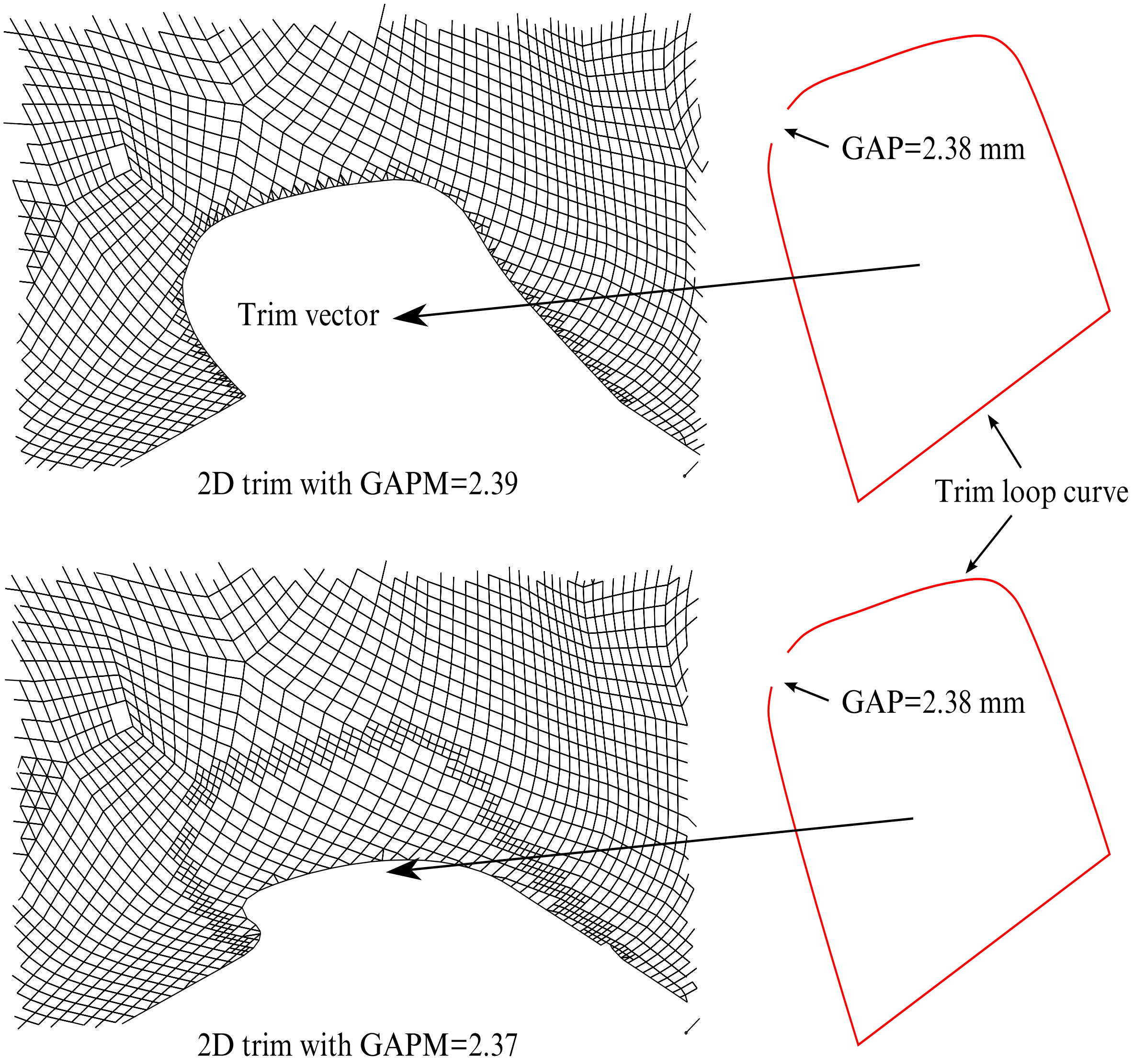


Figure 12-61. A 2D trimming with different GAPM values.

```
*CONTROL_CHECK_SHELL
$      PSID    IFAUTO    CONVEX      ADPT    ARATIO     ANGLE      SMIN
  &blksid1         1         1         1  0.250000150.000000  0.000000
*DEFINE_CURVE_TRIM_3D
$#     tcid    tctype      tflg      tdir     tctol      toln    nseed1
nseed2
     90907         2         1         0  1.250000  2.500000         0
0
sim_trimline_03.igs
*DEFINE_CURVE_TRIM_NEW
$#     tcid    tctype      tflg      tdir     tctol      toln    nseed1
nseed2
     90905         2         0         2  1.250000  1.000000         0
0
$# filename
sim_trimline_02.igs
*DEFINE_VECTOR_TITLE
vector for Trim curve 90905
$#      vid        xt        yt        zt        xh        yh        zh
cid
```

```
         2     0.000     0.000     0.000 -0.170000  0.950000 -0.260000
0
*ELEMENT_TRIM
&blksid
*DEFINE_TRIM_SEED_POINT_COORDINATES
$ NSEED,X1,Y1,Z1,X2,Y2,Z2
1,&seedx,&seedy,&seedz
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+---
-8
*CONTROL_FORMING_TRIM_MERGE
$   IMERGE      GAPM
         1      2.39
$ Note that the 3D trim curve has a gap of 2.3 and the 2D trim curve has a gap
of 2.38
*END
```

3. This feature is available starting in LS-DYNA Revision 84098.

*CONTROL_FORMING_TRIMMING

Purpose: Define a part subset to be trimmed by *DEFINE_CURVE_TRIM. This feature is intended for metal forming simulation. Currently trimming is enabled on 2D and 3D trimming of shell elements, 3D solid element, adaptive sandwiched parts (one layer of solid elements with top and bottom layers of shell elements), non-adaptive sandwiched parts (multiple layers of solid elements with top and bottom layers of shell elements), and 2-D trimming of thick shell elements (TSHELL). Note it is not applicable to axisymmetric solids or 2D plane strain/stress elements. For details, see *DEFINE_-CURVE_TRIM.

NOTE: Before revision 87566 this card was called ELEMENT_TRIM.

Card 1	1	2	3	4	5	6	7	8
Variable	PSID		ITYP					
Type	I		I					
Default	none		0					

VARIABLE	DESCRIPTION
PSID	Part set ID for trimming, see *SET_PART.
ITYP	Activation flag for sandwiched parts (laminates) trimming: ITYP.EQ.0: Trimming for solid elements. ITYP.EQ.1: Trimming for laminates.

Remarks:

This keyword is used together with *DEFINE_CURVE_TRIM to trim the parts defined in PSID at time zero, i.e., before any stamping process simulation begins. Elements in the part set will be automatically trimmed in the defined direction if they intersect the trim curves. See examples in keyword section *DEFINE_CURVE_TRIM.

Revision information:

1. Revision 87566: *ELEMENT_TRIM was changed to the current name *CONTROL_FORMING_TRIMING.

2. Revision 95745: *CONTROL_FORMING_TRIMING was changed to *CONTROL_FORMING_TRIMMING.

3. Revision 92088: 2-D trimming of solid elements is implemented.

4. Revision 92289: 2-D and 3-D trimming of laminates (ITYP) is added.

5. Revision 93467: 3-D trimming of solid elements is added.

6. Latter Revisions may incorporate more improvements and are suggested to be used for trimming.

CONTROL_FORMING_UNFLANGING_*{OPTION}

Available options include:

<BLANK>

OUTPUT

Purpose: The keyword unfolds flanges of a deformable blank onto a rigid tooling mesh using an implicit statics solver. This is typically used in trim line unfolding during a stamping die development process. The option OUTPUT must be used together with *CONTROL_FORMING_UNFLANGING to get the modified trim curves. Other keywords related to blank size development are, *CONTROL_FORMING_ONESTEP, and *INTERFACE_BLANKSIZE_DEVELOPMENT.

Card 1 for no option, <BLANK>:

Card 1	1	2	3	4	5	6	7	8
Variable	NOPTION	DVID	NUNBEND	STFBEND	STFCNT	IFLIMIT	DIST	ILINEAR
Type	I	I	I	F	F	I	F	I
Default	none	N/A	none	none	none	none	none	2

Card 2 for no option, <BLANK>:

Card 1	1	2	3	4	5	6	7	8
Variable	NB1	NB2	NB3	CHARLEN	NDOUTER			
Type	I	I	I	F	I			
Default	none	none	none	150.0	none			

Card 1 for option OUTPUT:

Card 1	1	2	3	4	5	6	7	8
Variable	THMX	THMN	EPSMX					
Type	I	I	I					
Default	10^{20}	0.0	10^{20}					

VARIABLE	DESCRIPTION
NOPTION	Flag to turn on an unfolding simulation: EQ.1: Activate the unfolding simulation program.
DVID	This variable is currently not being used.
NUNBEND	Estimated number of unbending, ranging from 10 to 100.
STFBEND	Unflanging stiffness, ranging from 0.1 to 10.0.
STFCNT	Normal stiffness, ranging from 0.1 to 10.0.
IFLIMIT	Iteration limit for the first phase of unfolding, typically ranging from 11 to 400.
DIST	Distance tolerance for auto-SPC along flange root. DIST (Figure 12-63) is usually slightly more than ½ of the flange thickness. *This field must be left blank for ILINEAR = 2.* Also, nodes along the root can be directly positioned on the rigid body surface (addendum), leaving a DIST of zero (Figure 12-63).
ILINEAR	Unfolding algorithm selection flag: EQ.0: Nonlinear unfolding. EQ.1: Linear unfolding. EQ.2: A hybrid unfolding method (Revision 87100 and later). The curved 3D meshes of the flange will first be mapped onto the tooling surface to be used as a starting porting for nonlinear iterations; unfolding completes when force balance is reached. (recommended).

VARIABLE	DESCRIPTION
NB1	The start node ID (Figure 12-64) on a flange root boundary (fixed end of the flange, see Figures 12-63 and 12-64). For closed-loop flange root boundary, only this parameter needs to be defined; for open-loop flange root boundary, define this parameter as well as NB2 and NB3. The solver will automatically identify and automatically impose the necessary boundary constraints on all the nodes along the entire three-dimensional flange root boundary.
NB2	The ID of a node in the middle of the flange root boundary, see Figure 12-64. Define this parameter for open-loop flange root boundary only.
NB3	The end node ID on a flange root boundary. Define this parameter for open-loop flange root boundary only. The "path" formed by NB1, NB2 and NB3 can be in any direction, meaning NB1 and NB3 (Figure 12-64) can be interchangeable.
CHARLEN	Maximum flange height (Figure 12-64) to limit the search region for the boundary nodes along the flange root. This value should be set bigger than the longest width (height) of the flange; and it is needed in some cases. This parameter is now automatically calculated as of Revision 92860.
NDOUTER	A node ID on the outer flange (free end of the flange) boundary. This node helps search of nodes along the flange root, especially when holes are present in the flange area, see Figure 12-64.
THMX	Maximum thickness beyond which elements are deleted; this is useful in removing wrinkling areas of the flange (shrink flange). Modified, unfolded flange outlines based on this parameter are stored in a file called "trimcurve_upd.k", written using the *DEFINE_CURVE_TRIM_3D; keyword. The modified flanges (before unfolding) are in a keyword file called "mdfiedflangedpart.k"; and the unmodified flange (unfolded) is in "trimcurve_nmd.k", also written using keyword *DEFINE_CURVE_TRIM_3D. See the example in Figure 12-64 for an explanation. Currently the modified flange and curves are not smooth, which will be improved in the future. To convert between *DEFINE_CURVE_TRIM_3D and IGES format, refer to Figures in *INTERFACE_BLANKSIZE.

VARIABLE	DESCRIPTION
THMN	Minimum thickness below which elements are deleted; this is useful in removing overly thinned areas of the flange (stretch flange). Updated flange information based on this parameter is stored in files listed above.
EPSMX	Maximum effective plastic strain beyond which elements are deleted; this is useful in removing flange areas with high effective plastic strains (stretch flange). Updated flange information based on this parameter is stored in files listed above.

Introduction:

Unfolding of flanges is one of the first steps in a stamping die development process. Immediately after tipping, binder and addendums are built for unfolding of flanges. According to process considerations (trim conditions, draw depth, and material utilization, etc.), the addendums are built either in parallel or perpendicular to the draw die axis, tangentially off the main surface off the breakline (see Appendix T), or any combinations of the three scenarios. Trim lines are developed by unfolding the flanges in finished (hemmed or flanged) position onto these addendums. Addendum length in some areas may have to be adjusted to accommodate the unfolded trim lines. Trim line development is very critical in hard tool development. Inaccurate trim lines lead to trim die rework, result in many hours of re-welding, re-machining and re-spotting of trim die components.

Input and output:

The inputs for the keyword are:

1. blank or flanges in the finished configuration, and,
2. the draw die surface in mesh.

Meshes for flanges should of a quality similar to the blank mesh one would build for a forming simulation. In *LS-PrePost 4.0*, this kind of mesh can be created using *Mesh → Automesh → Size*. Element formulation 16 with NIP set to 5 is recommended for the blank. The output results, in terms of unfolding steps and final unfolded flanges, are stored in the d3plot files. *LS-PrePost 4.0* function of *Curve → Spline → From Mesh → By part* can be used to create unfolded flange/trim curves from the unfolded flanges. Since the program uses an implicit statics solver, the double precision version of LS-DYNA must be used.

Other modeling guidelines:

1. All addendum and flanges need to be oriented as if they are in a draw position, with the drawing axis parallel to the global Z-axis; specifically the flanges need to be on top of the addendum, as noted in Figures 12-62, 12-63 and 12-64.

2. Normals of the to-be-unfolded flange side and tool surface side must be consistent and must face against each other when the flange is unfolded, see Figure 12-64.

3. Holes in the blank are allowed only for ILINEAR = 2.

4. Adaptive re-meshing is not supported.

5. To-be-unfolded flange and tool meshes must not share the same nodes. This can be easily done using the mesh detaching feature under EleTol → DetEle in *LS-PrePost*.

6. Meshes of the flange part and rigid tool can slightly overlap each other, but **large amounts of overlap** (area of flange already on addendum surface) **is not allowed**. In *LS-PrePost* the EleTol → PtTrim feature can trim off the overlapped flange portion. The curves used for the trimming can be obtained from the flange tangent curves on the addendum (which has a more regulated mesh pattern) using *LS-PrePost's* Curve → Spline → Method From Mesh → By Edge → Prop feature with appropriate angle definition. Furthermore, any holes are not allowed in the overlapping area.

7. *CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE should be used for the contact between the blank and tool. Negative tool offsets on the *CONTACT_… keyword is *not* supported.

8. The rigid tool (total fixed in *MAT_020) must be larger than the unfolded flanges, especially along symmetric lines. This may be obvious, nevertheless it is sometimes overlooked.

9. Nodes along the flange root are automatically fixed by defining NB1, NB2 and NB3, as shown in Figure 12-64.

10. No "zigzag" along the flange root boundary, meaning that the boundary along the flange root must be smooth. This restriction is removed as of Revision 92727.

11. Symmetric boundary conditions are supported.

12. Thickness and effective plastic strain are stored in a file "unflanginfo.out", which can be plotted in *LS-PrePost* 4.0, see Figure 12-64.

Application example:

A partial input deck is provided below for flange unfolding of a fender outer, modified from the original NCAC Taurus model. Shown in Figure 12-62 are the progressions of the unfolding process, where the finished flanges are to be unfolded onto the addendum (rigid body). A section view of the same unfolding before and after is found in Figure 12-63. ILINEAR is set at 2 while DIST is left blank. Total numbers of elements are 1251 on the blank and 6600 on the tooling. It took less than 3 minutes on an 8 CPU (SMP) machine. Note that additional keywords, such as *CONTROL_IMPLICIT_FORMING, etc. are used. Termination criterion is set using the variable DELTAU in *CONTROL_IMPLICIT_TERMINATION. Termination is reached when the relative displacement ratio criterion is met, as indicated in the messag file. Termination time of 10.0 (steps) is sufficient for most cases, but may need to be extended in some cases to satisfy the DELTAU in some cases.

```
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*KEYWORD
*INCLUDE
toolblankmesh.k
*CONTROL_FORMING_UNFLANGING
$  NOPTION      DVID   NUNBEND   STFBEND    STFCNT   IFLIMIT      DIST   ILINEAR
         1                 100       0.2      15.0       400                   2
$      NB1       NB2       NB3   CHARLEN   NDOUTER
       321       451       322      60.0      6245
*CONTROL_IMPLICIT_FORMING
1
*CONTROL_IMPLICIT_TERMINATION
$   DELTAU
$ set between 0.0005~0.001
    0.0005
*CONTROL_IMPLICIT_GENERAL
$   IMFLAG       DT0
         1     .1000
*CONTROL_IMPLICIT_SOLUTION
$  NSLOLVR    ILIMIT    MAXREF     DCTOL     ECTOL     RCTOL     LSTOL
         2         2      1100     0.100     1.e20     1.e20
$    dnorm   divflag   inistif
         0         2         0         1                   1
*PARAMETER
R ENDTIME       10.0
I elform          16
I nip              5
R bthick         1.0
*PARAMETER_EXPRESSION
R D3PLOTS    ENDTIME/60.0
*CONTROL_TERMINATION
&ENDTIME
*DATABASE_BINARY_D3PLOT
&D3PLOTS
*CONTROL_RIGID...
*CONTROL_HOURGLASS...
*CONTROL_BULK_VISCOSITY...
*CONTROL_SHELL...
*CONTROL_CONTACT
$   SLSFAC    RWPNAL    ISLCHK    SHLTHK    PENOPT    THKCHG     ORIEN
      0.01       0.0         2         1         4         0         4
$   USRSTR    USRFAC     NSBCS    INTERM     XPENE     SSTHK      ECDT   TIEDPRJ
         0         0        10         0       2.0         0
*CONTROL_ENERGY...
*CONTROL_ACCURACY...
```

```
*DATABASE_EXTENT_BINARY...
*SECTION_SHELL_TITLE
BLANK/FLANGE thickness and elform/nip specs.
&blksec      &elform     0.833      &nip       1.0
&bthick,&bthick,&bthick,&bthick
*PART...
*MAT_TRANSVERSELY_ANISOTROPIC_ELASTIC_PLASTIC...
*MAT_RIGID...
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE
$      SSID      MSID     SSTYP     MSTYP    SBOXID    MBOXID       SPR       MPR
   &blkpid   &diepid         3         3
$        FS        FD        DC        VC       VDC    PENCHK        BT        DT
     0.125       0.0       0.0       0.0      20.0         0       0.0 1.000E+20
$       SFS       SFM       SST       MST      SFST      SFMT       FSF       VSF
       1.0       1.0       0.0
$      SOFT    SOFSCL    LCIDAB    MAXPAR    PENTOL     DEPTH     BSORT    FRCFRQ
         0
$    PENMAX    THKOPT    SHLTHK     SNLOG      ISYM     I2D3D    SLDTHK    SLDSTF

$      IGAP    IGNORE    DPRFAC    DTSTIF                        FLANGL
         2
*END
```

In Figure 12-64 (top), with THMN set at 0.4mm, the stretch flange area of the corner, which has thickness less than 0.4mm, is removed; and the modified flange outlines are created accordingly (bottom). The partial input used is listed below.

```
*CONTROL_FORMING_UNFLANGING
$  NOPTION      DVID   NUNBEND   STFBEND    STFCNT   IFLIMIT      DIST   ILINEAR
         1                 100       0.2      15.0       400                   2
$       NB1       NB2       NB3   CHARLEN   NDOUTER
       321       451       322      60.0      6245
*CONTROL_FORMING_UNFLANGING
$      THMX      THMN     EPSMX
                 0.4
```

Revision information:

The feature is available in double precision SMP, and starting in LS-DYNA Revision 73190. Revision information is listed below for various parameters and features:

1. ILINEAR = 2: Revision 87100.
2. NDOUTER: Revision 87318.
3. CHARLEN: Revision 87210.
4. NB1, NB2, NB3: Revision 87100.
5. Option OUTPUT: Revision 86943.
6. Holes allowed: Revision 87167.
7. File “mdfiedflangedpart”: Revision 87105.

8. Symmetric boundary condition: Revision 88359.
9. CHARLEN automatically calculated: Revision 92860.
10. “Zigzag” flange root boundary allowed: Revision 92727.

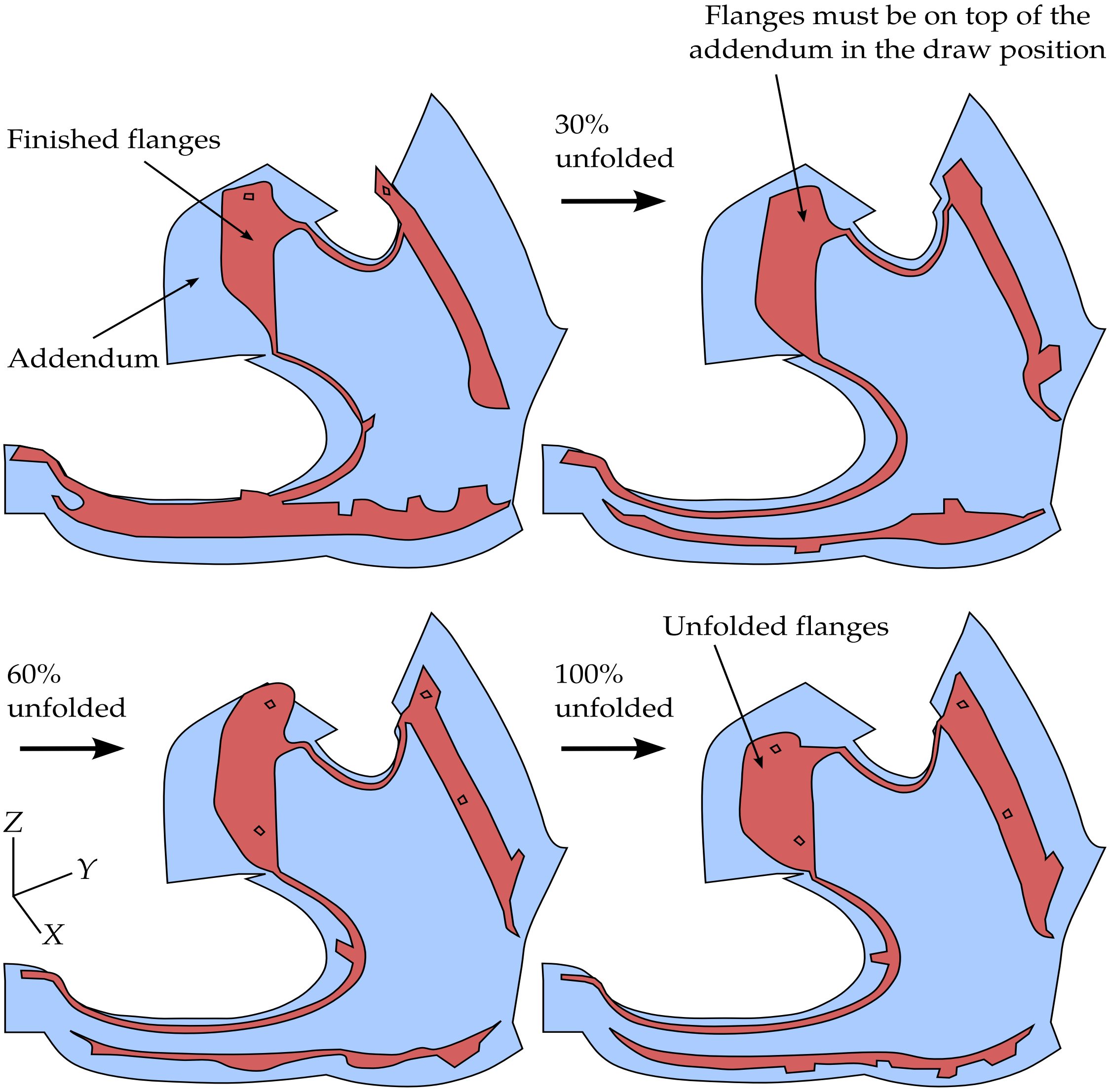


Figure 12-62. Flange unfolding progression of a fender outer (*original model courtesy of NCAC at George Washington University*).

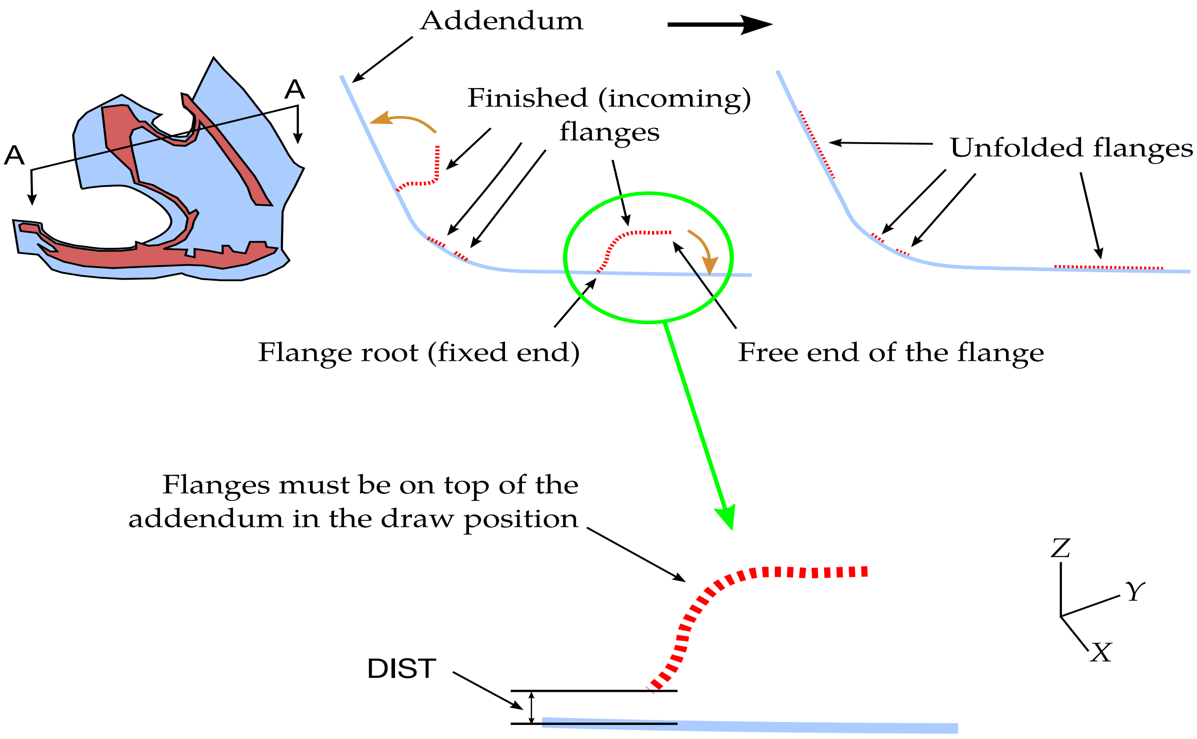
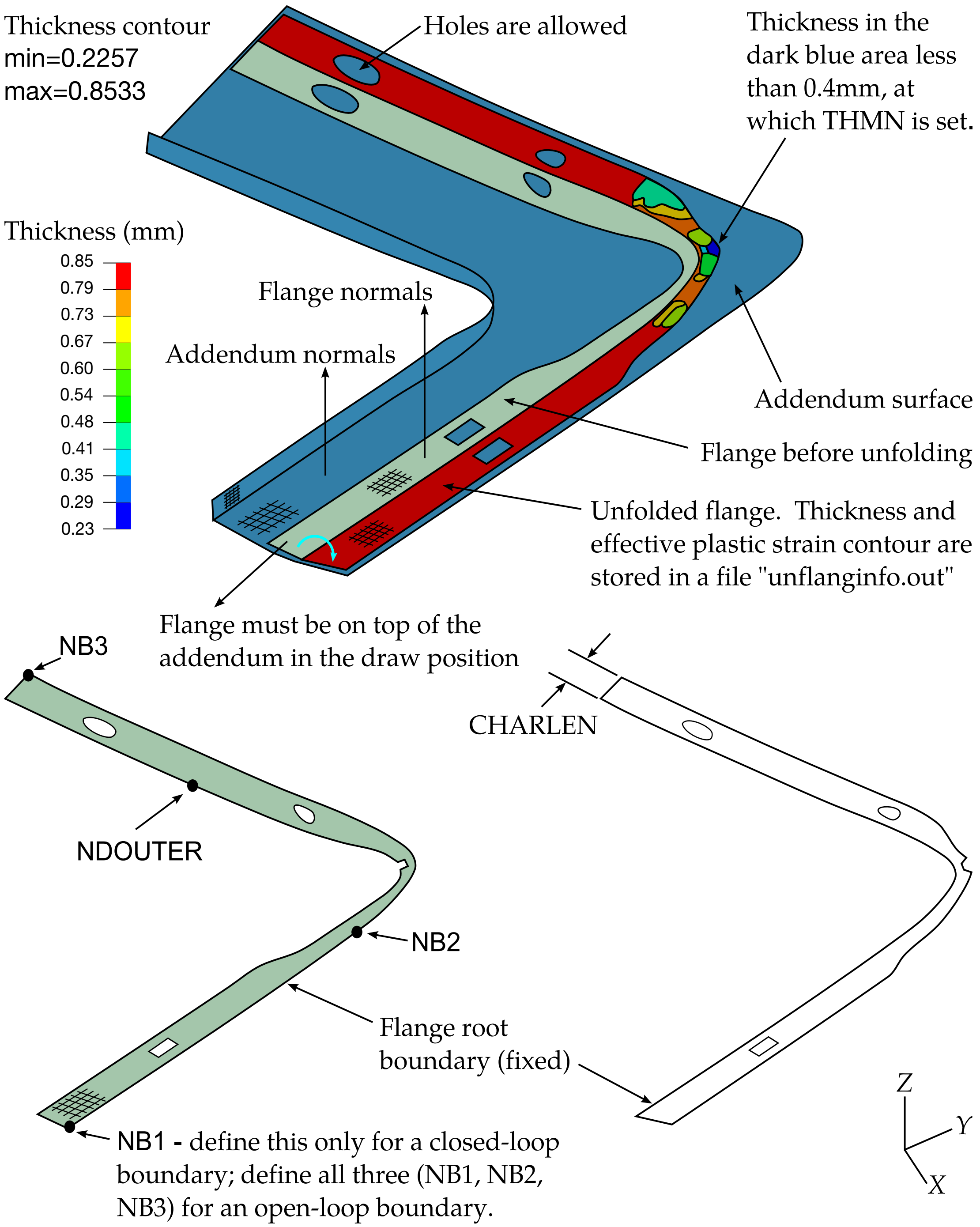


Figure 12-63. A section view showing flange unfolding before and after.



Thickness in the dark blue area less than 0.4mm, at which THMN is set.

Unfolded flange. Thickness and effective plastic strain contour are stored in a file "unflanginfo.out"

Flange must be on top of the addendum in the draw position

NB1 - define this only for a closed-loop boundary; define all three (NB1, NB2, NB3) for an open-loop boundary.

Original flange is modified based on THMN=0.4 and the mesh is stored in a file "mdfiedflangedpart.k". Boundary curves can be created using LSPP4.0 under Curve/Spline/From mesh/by part.

Modified boundary curves on unfolded flange are stored in a file "trimcurve_upd.k"; original boundary curves (without the corner cutout) is in "trimcurve_nmd.k".

Figure 12-64. Unfolding details and output files

*CONTROL_FORMING_USER

Purpose: This keyword, along with *CONTROL_FORMING_POSITION, or *CONTROL_FORMING_TRAVEL, allow user to set up a stamping process simulation. From this card various model parameters may be specified:

- material properties,
- material model,
- tooling kinematics,
- mesh adaptivity
- D3PLOT generation

NOTE: This option has been deprecated in favor of *CONTROL_FORMING_AUTOPOSITION_PARAMETER).

Card 1	1	2	3	4	5	6	7	8
Variable	BLANK	TYPE	THICK	R00	R45	R90	AL/FE	UNIT
Type	I	I	F	F	F	F	A	I
Default	none	0	none	1.0	R00	R00	F	1

Card 2	1	2	3	4	5	6	7	8
Variable	LCSS	K	N	E	DENSITY	PR	FS	MTYPE
Type	I	F	F	F	F	F	F	I
Default	none	none	none	none	none	none	0.1	37

Card 3	1	2	3	4	5	6	7	8
Variable	PATERN	VMAX	AMAX	LVLADA	SIZEADA	ADATIMS	D3PLT	GAP
Type	I	F	F	I	F	I	I	F
Default	1	1000.0	500000.	0	0	0	10	1.1t

VARIABLE	DESCRIPTION
BLANK	PID of a sheet blank, as in *PART.
TYPE	Flag of part or part set ID for the blank: EQ.0: Part ID. EQ.1: Part set ID.
THICK	Thickness of the blank. This variable is ignored if the thickness is already defined in *SECTION_SHELL.
R00, R45, R90	Material anisotropic parameters. For transverse anisotropy the R value is set to the average value of R00, R45, and R90.
AL/FE	This parameter is used to define the Young's Modulus, E, and density, ρ, for the sheet blank. If this variable is defined, E and ρ will be found by using the proper unit, as listed in Table 8.1, under *CONTROL_FORMING_TEMPLATE. EQ.A: the blank is aluminum EQ.F: the blank is steel (default)
UNIT	Units adopted in this simulation. Define a number between 1 and 10. Table 8.1 is used to determine the value for UNIT. This unit is used to obtain proper values for punch velocity, acceleration, time step, and physical and material properties.
LCSS	If the material for the blank has not been defined, this curve will be used to define the stress-strain relation. Otherwise, this variable is ignored.
PREBD	"Pull-over" distance for the upper and lower binders after closing in a 4-piece stretch draw, as shown in Figure 12-57.

VARIABLE	DESCRIPTION
K	Strength coefficient for exponential hardening ($\overline{\sigma} = k\bar{\varepsilon}^n$). If LCSS is defined, or if a blank material is defined with *MAT_036 or *MAT_037, this variable is ignored.
N	Exponent for exponential hardening ($\overline{\sigma} = k\bar{\varepsilon}^n$). If LCSS is defined, or if a blank material is defined with *MAT_036 or *MAT_037, this variable is ignored.
E	Young's Modulus. If AL/FE is user defined, E is unnecessary.
DENSITY	Material density of the blank. If AL/FE is defined, this variable is unnecessary.
PR	Poisson's ratio. If AL/FE is user defined, this variable is unnecessary.
FS	Coulomb friction coefficient. If contact is defined with *CONTACT_FORMING_..., this variable is ignored.
MTYP	Material model identification number, for example, 36 for *MAT_036 and 37 for *MAT_037. Currently only material models 36 and 37 are supported.
PATERN	Velocity profile of the moving tool. If the velocity and the profile are defined by *BOUNDARY_PRESCRIBED_MOTION_RIGID, and *DEFINE_CURVE, this variable is ignored. EQ.1: Ramped velocity profile. EQ.2: Smooth velocity curve.
VMAX	The maximum allowable tool travel velocity.
AMAX	The maximum allowable tool acceleration.
LVLADA	Maximum mesh adaptive level.
SIZEADA	Minimum element size permitted during mesh adaptivity.
ADATIMS	Total number of adaptive steps during the simulation.
D3PLT	The total number of output states in the D3PLOT database.
GAP	Minimum gap between two closing tools at home position, in the travel direction of the moving tool. This variable will be used for *CONTROL_FORMING_POSITION.

Keyword examples:

A partial keyword example provided below is for tools in their home positions in a simple 2-piece crash forming die. A steel sheet blank PID 1, is assigned with a thickness of 0.76mm (UNIT = 1) and *MAT_037 with anisotropic values indicated, to follow hardening curve of 90903, form in a 'ramped' type of velocity profile with maximum velocity of 5000mm/s and acceleration of 500000.0 mm/s2, adapt mesh 5 levels with smallest adapted element size of 0.9 for a total of 20 adaptive steps, create a total of 15 post-processing states, and to finish forming with a final gap of 1.1mm between the tools (PID3 and 5) at home position. The upper tool with PID 3 is to be moved back in Z axis to clear the interference with the blank before close toward the lower tool of target PID 5.

```
*CONTROL_FORMING_USER
$    BLANK      TYPE     THICK       R00       R45       R90     AL/FE      UNIT
         1         0      0.76       1.5       1.6       1.4         F         1
$     LCSS         K         N         E   DENSITY        PR        FS     MTYPE
     90903                                                                   37
$  PATTERN      VMAX      AMAX    LVLADA   SIZEADA   ADATIMS     D3PLT       GAP
         1    5000.0  500000.0         5       0.9      20.0      15.0       1.1
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*CONTROL_FORMING_POSITION
$ This is for tools in home position.
$      PID   PREMOVE    TARGET
         3                   5
```

The following partial keyword example is for tools already positioned in relationship to the blank and ready to close. All assigned properties for the blank remain the same. Here the upper tool PID3 is not going to be moved back, but instead it will move forward to close with the lower tool of target PID 5 in the direction specified by the vector ID 999.

```
*CONTROL_FORMING_USER
         1         0       1.0       1.5       1.6       1.4         F         1
     90903                                                                   37
         1    5000.0  500000.0         5       0.9      20.0      15.0       1.1
*CONTROL_FORMING_TRAVEL
$      PID       VID    TRAVEL    TARGET       GAP     PHASE    FOLLOW
         3       999                   5       1.1         1
```

Revision information:

This keyword is available starting in LS-DYNA Revision 48319.

*CONTROL_FREQUENCY_DOMAIN

Purpose: Set global control flags and parameters for frequency domain analysis.

Card 1	1	2	3	4	5	6	7	8
Variable	REFGEO	MPN						
Type	I	F						
Default	0	0.0						

VARIABLE	DESCRIPTION
REFGEO	Flag for reference geometry in acoustic eigenvalue analysis: EQ.0: use original geometry ($t = 0$), EQ.1: use deformed geometry at the end of transient analysis.
MPN	Large mass added per node, to be used in large mass method for enforced motion.

Remarks:

1. For acoustic eigenvalue analysis (see keyword *FREQUENCY_DOMAIN_ACOUSTIC_ FEM_EIGENVALUE), sometimes it is desired to extract the eigenvalues at the end of transient analysis, based on the deformed geometry. This is useful to study the effect of loading history on acoustic eigenvalues. In this case, one can set REFGEO = 1 to use the deformed geometry at the end of transient analysis.

2. For enforced motion excitation (e.g. nodal acceleration, velocity, or displacement) in FRF, SSD, or random vibration analysis, one can use the large mass method to compute the response. With the large mass method, the user attaches a large mass to the nodes under excitation. LS-DYNA converts the enforced motion excitation to nodal force on the same nodes in the same direction, to produce the desired enforced motion. MPN is the large mass attached to each node under excitation (usually it is in the range of 10^5-10^7 times of the original mass of the entire structure). User still need to apply the large mass to the nodes using the keyword *ELEMENT_MASS_*{OPTION}*.

   The large nodal force p is computed as follows, For nodal acceleration, $p = m_L \ddot{u}$

For nodal velocity, $p = i\,\omega m_L \dot{u}$

For nodal displacement, $p = -\,\omega^2 m_L u$

where ω is the round frequency, m_L is the large mass attached to each node (MPN), $\ddot{u}, \dot{u}$ and u are the enforced acceleration, velocity and displacement.

CONTROL_HOURGLASS_*{OPTION}

Available options include:

<BLANK>

936

The "936" option switches the hourglass formulation for shells so that it is identical to that used in LS-DYNA version 936. The modification in the hourglass control from version 936 was to ensure that all components of the hourglass force vector are orthogonal to rigid body rotations. However, problems that run under version 936 sometimes lead to different results in versions 940 and later. This difference in results is primarily due to the modifications in the hourglass force vector. Versions released after 936 should be more accurate.

Purpose: Redefine the default values of hourglass control type and coefficient.

Card 1	1	2	3	4	5	6	7	8
Variable	IHQ	QH						
Type	I	F						
Default		0.1						
Remarks	1,2	3,4						

VARIABLE	DESCRIPTION

IHQ Default hourglass control type:

EQ.0: see Remark 1,

EQ.1: standard viscous form (may inhibit body rotation if solid element shapes are skewed),

EQ.2: viscous form, Flanagan-Belytschko integration for solid elements,

EQ.3: viscous form, Flanagan-Belytschko with exact volume integration for solid elements,

EQ.4: stiffness form of type 2 (Flanagan-Belytschko),

EQ.5: stiffness form of type 3 (Flanagan-Belytschko) for solid

VARIABLE	DESCRIPTION
	elements,
	EQ.6: Belytschko-Bindeman [1993] assumed strain co-rotational stiffness form for 2D and 3D solid elements,
	EQ.7: Linear total strain form of type 6 hourglass control.
	EQ.8: Activates full projection warping stiffness for shell formulations 16 and -16, and is the default for these shell formulations. A speed penalty of 25% is common for this option.
	EQ.9: Puso [2000] enhanced assumed strain stiffness form for 3D hexahedral elements,
	EQ.10: Cosserat Point Element (CPE) developed by Jabareen and Rubin [2008] for 3D hexahedral elements and Jabareen et.al [2013] for 10-noded tetrahedral elements. See Remark 6.
QH	Default hourglass coefficient.

Remarks:

1. Hourglass control is viscosity or stiffness that is added to quadrilateral shell elements and hexahedral solid elements that use reduced integration. It also applies to type 1 tshells. Without hourglass control, these elements would have zero energy deformation modes which could grow large and destroy the solution. *CONTROL_HOURGLASS can be used to redefine the default values of the hourglass control type and coefficient. If omitted or if IHQ = 0, the default hourglass control types are as follows:

   a) For shells: viscous type for explicit; stiffness type for implicit.

   b) For solids: type 2 for explicit; type 6 for implicit.

   c) For tshell formulation 1: type 2.

   These default values are used unless HGID on *PART is used to point to *HOURGLASS data which overrides the default values for that part.

   For explicit analysis, shell elements can be used with viscous hourglass control, (IHQ = 1 = 2 = 3) or stiffness hourglass control (IHQ = 4 = 5). Only shell forms 16 and -16 use the warping stiffness invoked by IHQ = 8. For implicit analysis, the viscous form is unavailable.

For explicit analysis, hexahedral elements can be used with any of the hourglass control types except IHQ = 8. For implicit analysis, only IHQ = 6, 7, 9, and 10 are available.

If IHQ is set to a value that is invalid for some elements in a model, then the hourglass control type for those elements is automatically reset to a valid value. For explicit analysis, if IHQ = 6, 7, 9, or 10, then shell elements will be switched to type 4 except for form 16 and -16 shells that are switched to type 8. If IHQ = 8, then solid elements and shell elements that are not form 16 or -16 will be switched to type 4. For implicit analysis, if IHQ = 1-5, then solid elements will be switched to type 6, and if IHQ = 1, 2, 3, 6, 7, 9, or 10, then shell elements will switched to type 4.

2. Viscous hourglass control has been used successfully with shell elements when the response with stiffness based hourglass control was overly stiff. As models have grown more detailed and are better able to capture deformation modes, there is less need for viscous forms. To maintain back compatibility, viscous hourglass control remains the default for explicit analysis, but there may be better choices, particularly the newer forms for bricks (6, 7, 9, and 10).

3. QH is a coefficient that scales the hourglass viscosity or stiffness. With IHQ = 1 through 5 and IHQ = 8, values of QH that exceed 0.15 may cause instabilities. Hourglass types 6, 7, 9, and 10 will remain stable with larger QH and can work well with QH = 1.0 for many materials. However, for plasticity models, a smaller value such as QH = 0.1 may work better since the hourglass stiffness is based on elastic properties.

4. Hourglass types 6, 7, 9, and 10 for hexahedral elements are based on physical stabilization using an enhanced assumed strain method. When element meshes are not particularly skewed or distorted, their behavior may be very similar and all can produce accurate coarse mesh bending results for elastic material with QH = 1.0. However, form 9 gives more accurate results for distorted or skewed elements. In addition, for materials 3, 18 and 24 there is the option to use a negative value of QH. With this option, the hourglass stiffness is based on the current material properties, i.e., the plastic tangent modulus, and scaled by |QH|.

5. Hourglass type 7 is a variation on form 6. Instead of updating the hourglass forces incrementally using the current stiffness and an increment of deformations, the total hourglass deformation is evaluated each cycle. This ensures that elements always spring back to their initial geometry if the load is removed and the material has not undergone inelastic deformation. Hourglass type 7 is recommended for foams that employ *INITIAL_FOAM_REFERENCE_GEOMETRY. However the CPU time for type 7 is roughly double that for type 6, so it is only recommended when needed.

6. Hourglass type 10 for 1-point solid elements or 10-noded tetrahedron of type 16 are strucural elements based on Cosserat point theory that allows for accurate representation of elementary deformation modes (stretching, bending and torsion) for general element shapes and hyperelastic materials. To this end, the theory in Jabareen and Rubin [2008] and Jabareen et.al [2013] has been generalized in the implementation to account for any material response. The deformation is separated into a homogenous and an inhomogeneous part where the former is treated by the constitutive law and the latter by a hyperelastic formulation that is set up to match analytical results for the deformation modes mentioned above. Tests have shown that the element is giving more accurate results than other hexahedral elements for small deformation problems and more realistic behavior in general.

*CONTROL_IMPLICIT

Purpose: Set parameters for implicit calculation features.

*CONTROL_IMPLICIT_AUTO

*CONTROL_IMPLICIT_BUCKLE

*CONTROL_IMPLICIT_CONSISTENT_MASS

*CONTROL_IMPLICIT_DYNAMICS

*CONTROL_IMPLICIT_EIGENVALUE

*CONTROL_IMPLICIT_FORMING

*CONTROL_IMPLICIT_GENERAL

*CONTROL_IMPLICIT_INERTIA_RELIEF

*CONTROL_IMPLICIT_JOINTS

*CONTROL_IMPLICIT_MODAL_DYNAMIC

CONTROL_IMPLICIT_MODAL_DYNAMIC_DAMPING_*{OPTION}

CONTROL_IMPLICIT_MODAL_DYNAMIC_MODE_*{OPTION}

CONTROL_IMPLICIT_MODES_*{OPTION}

*CONTROL_IMPLICIT_ROTATIONAL_DYNAMICS

*CONTROL_IMPLICIT_SOLUTION

*CONTROL_IMPLICIT_SOLVER

*CONTROL_IMPLICIT_STABILIZATION

*CONTROL_IMPLICIT_STATIC_CONDENSATION

*CONTROL_IMPLICIT_TERMINATION

CONTROL_IMPLICIT_AUTO_*{OPTION}

Available options for *OPTION* include:

<BLANK>

DYN

SPR

Purpose: Define parameters for automatic time step control during implicit analysis (see also *CONTROL_IMPLICIT_GENERAL). The DYN option allows setting controls specifically for the dynamic relaxation phase. The SPR option allows setting controls specifically for the springback phase.

Card 1	1	2	3	4	5	6	7	8
Variable	IAUTO	ITEOPT	ITEWIN	DTMIN	DTMAX	DTEXP	KFAIL	KCYCLE
Type	I	I	I	F	F	F		
Default	0	11	5	DT/1000.	DT×10.	none		

VARIABLE	DESCRIPTION
IAUTO	Automatic time step control flag EQ.0: constant time step size EQ.1: automatically adjust time step size EQ.2: automatically adjust time step size and synchronize with thermal mechanical time step. LT.0: Curve ID = (-IAUTO) gives time step size as a function of time. If specified, DTMIN and DTMAX will still be applied.
ITEOPT	Optimum equilibrium iteration count per time step. See Figure 12-65.
ITEWIN	Allowable iteration window. If iteration count is within ITEWIN iterations of ITEOPT, step size will not be adjusted for the next step.

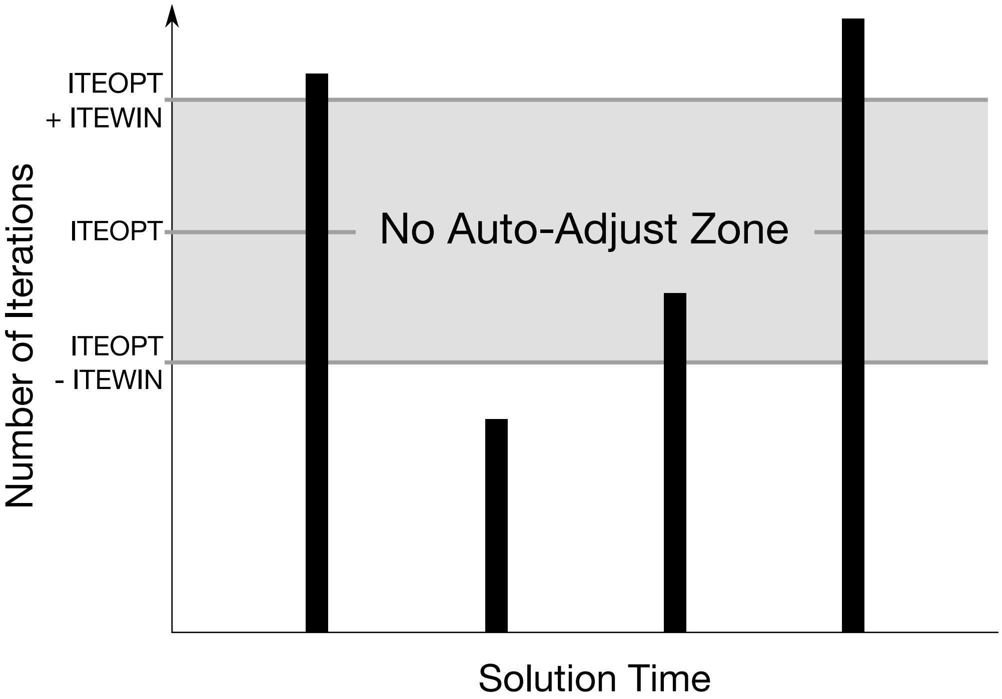


Figure 12-65. Iteration Window as defined by ITEOPT and ITEWIN.

VARIABLE	DESCRIPTION
DTMIN	Minimum allowable time step size. Simulation stops with error termination if time step falls below DTMIN.
DTMAX	Maximum allowable time step size. LT.0: curve ID = (-DTMAX) gives max step size as a function of time. Also, the step size is adjusted automatically so that the time value of each point in the curve is reached exactly (see Figures 12-66 and 12-67).
DTEXP	Time interval to run in explicit mode before returning to implicit mode. Applies only when automatic implicit-explicit switching is active (IMFLAG = 4 or 5 on *CONTROL_IMPLICIT_GENERAL). Also, see KCYCLE. EQ.0: defaults to the current implicit time step size. LT.0: curve ID = (-DTEXP) gives the time interval as a function of time.
KFAIL	Number of failed attempts to converge implicitly for the current time step before automatically switching to explicit time integration. Applies only when automatic implicit-explicit switching is active. The default is one attempt. If IAUTO = 0, any input value is reset to unity.

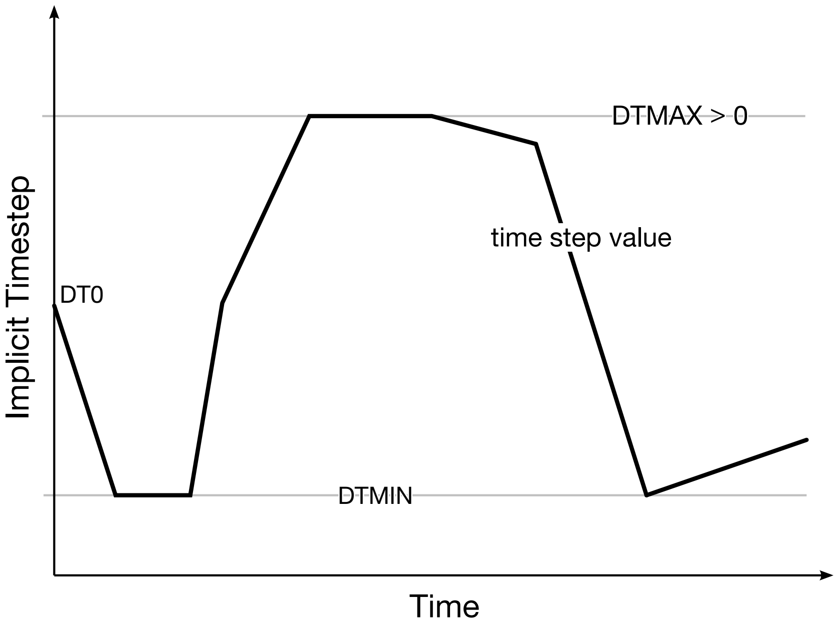


Figure 12-66. The implicit time step size changes continuously as a function of convergence within the bounds set by DTMIN and DTMAX

VARIABLE	DESCRIPTION
KCYCLE	Number of explicit cycles to run in explicit mode before returning to the implicit mode. The actual time interval that is used will be the maximum between DTEXP and KCYCLE*(latest estimate of the explicit time step size).

Remarks:

VARIABLE	REMARK
IAUTO	The default for IAUTO depends on the analysis type. For "springback" analysis, automatic time step control and artificial stabilization are activated by default.
ITEOPT	With IAUTO = 1 or 2, the time step size is adjusted if convergence is reached in a number of iterations that falls outside the specified "iteration window", increasing after "easy" steps, and decreasing after "difficult" but successful steps. ITEOPT defines the midpoint of the iteration window. A value of ITEOPT = 30 or more can be more efficient for highly nonlinear simulations by allowing more iterations in each step, hence fewer total steps.
ITEWIN	The step size is not adjusted if the iteration count falls within ITEWIN of ITEOPT. Large values of ITEWIN make the controller more tolerant of variations in iteration count.

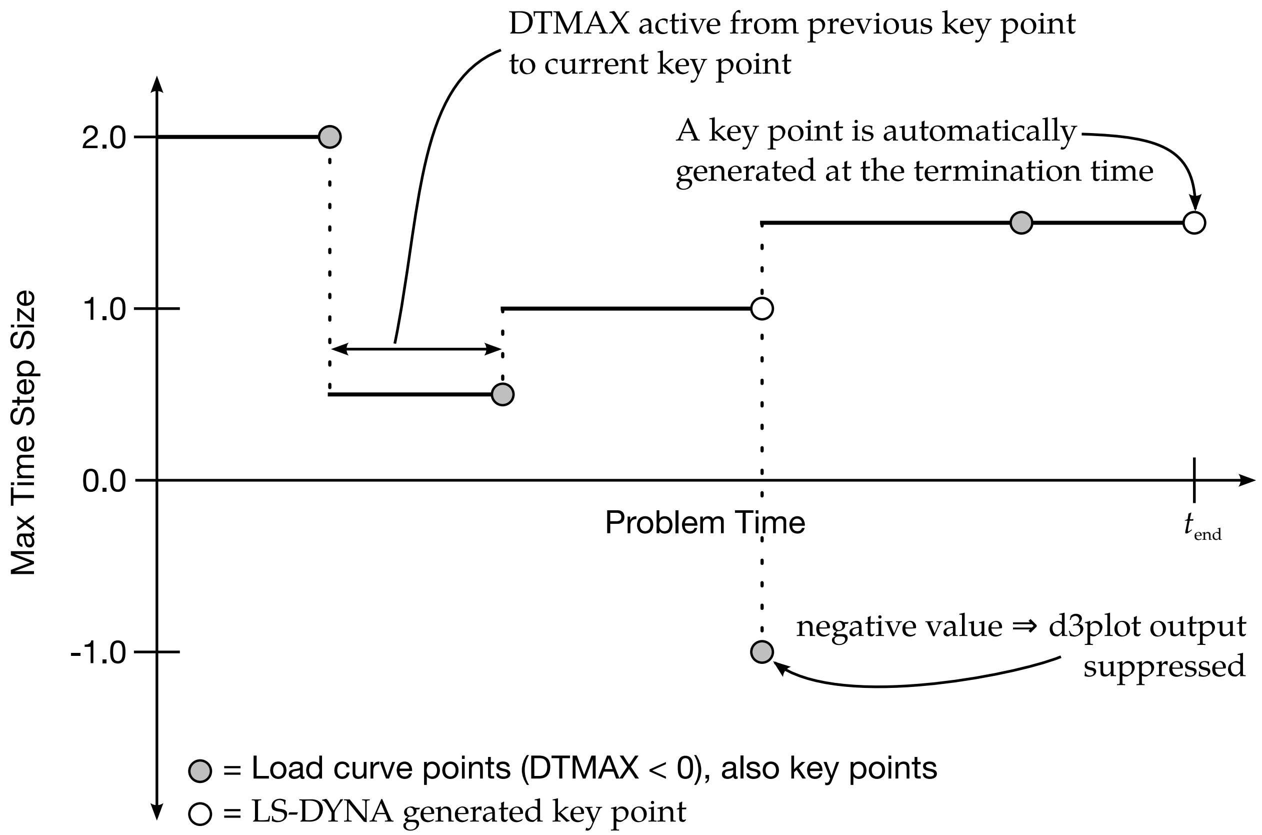


Figure 12-67. *DTMAX < 0.* The maximum time step is set by a load curve of LCID = −DTMAX interpolated using piecewise constants. The abscissa values of the load curve determine the set of *key points*. The *absolute value* of the ordinate values set the maximum time step size. *Key points* are special time values for which the integrator will adjust the time step so as to reach *exactly*. For each key point with a positive function value, LS-DYNA will write the state to the binary database.

VARIABLE	REMARK
DTMAX	To strike a particular simulation time exactly, create a key point curve (Figure 12-67) and enter DTMAX = -(curve ID). This is useful to guarantee that important simulation times, such as when peak load values occur, are reached exactly.
DTEXP	When the automatic implicit-explicit switching option is activated (IMFLAG = 4 or 5 on *CONTROL_IMPLICIT_GENERAL), the solution method will begin as implicit, and if convergence of the equilibrium iterations fails, automatically switch to explicit for a time interval of DTEXP. A small value of DTEXP should be chosen so that significant dynamic effects do not develop during the explicit phase, since these can make recovery of nonlinear equilibrium difficult during the next implicit time step. A reasonable starting value of DTEXP may equal several hundred

VARIABLE	REMARK
	explicit time steps.

*CONTROL_IMPLICIT_BUCKLE

Purpose: Activate implicit buckling analysis when termination time is reached (see also *CONTROL_IMPLICIT_GENERAL). Optionally, buckling analyses are performed at intermittent times.

Card 1	1	2	3	4	5	6	7	8
Variable	NMODE	BCKMTH						
Type	I	I						
Default	0	see below						

VARIABLE	DESCRIPTION
NMODE	Number of buckling modes to compute EQ.0: none (DEFAULT) GT.0: compute n lowest buckling modes LT.0: curve ID = (-NEIG) used for intermittent buckling analysis
BCKMTH	Method used to extract buckling modes EQ.1: Use Block Shift and Invert Lanczos. Default of all problems not using *CONTROL_IMPLICIT_INERTIA_RELIEF. EQ.2: Use Power Method. Only valid option for problems using *CONTROL_IMPLICIT_INERTIA_RELIEF. Optional for other problems. See Remarks.

Remarks:

Buckling analysis is performed at the end of a static implicit simulation or at specified times during the simulation. The simulation may be linear or nonlinear but must be implicit. After loads have been applied to the model, the buckling eigenproblem is solved:

$$[\mathbf{K}_M + \lambda \mathbf{K}_G]\{u\} = 0$$

where $\mathbf{K}_M$ is the material tangent stiffness matrix, and the geometric or initial stress stiffness matrix $\mathbf{K}_G$ is a function of internal stress in the model. The lowest n eigenvalues and eigenvectors are computed. The eigenvalues, written to text file "eigout", represent multipliers to the applied loads which give buckling loads. The eigenvectors, written to binary database "d3eigv", represent buckling mode shapes. View and animate these modes using LS-PrePost. When NMODE > 0, eigenvalues will be computed at the termination time and LS-DYNA will terminate.

When NMODE < 0, an intermittent buckling analysis will be performed. This is a transient simulation during which loads are applied, with buckling modes computed periodically during the simulation. Changes in geometry, stress, material, and contact conditions will affect the buckling modes. The transient simulation must be implicit. The curve ID = -NMODE indicates when to extract the buckling modes, and how many to extract. Define one curve point at each desired extraction time, with a function value equal to the number of buckling modes desired at that time. A d3plot database will be produced for the transient solution results. Consecutively numbered d3eigv and eigout databases will be produced for each intermittent extraction. The extraction time is indicated in each database's analysis title.

The buckling modes can be computed using either Block Shift and Invert Lanczos or the Power Method. It is strongly recommended that the Block Shift and Invert Lanczos method is used as it is a more powerful and robust algorithm. For problems using *CONTROL_IMPLICIT_INERTIA_RELIEF the Power Method must be used and any input value for BCKMTH will be overridden with the required value of 2. There may be some problems, which are not using *CONTROL_IMPLICIT_INERTIA_RELIEF, where the Power Method may be more efficient than Block Shift and Invert Lanczos. But the Power Method is not as robust and reliable as Lanczos and results should be verified. Furthermore convergence of the Power Method is better for buckling problems where the expected buckling mode is close to one in magnitude and the dominant mode is separated from the secondary modes. The number of modes extracted via the Power Method should be kept in the range of 1 to 5.

The geometric stiffness terms needed for buckling analysis will be automatically computed when the buckling analysis time is reached, regardless of the value of the geometric stiffness flag IGS on *CONTROL_IMPLICIT_GENERAL.

A double precision executable should be used for best accuracy in buckling analysis.

Parameters CENTER, LFLAG, LFTEND, RFLAG, RHTEND and SHFSCL from *CONTROL_IMPLICIT_EIGENVALUE are applicable to buckling analysis. For buckling analysis CENTER, LFTEND, RHTEND and SHFSCL are in units of the eigenvalue spectrum.

*CONTROL_IMPLICIT_CONSISTENT_MASS

Purpose: Use the consistent mass matrix in implicit dynamics and eigenvalue solutions.

Card 1	1	2	3	4	5	6	7	8
Variable	IFLAG							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
IFLAG	Consistent mass matrix flag EQ.0: Use the standard lumped mass formulation (DEFAULT) EQ.1: Use the consistent mass matrix.

Remarks:

The consistent mass matrix formulation is currently available for the three and four node shell elements, solid elements types 1, 2, 10, 15, 16, and 18 (See *SECTION_SOLID), and beam types 1, 2, 3, 4, and 5 (See *SECTION_BEAM). All other element types continue to use a lumped mass matrix.

CONTROL_IMPLICIT_DYNAMICS_*{OPTION}

Available options include:

<BLANK>

DYN

SPR

Purpose: Activate implicit dynamic analysis and define time integration constants (see also *CONTROL_IMPLICIT_GENERAL). The DYN option allows setting controls specifically for the dynamic relaxation phase. The SPR option allows setting control specifically for the springback phase.

Card 1	1	2	3	4	5	6	7	8
Variable	IMASS	GAMMA	BETA	TDYBIR	TDYDTH	TDYBUR	IRATE	ALPHA
Type	I	F	F	F	F	F	I	F
Default	0	.50	.25	0.0	10^{28}	10^{28}	0	0

VARIABLE	DESCRIPTION
IMASS	Implicit analysis type LT.0: curve ID = (-SCALE) used to control amount of implicit dynamic effects applied to the analysis. TDYBIR, TDYDTH and TDYBUR are ignored with this option. EQ.0: static analysis EQ.1: dynamic analysis using Newmark time integration. EQ.2: dynamic analysis by modal superposition following the solution of the eigenvalue problem EQ.3: dynamic analysis by modal superposition using the eigenvalue solution in the d3eigv files that are in the runtime directory.
GAMMA	Newmark time integration constant (see remarks below).
BETA	Newmark time integration constant (see remarks below).

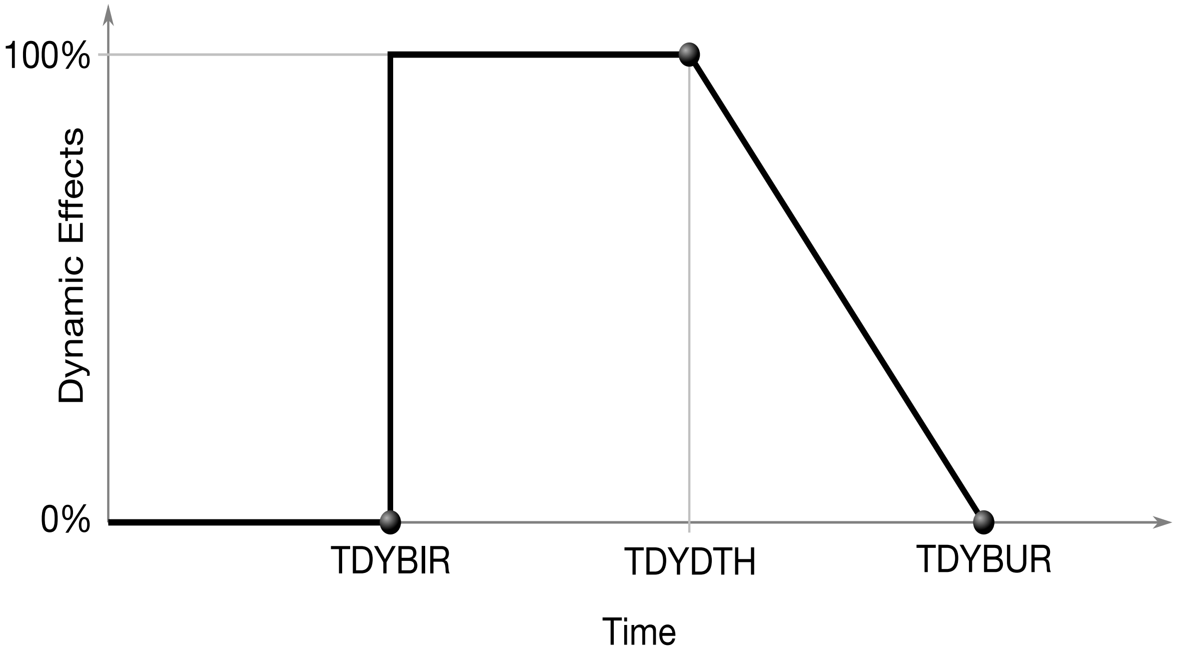


Figure 12-68. Birth, death, and burial time for implicit dynamics. The terms involving $\boldsymbol{M}$ and $\boldsymbol{D}$ are scaled by a factor between ranging between 1 and 0 to include or exclude dynamical effects, respectively.

VARIABLE	DESCRIPTION
TDYBIR	Birth time for application of dynamic terms. See Figure 12-68.
TDYDTH	Death time for application of dynamic terms.
TDYBUR	Burial time for application of dynamic terms.
IRATE	Rate effects switch: EQ.0: rate effects are on in constitutive models EQ.1: rate effects are off in constitutive models EQ.2: rate effects are off in constitutive models for both explicit and implicit.
ALPHA	Composite time integration constant (See remarks below). GT.0: Bathe composite scheme is activated LT.0: HHT scheme is activated

Remarks:

For the dynamic problem, the linearized equilibrium equations may be written in the form

$$\boldsymbol{M}\ddot{\boldsymbol{u}}^{n+1} + \boldsymbol{D}\dot{\boldsymbol{u}}^{n+1} + \boldsymbol{K}_t(\boldsymbol{x}^n)\Delta\boldsymbol{u} = \boldsymbol{P}(\boldsymbol{x}^n)^{n+1} - \boldsymbol{F}(\boldsymbol{x}^n)$$

where

$$\boldsymbol{M} = \text{lumped mass matrix}$$

$$\boldsymbol{D} = \text{damping matrix}$$

$$\boldsymbol{u}^{n+1} = \boldsymbol{x}^{n+1} - \boldsymbol{x}^0 = \text{nodal displacement vector}$$

$$\dot{\boldsymbol{u}}^{n+1} = \text{nodal point velocities at time } n+1$$

$$\ddot{\boldsymbol{u}}^{n+1} = \text{nodal point acceleration at time } n+1$$

Between the birth and death times 100% of the dynamic terms, that is the terms involving $\boldsymbol{M}$ and $\boldsymbol{D}$, are applied. Between the death and burial time the dynamic terms are decreased linearly with respect to time until 0% of the dynamic terms are applied after the burial time. This feature is useful for problems that are initially singular because the parts are not in contact initially such as in metal stamping. For these problems dynamics is required for stable convergence. When contact is established the problem becomes well conditioned and the dynamic terms are no longer required for stable convergence. It is recommend that for such problems the user set the death time to be after contact is established and the burial time for 2 or 3 time steps after the death time.

For problems with more extensive loading and unloading patterns the user can control the amount of dynamic effects added to the model by using a load curve, see IMASS.LT.0. This curve should have ordinate values between 0.0 and 1.0. The user should use caution in ramping the load curve and the associated dynamic effects from 1.0 to 0.0. Such a ramping down should take place over 2 or 3 implicit time steps.

The time integration is by default the unconditionally stable, one-step, Newmark-$\boldsymbol{\beta}$ time integration scheme

$$\ddot{\boldsymbol{u}}^{n+1} = \frac{\Delta\boldsymbol{u}}{\beta\Delta t^2} - \frac{\dot{\boldsymbol{u}}^n}{\beta\Delta t} - \frac{1}{\beta}\left(\frac{1}{2} - \beta\right)\ddot{\boldsymbol{u}}^n$$

$$\dot{\boldsymbol{u}}^{n+1} = \dot{\boldsymbol{u}}^n + \Delta t(1-\gamma)\ddot{\boldsymbol{u}}^n + \gamma\Delta t\ddot{\boldsymbol{u}}^{n+1}$$

$$\boldsymbol{x}^{n+1} = \boldsymbol{x}^n + \Delta\boldsymbol{u}$$

Here, Δt is the time step size, and β and γ are the free parameters of integration. For $\gamma = 1/2$ and $\beta = 1/4$ the method reduces to the trapezoidal rule and is energy conserving. If

$$\gamma > \frac{1}{2}$$

$$\beta > \frac{1}{4}\left(\frac{1}{2} + \gamma\right)^2,$$

Then numerical damping is induced into the solution leading to a loss of energy and momentum.

The Newmark method, and the trapezoidal rule in particular, is known to lack the robustness required for simulating long term dynamic implicit problems. Even though numerical damping may improve the situation from this aspect, it is difficult to know how to set γ and β without deviating from desired physical properties of the system. In the literature, a vast number of *composite* time integration algorithms have been proposed to handle this, and a family of such methods is implemented and governed by the value of α (ALPHA, parameter 8 on card 1). For $\alpha > 0$, every other implicit time step is a three point backward Euler step given as

$$\ddot{\boldsymbol{u}}^{n+1} = \frac{(1+\alpha)}{\Delta t}(\dot{\boldsymbol{u}}^{n+1} - \dot{\boldsymbol{u}}^{n}) - \frac{\alpha}{\Delta t_{-}}(\dot{\boldsymbol{u}}^{n} - \dot{\boldsymbol{u}}^{n-1})$$

$$\dot{\boldsymbol{u}}^{n+1} = \frac{(1+\alpha)}{\Delta t}\Delta\boldsymbol{u} - \frac{\alpha}{\Delta t_{-}}\Delta\boldsymbol{u}_{-}$$

where $\Delta t_{-} = t^{n} - t^{n-1}$ and $\Delta\boldsymbol{u}_{-} = \boldsymbol{u}^{n} - \boldsymbol{u}^{n-1}$ are constants. Because of this three step procedure, the method is particularly suitable for nodes/bodies undergoing curved motion as it better accounts for curvature than the default Newmark step. For $\alpha = 1/2$, and default values of γ and β, the method defaults to the Bathe time integration scheme, Bathe [2007], and is reported to preserve energy and momentum to a reasonable degree. The improvement in stability over the Newmark method is primarily attributed to numerical dissipation, but fortunately this dissipation appears to mainly be due to damping of high frequency content and the underlying physics is therefore not affected as such, see Bathe and Nooh [2012].

For a negative value of ALPHA, the HHT, Hilber-Hughes-Taylor [1977], scheme is activated. This scheme is similar to that of the Newmark method, but the equilibrium is sought at time step $n+1+\alpha$ instead of at $n+1$. As a complement to the Newmark scheme above, we introduce

$$\dot{\boldsymbol{u}}^{\alpha} = -\alpha\dot{\boldsymbol{u}}^{n} + (1+\alpha)\dot{\boldsymbol{u}}^{n+1}$$

$$\boldsymbol{x}^{\alpha} = -\alpha\boldsymbol{x}^{n} + (1+\alpha)\boldsymbol{x}^{n+1}$$

and solve a modified system of equilibrium equations

$$\boldsymbol{M}\ddot{\boldsymbol{u}}^{n+1} + \boldsymbol{D}\dot{\boldsymbol{u}}^{\alpha} + \boldsymbol{F}(\boldsymbol{x}^{\alpha}) = \boldsymbol{P}(\boldsymbol{x}^{\alpha}).$$

This method is stable for $-\frac{1}{3} \le \alpha \le 0$ and $\gamma = \frac{1-2\alpha}{2}$ and $\beta = \frac{(1-\alpha)^2}{4}$, which becomes the default values of γ and β if not explicitly set. Parameter α controls the amount of dissipation in the problem, for $\alpha = 0$ an undamped Newmark scheme is obtained, whereas $\alpha = -\frac{1}{3}$ introduces significant damping. From the literature, a value of $\alpha = -0.05$ appears to be a good choice.

When modal superposition is invoked, NEIGV on *CONTROL_IMPLICIT_EIGENVALUE indicates the number of modes to be used. With modal superposition, stresses are computed only for linear shell formulation 18.

*CONTROL_IMPLICIT_EIGENVALUE

Purpose: Activate implicit eigenvalue analysis and define associated input parameters (see also *CONTROL_IMPLICIT_GENERAL).

Card 1	1	2	3	4	5	6	7	8
Variable	NEIG	CENTER	LFLAG	LFTEND	RFLAG	RHTEND	EIGMTH	SHFSCL
Type	I	F	I	F	I	F	I	F
Default	0	0.0	0	-infinity	0	+infinity	2	0.0

This card is optional.

Card 2	1	2	3	4	5	6	7	8
Variable	ISOLID	IBEAM	ISHELL	ITSHELL	MSTRES	EVDUMP	MSTRSCL	
Type	I	I	I	I	I	I	F	
Default	0	0	0	0	0	0	0.001	

VARIABLE	DESCRIPTION
NEIG	Number of eigenvalues to extract. This must be specified. The other parameters below are optional. LT.0: curve ID = (-NEIG) used for intermittent eigenvalue analysis
CENTER	Center frequency. This option finds the nearest NEIG eigenvalues located about this value.
LFLAG	Left end point finite flag. EQ.0: left end point is -infinity EQ.1: left end point is LFTEND.
LFTEND	Left end point of interval. Only used when LFLAG = 1.

VARIABLE	DESCRIPTION
RFLAG	Right end point finite flag: EQ.0: right end point is +infinity EQ.1: right end point is RHTEND.
RHTEND	Right end point of interval. Only used when RFLAG = 1.
EIGMTH	Eigenvalue extraction method: EQ.2: Block Shift and Invert Lanczos (default). EQ.3: Lanczos with [M] = [I] (for debug only). EQ.5: Same as 3 but include Dynamic Terms
SHFSCL	Shift scale. Generally not used, but see explanation below.
ISOLID	If nonzero, reset all solid element formulations to ISOLID for the implicit computations. Can be used for all implicit computations not just eigenvalue computations.
IBEAM	If nonzero, reset all beam element formulations to IBEAM for the implicit computations. Can be used for all implicit computations not just eigenvalue computations.
ISHELL	If nonzero, reset all shell element formulations to ISHELL for the implicit computations. Can be used for all implicit computations not just eigenvalue computations.
ITSHELL	If nonzero, reset all thick shell element formulations to ITSHELL for the implicit computations. Can be used for all implicit computations not just eigenvalue computations.
MSTRES	Flag for computing the stresses for the eigenmodes: EQ.0: Do not compute the stresses. EQ.1: Compute the stresses.

VARIABLE	DESCRIPTION
EVDUMP	Flag for writing eigenvalues and eigenvectors to file "Eigen_Vectors" (SMP only): EQ.0: Do not write eigenvalues and eigenvectors. GT.0: Write eigenvalues and eigenvectors using an ASCII format. LT.0: Write eigenvalues and eigenvectors using a binary format.
MSTRSCL	Scaling for computing the velocity based on the mode shape for the stress computation.

Remarks:

To perform an eigenvalue analysis, activate the implicit method by selecting IMFLAG = 1 on *CONTROL_IMPLICIT_GENERAL, and indicate a nonzero value for NEIG above. By default, the lowest NEIG eigenvalues will be found. If a nonzero center frequency is specified, the NEIG eigenvalues nearest to CENTER will be found.

When NEIG > 0, eigenvalues will be computed at time = 0 and LS-DYNA will terminate.

When NEIG < 0, an intermittent eigenvalue analysis will be performed. This is a transient simulation during which loads are applied, with eigenvalues computed periodically during the simulation. Changes in geometry, stress, material, and contact conditions will affect the eigenvalues. The transient simulation can be either implicit or explicit according to IMFLAG = 1 or IMFLAG = 6, respectively, on *CONTROL_IMPLICIT_GENERAL. The curve ID = -NEIG indicates when to extract eigenvalues, and how many to extract. Define one curve point at each desired extraction time, with a function value equal to the number of eigenvalues desired at that time. A d3plot database will be produced for the transient solution results. Consecutively numbered d3eigv and eigout databases will be produced for each intermittent extraction. The extraction time is indicated in each database's analysis title.

The Block Shift and Invert Lanczos code is from BCSLIB-EXT, Boeing's Extreme Mathematical Library.

When using Block Shift and Invert Lanczos, the user can specify a semifinite or finite interval region in which to compute eigenvalues. Setting LFLAG = 1 changes the left end point from -infinity to the value specified by LFTEND. Setting RFLAG = 1 changes the right end point from +infinity to the values given by RHTEND. If the interval includes CENTER (default value of 0.0) then the problem is to compute the NEIG eigenvalues nearest to CENTER. If the interval does not include CENTER, the problem is to compute the smallest in magnitude NEIG eigenvalues.

If all of the eigenvalues are desired in an interval where both end points are finite just input a large number for NEIG. The software will automatically compute the number of eigenvalues in the interval and lower NEIG to that value. The most general problem specification is to compute NEIG eigenvalues nearest CENTER in the interval [LFTEND,RHTEND]. Computing the lowest NEIG eigenvalues is equivalent to computing the NEIG eigenvalues nearest 0.0.

For some problems it is useful to override the internal heuristic for picking a starting point for Lanczos shift strategy, that is the initial shift. In these rare cases, the user may specify the initial shift via the parameter SHFSCL. SHFSCL should be in the range of first few nonzero frequencies.

Parameters CENTER, LFTEND, RHTEND, and SHFSCL are in units of Hertz for eigenvalue problems. These four parameters along with LFLAG and RFLAG are applicable for buckling problems.. For buckling problems CENTER, LFTEND, RHTEND, and SHFSCL are in units of the eigenvalue spectrum.

Eigenvectors are written to an auxiliary binary plot database named "d3eigv", which is automatically created. These can be viewed using a postprocessor in the same way as a standard "d3plot" database. The time value associated with each eigenvector plot is the corresponding frequency in units of cycles per unit time. A summary table of eigenvalue results is printed to the "eigout" file. In addition to the eigenvalue results, modal participation factors and modal effective mass tables are written to the "eigout" file. The user can export individual eigenvectors using LSPrePost.

The user can request stresses to be computed and written to d3eigv via MSTRES. A velocity is computed by dividing the displacements from the eigenmode by MSTRSCL. The element routine then computes the stresses based on this velocity, but then those stresses are inversely scaled by MSTRSCL before being written to d3eigv. Thus MSTRSCL has no effect on results of linear element formulations. The strains associated with the stresses output using the MSTRES option can be obtained by setting the STRFLG on *DATABASE_EXTENT_BINARY.

Eigenvalues and eigenvectors can be written to file "Eigen_Vectors" by using a nonzero value for EVDUMP. If EVDUMP > 0 an ASCII file is used. If EVDUMP < 0 a simple binary format is used. The binary format is to reduce file space. The eigenvectors written to this file will be orthonormal with respect to the mass matrix. Eigenvector dumping is an SMP only feature.

The print control parameter, LPRINT, and ordering method parameter, ORDER, from the *CONTROL_IMPLICIT_SOLVER keyword card also apply to the Block Shift and Invert Eigensolver.

CONTROL_IMPLICIT_FORMING_*{OPTION}

Available options include:

<BLANK>

DYN

SPR

Purpose: This keyword is used to perform implicit static analysis, especially for metal forming processes, such as gravity loading, binder closing, flanging, and stamping subassembly simulation. A systematic study had been conducted to identify the key factors affecting implicit convergence, and the preferred values are automatically set with this keyword. In addition to forming application, this keyword can also be used in other applications, such as dummy loading and roof crush, etc. The DYN option allows setting controls specifically for the dynamic relaxation phase. The SPR option allows setting controls specifically for the springback phase.

Card 1	1	2	3	4	5	6	7	8
Variable	IOPTION	NSMIN	NSMAX	BIRTH	DEATH	PENCHK		
Type	I	I	I	F	F	F		
Default	1	none	2	0.0	1.e+20	0.0		

VARIABLE **DESCRIPTION**

IOPTION Solution type:

EQ.1: Gravity loading simulation, see remarks below.

EQ.2: Binder closing and flanging simulation, see remarks below.

NSMIN Minimum number of implicit steps for IOPTION = 2.

NSMAX Maximum number of implicit steps for IOPTION = 2.

BIRTH Birth time to activate this feature.

DEATH Death time.

VARIABLE	DESCRIPTION
PENCHK	Relative allowed penetration with respect to the part thickness in contact for IOPTION = 2.
DT0	Initial time step size, as defined in *CONTROL_IMPLICIT_GENERAL, which is no longer needed if DT0 is specified here.

General remarks:

This keyword provides a simplified interface for implicit static analysis. If no other implicit cards are used, the stiffness matrix is reformed every iteration. Convergence tolerances (DCTOL, ECTOL, etc.) are automatically set and recommended no to be changed. In almost all cases, only two additional implicit control cards (*CONTROL_-IMPLICIT_GENERAL, and_AUTO) may be needed to control the stepping size, where variables *DT0*, *DTMIN* and *DTMAX* can be used for control.

If multiple steps are required for IOPTION = 1, *CONTROL_IMPLICIT_GENERAL must be placed **after** *CONTROL_IMPLICIT_FORMING with *DT0* specified as a certain fraction of the ENDTIM (see *CONTROL_TERMINATION). Otherwise, even with *DT0* specified as a fraction of the ENDTIM, only one step (with step size of ENDTIM) will be performed.

As always, the variable *IGAP* should be set to "2" in *CONTACT_FORMING… cards for a more realistic contact simulation in forming. The contact type *CONTACT_-FORMING_SURFACE_TO_SURFACE is recommended to be used with implicit analysis.

Smaller penalty stiffness scale factor *SLSFAC* produces a certain amount of contact penetration but yields faster simulation time, and therefore is recommended for gravity and closing (in case of no physical beads) simulation. Subsequent forming process is likely to follow and contact conditions will be reestablished there, where a tighter, default *SLSFAC* (0.1) should be used.

It is recommended that the fully integrated element type 16 is to be used for all implicit calculation. For solids, type "-2" is recommended.

Executable with double precision is to be used for all implicit calculation.

Models with over 100,000 deformable elements are more efficient to be simulated with MPP for faster turnaround time.

Gravity loading example:

An example of the implicit gravity is provided below, where a blank is loaded with gravity into a toggle die. A total of five steps are used, controlled by the variable *DT0*. The results are shown in Figure 12-69. If this binder closing is done with explicit dynamics, efforts need to be made to reduce the inertia effects on the blank since contact with the upper binder only happens along the periphery and a large middle portion of the blank is not driven or supported by anything. With implicit static method, there is no inertia effect at all on the blank during the closing, and no tool speed, time step size, etc. to be concerned about.

The implicit gravity application for both air and toggle draw process is available through *LS-PrePost 4.0* in *Metal Forming Application/eZ Setup* (http://ftp.lstc.com/anonymous/outgoing/lsprepost/4.0/metalforming/).

```
*KEYWORD
*PARAMETER
⋮
*CONTROL_TERMINATION
1.0
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*CONTROL_IMPLICIT_FORMING
$  IOPTION
         1
*CONTROL_IMPLICIT_GENERAL
$   IMFLAG       DT0
         1       0.2
*CONTROL_CONTACT
$   SLSFAC    RWPNAL    ISLCHK    SHLTHK    PENOPT    THKCHG     ORIEN
      0.03       0.0         2         1         4         0         4
$   USRSTR    USRFAC     NSBCS    INTERM     XPENE     SSTHK      ECDT   TIEDPRJ
         0         0        10         0       1.0         0
*PART
 Blank
   &blkpid   &blksec   &blkmid
*SECTION_SHELL
$      SID    ELFORM      SHRF       NIP     PROPT   QR/IRID     ICOMP     SETYP
&blksec          16     0.833         7       1.0
$       T1        T2        T3        T4      NLOC
&bthick,&bthick,&bthick,&bthick
*CONTACT_FORMING_SURFACE_TO_SURFACE
$     SSID      MSID     SSTYP     MSTYP    SBOXID    MBOXID       SPR       MPR
   &blksid  &lpunsid         2         2                             1         1
$       FS        FD        DC        VC       VDC    PENCHK        BT        DT
      0.12       0.0       0.0       0.0      20.0         0       0.0     1E+20
$      SFS       SFM       SST       MST      SFST      SFMT       FSF       VSF
       1.0       1.0       0.0     &mstp
$     SOFT    SOFSCL    LCIDAB    MAXPAR    PENTOL     DEPTH     BSORT    FRCFRQ
         0
$   PENMAX    THKOPT    SHLTHK     SNLOG      ISYM     I2D3D    SLDTHK    SLDSTF
                                       1
$     IGAP    IGNORE    DPRFAC    DTSTIF                        FLANGL
         2
⋮
*LOAD_BODY_Z
90994
*DEFINE_CURVE_TITLE
Body Force on blank
90994
```

```
0.0,9810.0
10.0,9810.0
*LOAD_BODY_PARTS
&blksid
*END
```

Binder closing example:

An example of binder closing and its progression is shown in Figures 12-70, 12-71, 12-72, and 12-73, using the NUMISHEET'05 deck lid inner, where a blank is being closed in a toggle die (modified). An adaptive level of three was used in the closing process. Gravity is and should be always applied at the same time, regardless if a prior gravity loading simulation is performed or not, as listed at the end of the input deck. The presence of the gravity helps the blank establish an initial contact with the tool, thus improving the convergence rate. The upper binder is moved down by a closing distance (defined by a parameter &bindmv) using a displacement boundary condition (VAD = 2), with a simple linearly increased triangle-shaped load curve. The variable *DT0* is set at 0.01, determined by the expected total deformation. The solver will automatically adjust based on the initial contact condition. The maximum step size is controlled by the variable *DTMAX*, and this value needs to be sufficiently small (<0.02) to avoid missing contact, but yet not too small causing a long running time. In some cases, this variable can be set larger, but the current value works for most cases.

```
*KEYWORD
*PARAMETER
⋮
*CONTROL_TERMINATION
1.0
*CONTROL_IMPLICIT_FORMING
$  IOPTION     NSMIN     NSMAX
         2         2       100
*CONTROL_IMPLICIT_GENERAL
$   IMFLAG       DT0
         1      0.01
*CONTROL_IMPLICIT_AUTO
$    IAUTO    ITEOPT    ITEWIN     DTMIN     DTMAX
         0         0         0      0.01      0.03
*CONTROL_ADAPTIVE
⋮
*CONTROL_CONTACT
$   SLSFAC    RWPNAL    ISLCHK    SHLTHK    PENOPT    THKCHG     ORIEN
      0.03       0.0         2         1         4         0         4
$   USRSTR    USRFAC     NSBCS    INTERM     XPENE     SSTHK      ECDT   TIEDPRJ
         0         0        10         0       1.0         0
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
⋮
*PART
 Blank
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
   &blkpid   &blksec   &blkmid                                  &adpyes
*SECTION_SHELL
$      SID    ELFORM      SHRF       NIP     PROPT   QR/IRID     ICOMP     SETYP
&blksec            16     0.833         7       1.0
$       T1        T2        T3        T4      NLOC
&bthick,&bthick,&bthick,&bthick
⋮
*CONTACT_FORMING_SURFACE_TO_SURFACE
```

```
$     SSID      MSID     SSTYP     MSTYP    SBOXID    MBOXID       SPR       MPR
   &blksid  &lpunsid         2         2                             1         1
$       FS        FD        DC        VC       VDC    PENCHK        BT        DT
      0.12       0.0       0.0       0.0      20.0         0       0.0     1E+20
$      SFS       SFM       SST       MST      SFST      SFMT       FSF       VSF
       1.0       1.0       0.0     &mstp
$     SOFT    SOFSCL    LCIDAB    MAXPAR    PENTOL     DEPTH     BSORT    FRCFRQ
         0
$   PENMAX    THKOPT    SHLTHK     SNLOG      ISYM     I2D3D    SLDTHK    SLDSTF
                                       1
$     IGAP    IGNORE    DPRFAC    DTSTIF                        FLANGL
         2
*CONTACT_...
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*BOUNDARY_PRESCRIBED_MOTION_RIGID
$   typeID       DOF       VAD      LCID        SF       VID     DEATH     BIRTH
&bindpid           3         2         3      -1.0         0
*DEFINE_CURVE
3
0.0,0.0
1.0,&bindmv
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$ Activate gravity on blank:
*LOAD_BODY_Z
90994
*DEFINE_CURVE_TITLE
Body Force on blank
90994
0.0,9810.0
10.0,9810.0
*LOAD_BODY_PARTS
&blksid
*END
```

Binder closing with real beads example:

Binder closing with real beads can also be done with implicit static, and with adaptive mesh. An example is shown in Figure 12-74, where a hood outer is being closed implicitly. It is noted a small buckle can be seen near the draw bead region along the fender line. These kind of small forming effects can be more accurately detected with implicit static method.

The implicit static closing can now be set up in *LS-PrePost v4.0 Metal Forming Application/eZ Setup* (http://ftp.lstc.com/anonymous/outgoing/lsprepost/4.0/metal-forming/).

Flanging example:

An example of flanging simulation using this feature is shown in Figures 12-75, 12-76 and 12-77, with NUMISHEET'02 fender outer, where flanging is conducted along the hood line. A partial input is provided below, where *DTMAX* is controlled by a load curve for contact and speed. The use of *DTMAX* with a load curve is an exception to the rule, where most of the time this is not needed. Smaller step sizes are better in some cases than larger step sizes, which may take longer to converge resulting from cutbacks in step sizes. Gravity, pad closing and flanging were set to 10%, 10% and 80% of the total step size, respectively. Pad travels a distance of '&padtrav' starting at 0.1, when it is to be automatically moved to close the gap with the blank due to gravity loading (*CONTACT_AUTO_MOVE), and finishing at 0.2 and held in that position until the end. Flanging steel travels a distance of '&flgtrav' starting at 0.2 and completing at 1.0. A detailed section view of the simulation follows in Figure 12-78.

```
*KEYWORD
*PARAMETER ...
*CONTROL_TERMINATION
1.0
*CONTROL_IMPLICIT_FORMING
$  IOPTION     NSMIN     NSMAX
         2         2       200
*CONTROL_IMPLICIT_GENERAL
         1     0.100
*CONTROL_IMPLICIT_AUTO
$    IAUTO    ITEOPT    ITEWIN     DTMIN     DTMAX
         0         0         0     0.005     -9980
*DEFINE_CURVE
9980
0.0,0.1
0.1,0.1
0.2,0.1
0.7,0.005
1.0,0.005
*CONTROL_ADAPTIVE...
*CONTROL_CONTACT...
*PART...
*SECTION_SHELL...
*CONTACT_...
*CONTACT_FORMING_SURFACE_TO_SURFACE_ID_MPP
2
0,200,,3,2,1.005
$     SSID      MSID     SSTYP     MSTYP    SBOXID    MBOXID       SPR       MPR
   &blksid   &padsid         2         2
$       FS        FD        DC        VC       VDC    PENCHK        BT        DT
      0.12       0.0       0.0       0.0      20.0         0       0.0     1E+20
$      SFS       SFM       SST       MST      SFST      SFMT       FSF       VSF
       1.0       1.0       0.0     &mstp
$     SOFT    SOFSCL    LCIDAB    MAXPAR    PENTOL     DEPTH     BSORT    FRCFRQ
         0
$   PENMAX    THKOPT    SHLTHK     SNLOG      ISYM     I2D3D    SLDTHK    SLDSTF
                                       1
$     IGAP    IGNORE    DPRFAC    DTSTIF                        FLANGL
         2
*BOUNDARY_PRESCRIBED_MOTION_RIGID
$   typeID       DOF       VAD      LCID        SF       VID     DEATH     BIRTH
   &padpid         3         2         3      -1.0         0
   &flgpid         3         2         4      -1.0         0
*DEFINE_CURVE
3
0.0,0.0
0.1,0.0
0.2,&padtrav
1.0,&padtrav
*DEFINE_CURVE
4
0.0,0.0
0.2,0.0
1.0,&flgtrav
$ Activate gravity on blank:
*LOAD_BODY_PARTS
```

```
&blksid
*LOAD_BODY_Z
90994
*DEFINE_CURVE_TITLE
Body Force on blank
90994
0.0,9810.0
10.0,9810.0
*CONTACT_AUTO_MOVE
$       ID    ContID        VID      LCID      ATIME
        -1         2         89         3        0.1
*END
```

Flanging simulation using IOPTION of 1:

IOPTOIN 1 can also be used for closing and flanging simulation, or other applications that go through large plastic strains or deformation. This is used when an equal step size throughout the simulation is desired, and is done by specifying the equal step size in the variable *DT0* in *CONTROL_IMPLICIT_GENERAL, as shown in the following keywords (other cards similar and not included), where *DT0* of 0.014 is chosen. Such an application is shown in Figures 12-79 and 12-80.

```
*CONTROL_TERMINATION
1.0
*CONTROL_IMPLICIT_FORMING
$  IOPTION
         1
*CONTROL_IMPLICIT_GENERAL
$   IMFLAG       DT0
         1     0.014
```

Switching between implicit dynamic and implicit static for gravity loading:

For sheet blank gravity loading, it is now possible to start the simulation using implicit dynamic method, switching to implicit static method at a user defined time until completion. This feature is activated by setting the variable TDYDTH in *CONTROL_-IMPLICIT_DYNAMICS and was recently (Rev. 81400) linked together with *CONTROL_IMPLICIT_FORMING. In a partial keyword example below, death time for the implicit dynamic is set at 0.55 second. The test model shown in Figure 12-81 (left) results in a gravity loaded blank shape in Figure 12-81 (right). Without the switching, the blank will look like as shown in Figure 12-82. The gravity loaded blank shape is more reasonable with the switching. A check on the energy history reveals that the kinetic energy dissipated completely at 0.60 second, Figure 12-83.

```
*CONTROL_TERMINATION
1.0
*CONTROL_IMPLICIT_FORMING
$  IOPTION     NSMIN     NSMAX     BIRTH     DEATH    PENCHK
         1
*CONTROL_IMPLICIT_DYNAMICS
$    IMASS     GAMMA      BETA    TDYBIR    TDYDTH    TDYBUR     IRATE
         1     0.600     0.380                0.55
```

Revision information:

This implicit capability is available in R5.0 and later releases. This keyword is implemented in LS-PrePost4.0 eZSetup for metal forming application.

1)Revision 64802: Multi-step gravity loading simulation.

2)Revision 81400: Switching feature between implicit dynamic and implicit static.

3)Revision 104837: variable DT0.

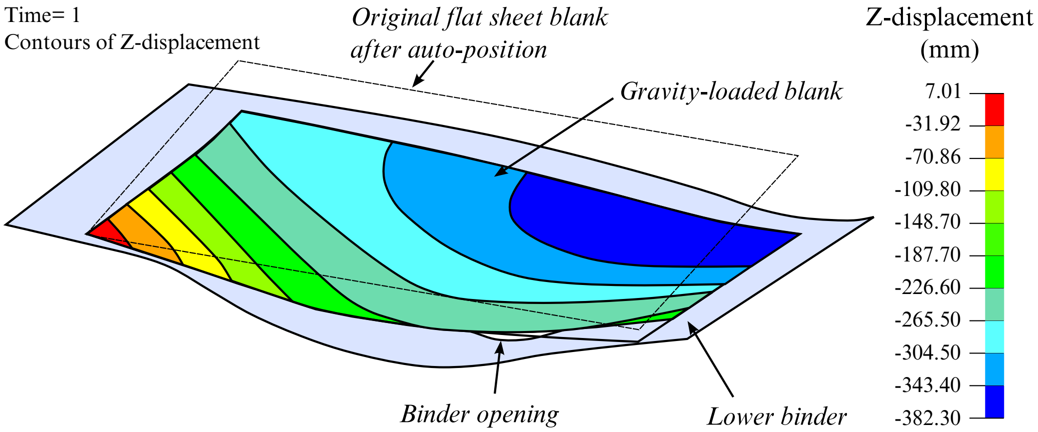


Figure 12-69. Gravity loading on a box side outer toggle die (courtesy of Autodie, LLC).

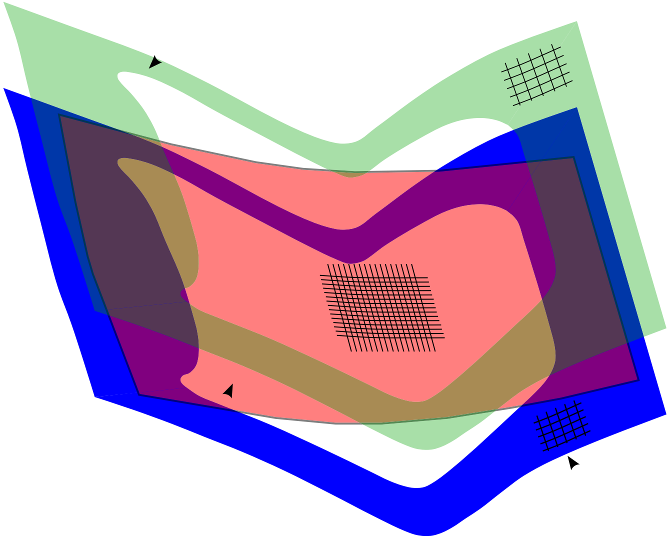


Figure 12-70. Initial auto-positioning (NUMISHEET2005 decklid inner).

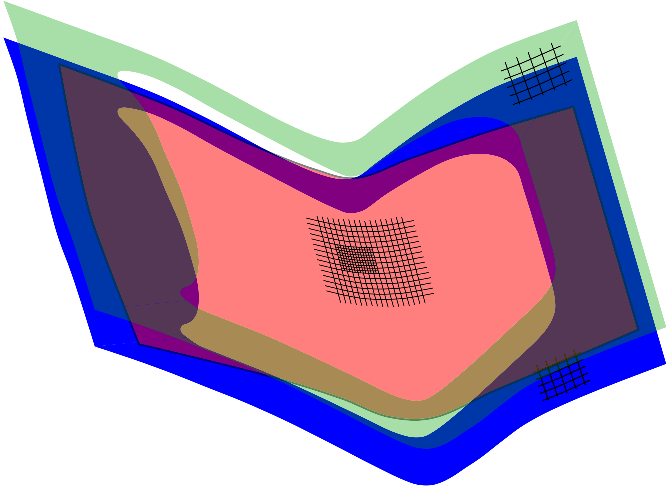


Figure 12-71. At 50% upper travel.

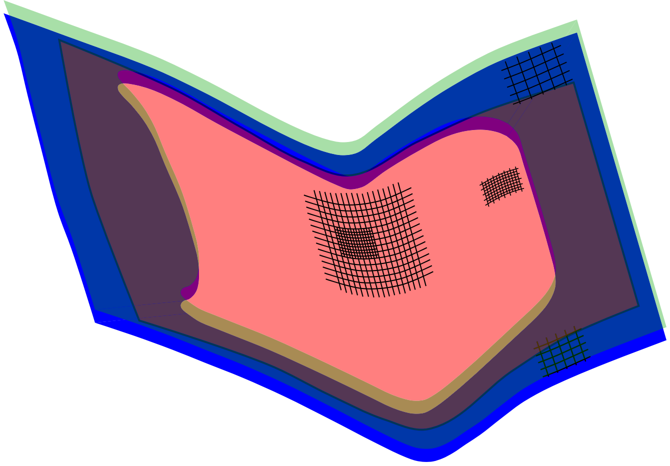


Figure 12-72. At 80% upper travel.

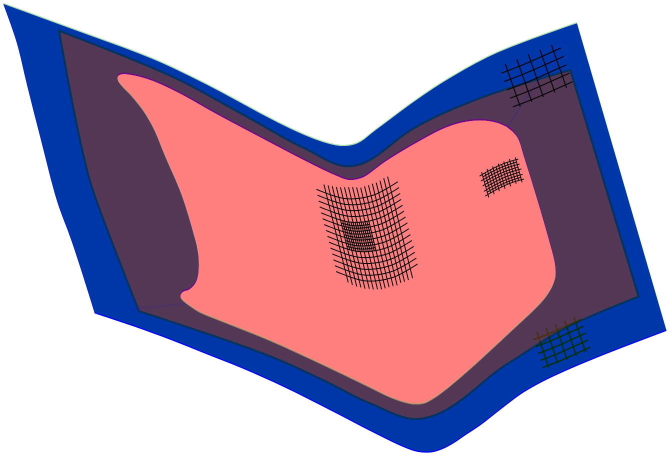


Figure 12-73. Upper travels to home.

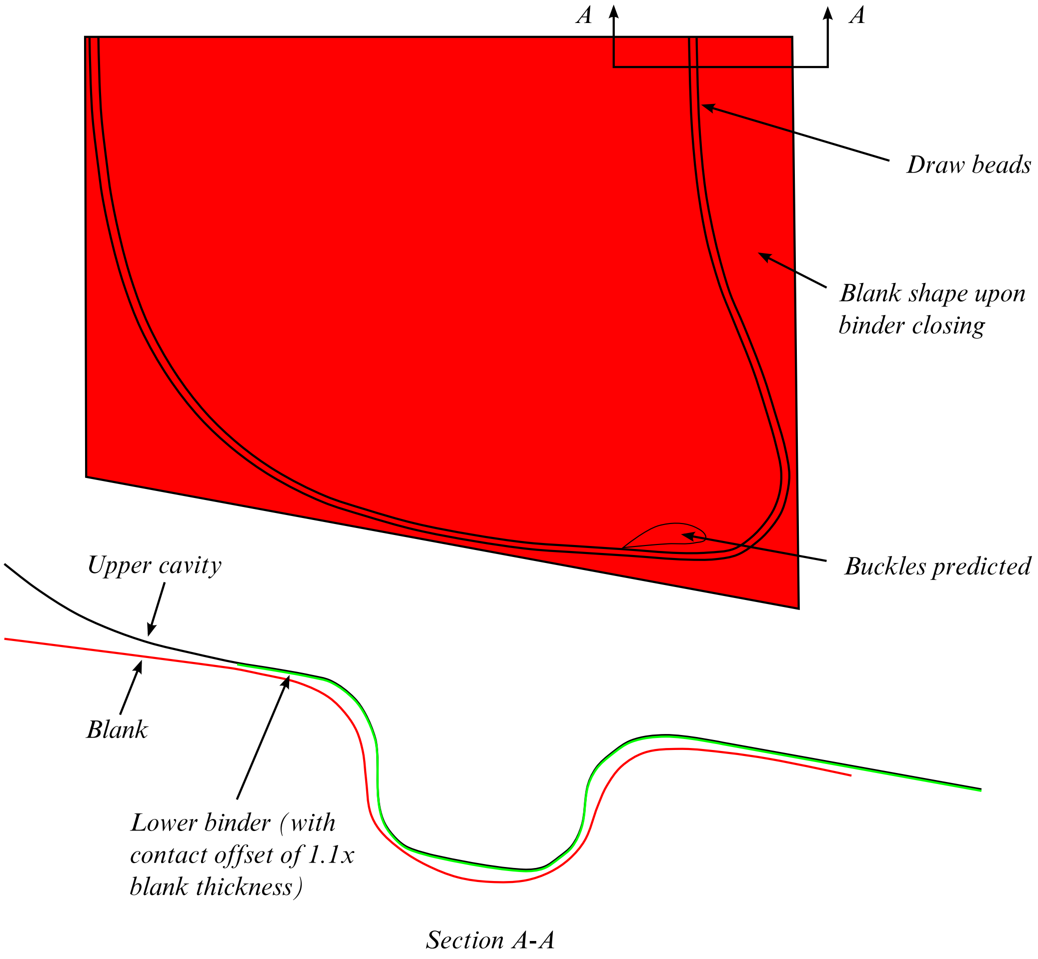

A
A
Draw beads
Blank shape upon binder closing
Buckles predicted
Upper cavity
Blank
Lower binder (with contact offset of 1.1x blank thickness)
Section A-A

Figure 12-74. Binder closing with beads on a hood outer.

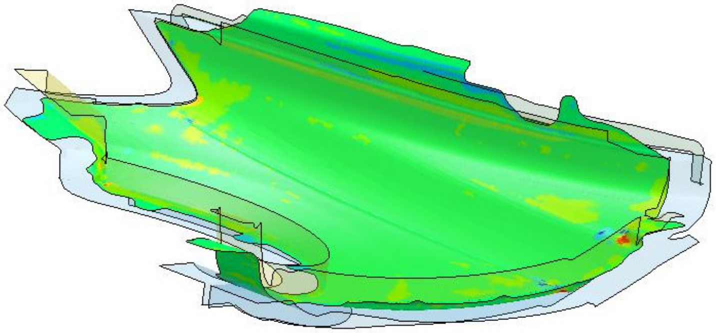


Figure 12-75. Mean stress at pad closing.

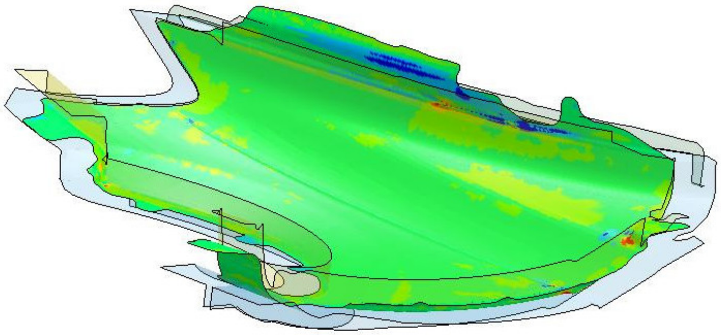


Figure 12-76. Mean stress at 40% Travel.

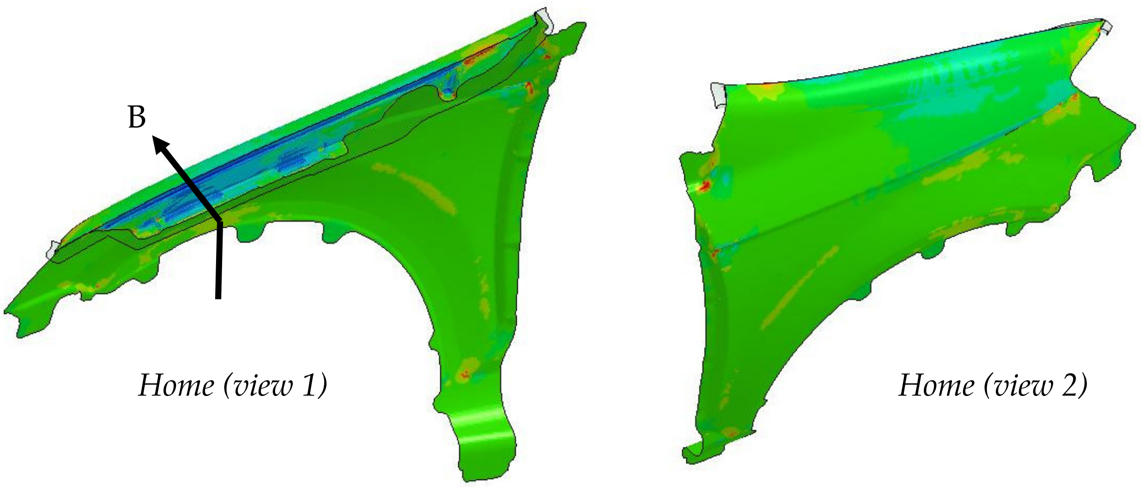


Figure 12-77. Mean stress at flanging home (compression/surface lows in red).

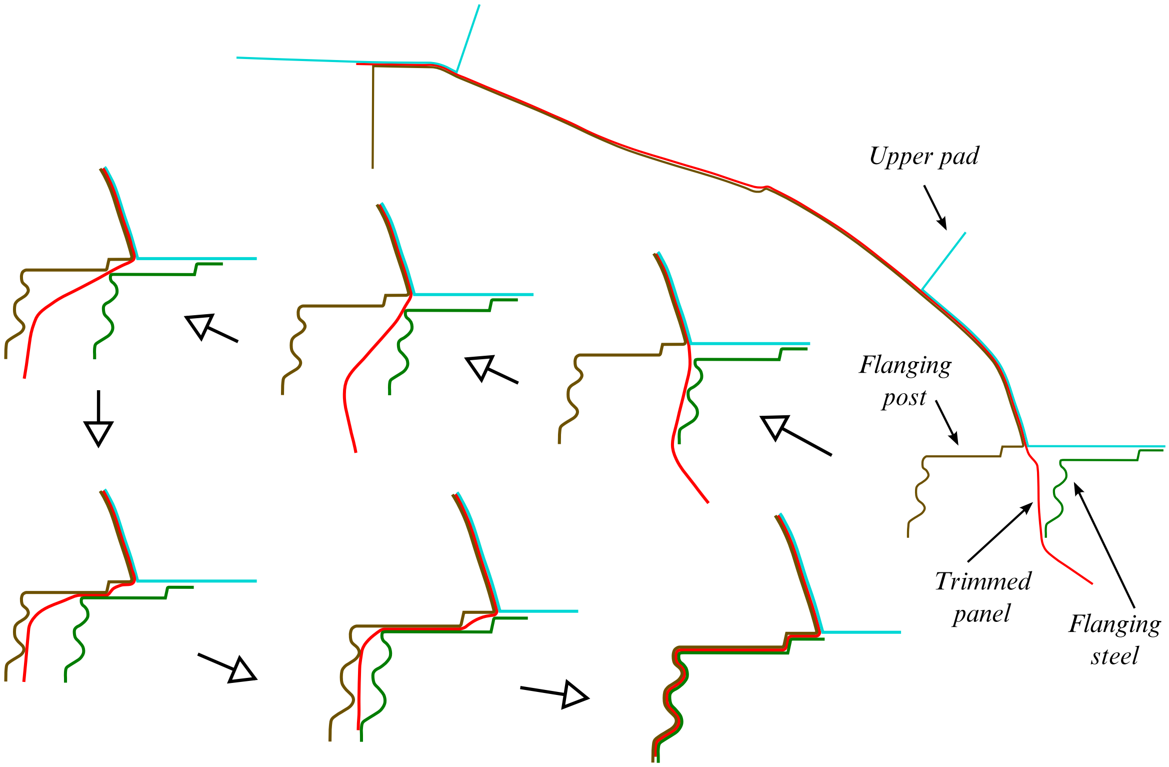


Figure 12-78. Flanging progression along section B (flanging post stationary).

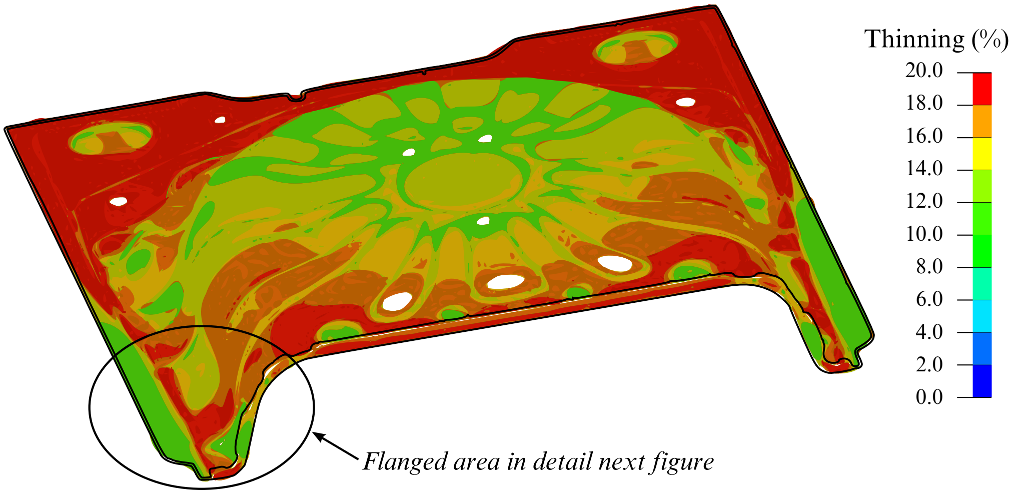


Figure 12-79. Flanging simulation of a rear floor pan using IOPTION 1 (Courtesy of Chrysler, LLC).

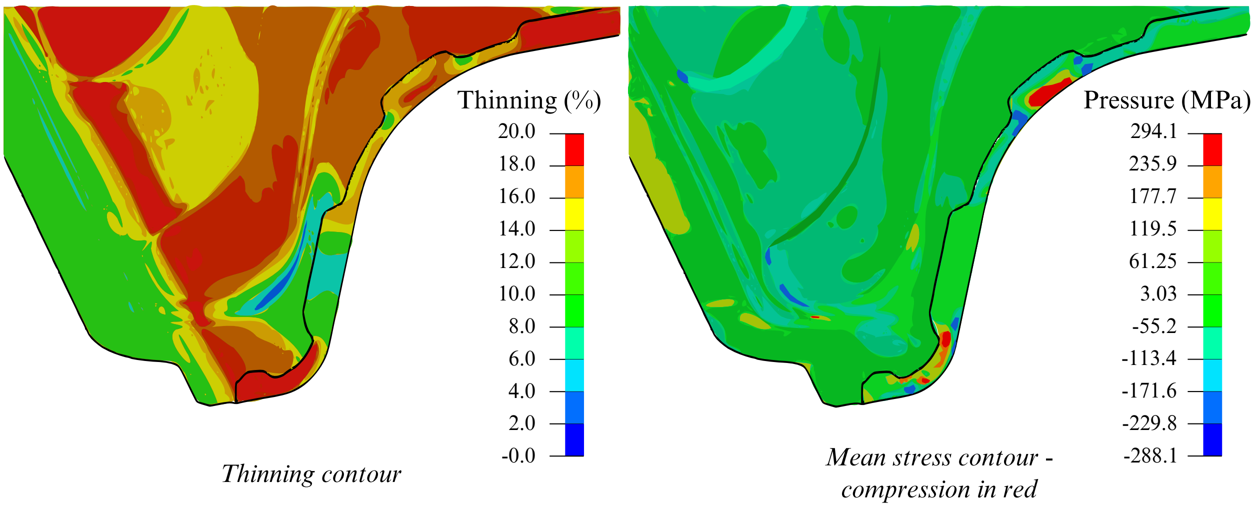


Figure 12-80. Localized view of the last figure.

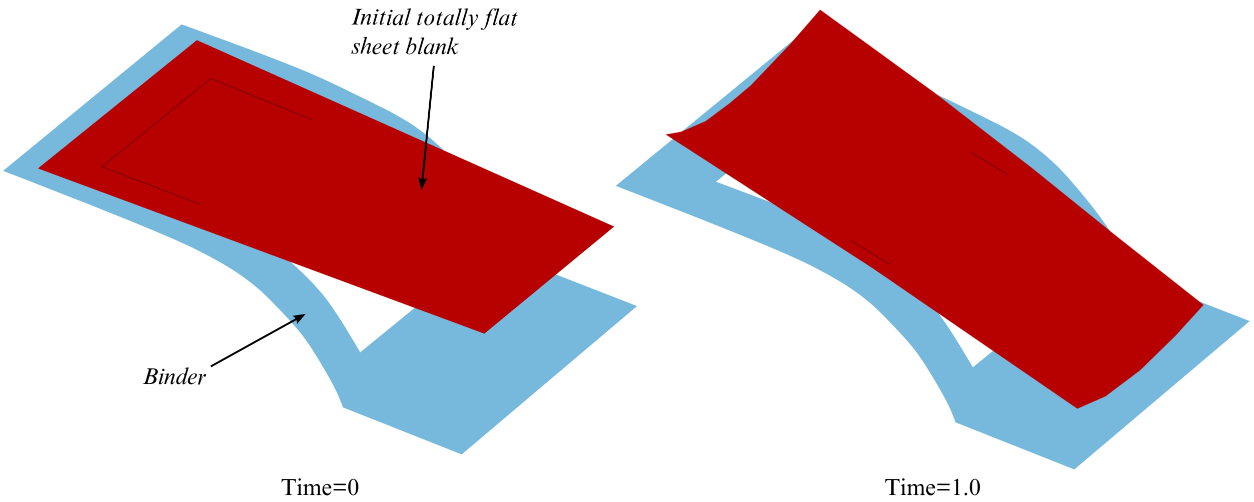


Figure 12-81. Test model (left) and gravity loaded blank (right) with switching from implicit dynamic to implicit static.

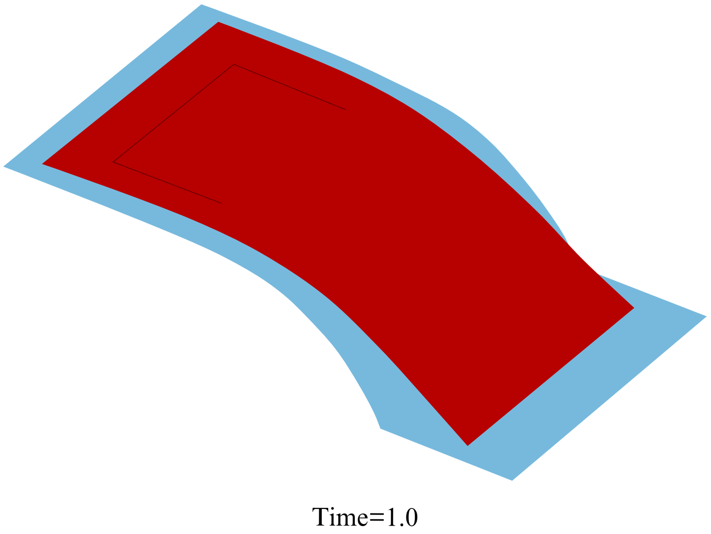


Figure 12-82. Gravity loaded blank without the "switching".

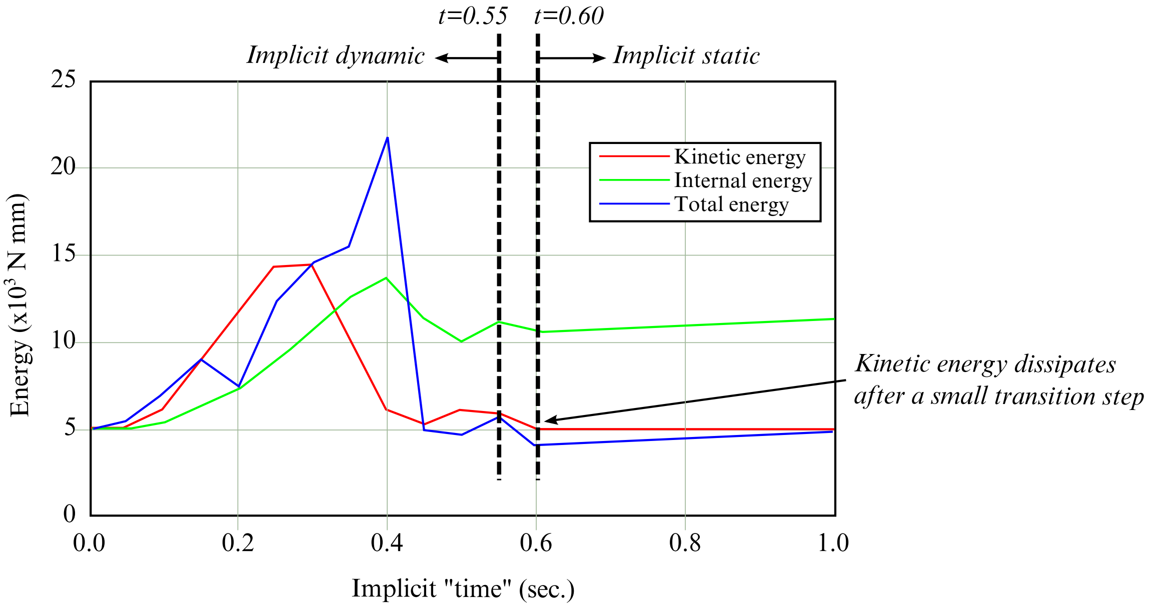


Figure 12-83. Switching between implicit dynamic and implicit static.

CONTROL_IMPLICIT_GENERAL_*{OPTION}

Availlable option s include:

<BLANK>

DYN

SPR

Purpose: Activate implicit analysis and define associated control parameters. This keyword is required for all implicit analyses. The DYN option allows setting controls specifically for the dynamic relaxation phase. The SPR option allows setting controls specifically for the springback phase.

Card 1	1	2	3	4	5	6	7	8
Variable	IMFLAG	DT0	IMFORM	NSBS	IGS	CNSTN	FORM	ZERO_V
Type	I	F	I	I	I	I	I	I
Default	0	none	2	1	2	0	0	0

VARIABLE **DESCRIPTION**

IMFLAG Implicit/Explicit analysis type flag

EQ.0: explicit analysis

EQ.1: implicit analysis

EQ.2: explicit followed by implicit, (seamless springback). *INTERFACE_SPRINGBACK_SEAMLESS is required to activate seamless springback.

EQ.4: implicit with automatic implicit-explicit switching

EQ.5: implicit with automatic switching and mandatory implicit finish

EQ.6: explicit with intermittent eigenvalue extraction

LT.0: curve ID = -IMGFLAG specifies IMFLAG as a function of time.

DT0 Initial time step size for implicit analysis

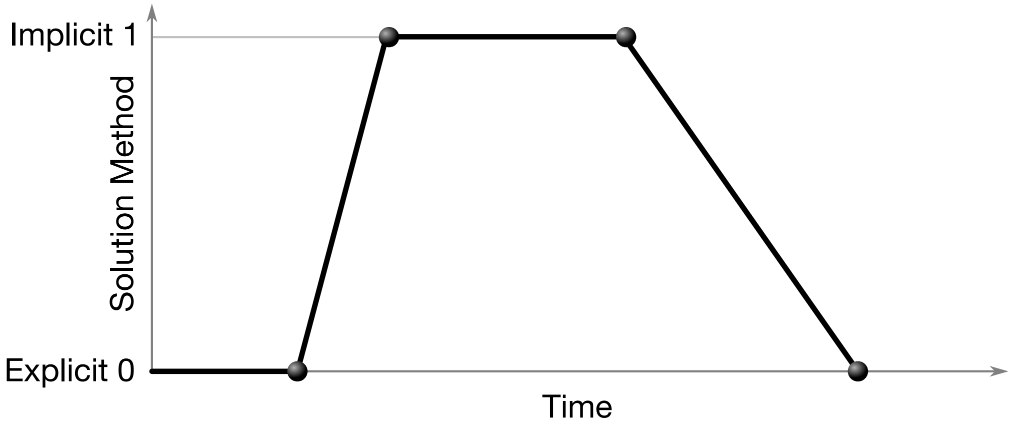


Figure 12-84. Solution method, implicit or explicit, controlled by a load curve.

VARIABLE	DESCRIPTION
IMFORM	Element formulation flag for seamless springback; see *INTERFACE_SPRINGBACK_SEAMLESS. EQ.1: switch to fully integrated shell formulation for springback EQ.2: retain original element formulation (default)
NSBS	Number of implicit steps in seamless springback; see *INTERFACE_SPRINGBACK_SEAMLESS.
IGS	Geometric (initial stress) stiffness flag EQ.1: include EQ.2: ignore
CNSTN	Indicator for consistent tangent stiffness (solid materials 3 & 115 only): EQ.0: do not use (default) EQ.1: use.
FORM	Fully integrated element formulation (IMFLAG = 2 and IMFORM = 1 only) EQ.0: type 16 EQ.1: type 6.

VARIABLE **DESCRIPTION**

ZERO_V

Zero out the velocity before switching from explicit to implicit.

EQ.0: The velocities are not zeroed out.

EQ.1: The velocities are set to zero.

Remarks:

VARIABLE **REMARK**

IMFLAG

The default value 0 indicates a standard explicit analysis will be performed. Using value 1 causes an entirely implicit analysis to be performed. Value 2 is automatically activated when the keyword *INTERFACE_SPRINGBACK_SEAMLESS is present, causing the analysis type to switch from explicit to implicit when the termination time is reached. Other nonzero values for IMFLAG can also be used with *INTERFACE_SPRINGBACK_SEAMLESS. After this switch, the termination time is extended by NSBS*DT0, or reset to twice its original value if DT0 = 0.0. The implicit simulation then proceeds until the new termination time is reached. Contact interfaces are automatically disabled during the implicit phase of seamless springback analysis. Furthermore, implicit stabilization (*CONTROL_IMPLICIT_STABILIZATION) and automatic step size adjustment (*CONTROL_IMPLICIT_AUTO) on by default for seamless springback.

When the automatic implicit-explicit switching option is activated (IMFLAG = 4 or 5), the solution method will begin as implicit. If convergence of the equilibrium iterations fails, the solution will automatically switch to explicit for a time interval of DTEXP (see *CONTROL_IMPLICIT_AUTO). After this time interval, the solution method will switch back to implicit and attempt to proceed. The implicit simulation may be either static or dynamic. When this feature is used in a static implicit job, simulation time is no longer arbitrary, and must be chosen along with DTEXP in a realistic way to allow efficient execution of any explicit phases. Mass scaling may also be activated (see *CONTROL_TIMESTEP), and will apply only during the explicit phases of the calculation. In cases where much switching occurs, users must exercise caution to ensure that negligible dynamic effects are introduced by the explicit phases.

When IMFLAG = 5, the final step of the simulation must be implicit. The termination time will be extended automatically as necessary, until a successfully converged implicit step can be

VARIABLE REMARK

obtained. This is useful for example in difficult metal forming springback simulations.

When IMFLAG = 6, an explicit simulation will be performed. Eigenvalues will be extracted intermittently according to a curve indicated by NEIG=(-curve ID) on *CONTROL_IMPLICIT_-EIGENVALUE. Beware that dynamic stress oscillations which may occur in the explicit simulation will influence the geometric (initial stress) stiffness terms used in the eigen solution, potentially producing misleading results and/or spurious modes. As an alternative, eigenvalues can also be extracted intermittently during an implicit analysis, using IMFLAG = 1 and NEIG=(-curve ID).

When IMFLAG < 0, a curve ID is indicated which gives the solution method as a function of time. Define a curve value of zero during explicit phases, and a value of one during implicit phases. Use steeply sloping sections between phases. An arbitrary number of formulation switches may be activated with this method. See Figure 12-84.

DT0

This parameter selects the initial time step size for the implicit phase of a simulation. The step size may change during a multiple step simulation if the automatic time step size control feature is active (see *CONTROL_IMPLICIT_AUTO.)

INFORM

Adaptive mesh must be activated when using element formulation switching. For best springback accuracy, use of shell type 16 is recommended during the entire stamping and springback analysis, in spite of the increased cost of using this element during the explicit stamping phase.

NSBS

The NSBS option allows a seamless springback analysis, invoked with *INTERFACE_SPRINGBACK_SEAMLESS, to use multiple unloading steps. Implicit seamless springback beings at time, $t = \text{ENDTIM}$ and finishes at $t = \text{ENDTIM} + \text{NSBS} \times \text{DT0}$ were ENDTIM is specified in *CONTROL_TERMINATION and DT0 is specified in *CONTROL_IMPLICIT_GENERAL.

IGS

The geometric stiffness adds the effect of initial stress to the global stiffness matrix. This effect is seen in a piano string whose natural frequency changes with tension. Geometric stiffness does not always improve nonlinear convergence, especially when

VARIABLE	REMARK
	compressive stresses are present, so its inclusion is optional. Furthermore, the geometric stiffness may lead to convergence problems with incompressible, or nearly incompressible, materials.

*CONTROL_IMPLICIT_INERTIA_RELIEF

Purpose: Allows analysis of linear static problems that have rigid body modes.

Card 1	1	2	3	4	5	6	7	8
Variable	IRFLAG	THRESH	IRCNT					
Type	I	F	I					
Default	0	0.001	0					

Additional Mode List Cards. This card should be included only when the user wants to specify the modes to use. Include as many cards as needed to provide all values. This input ends at the next keyword ("*") card. The mode numbers do not have to be consecutive.

Card 2	1	2	3	4	5	6	7	8
Variable	MODE1	MODE2	MODE3	MODE4	MODE5	MODE6	MODE7	MODE8
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
IRFLAG	Inertia relief flag EQ.0: do not perform inertia relief EQ.1: do perform inertia relief
THRESH	Threshold for what is a rigid body mode. The default is set to 0.001 Hertz where it is assumed that the units are in seconds.
IRCNT	The user can specify to use the lowest IRCNT modes instead of using THRESH to determine the number of modes.
MODEi	Ignore THRESH and IRCNT and use a specific list of modes, skipping those that should not be used.

*CONTROL_IMPLICIT_JOINTS

Purpose: Specify penalty or constraint treatment of joints for implicit analysis.

Card 1	1	2	3	4	5	6	7	8
Variable	ISPHER	IREVOL	ICYLIN					
Type	I	I	I					
Default	1	1	1					

VARIABLE	DESCRIPTION
ISPHER	Treatment of spherical joints EQ.1: use constraint method for all spherical joints (default) EQ.2: use penalty method for all spherical joints
IREVOL	Treatment of revolute joints EQ.1: use constraint method for all revolute joints (default) EQ.2: use penalty method for all revolute joints
ICYLIN	Treatment of cylindrical joints EQ.1: use constraint method for all cylindrical joints (default) EQ.2: use penalty method for all cylindrical joints

Remarks:

For most implicit applications one should use the constraint (default) method for the treatment of joints. When explicit-implicit switching is used the joint treatment should be consistent. This keyword allows the user to choose the appropriate treatment for their application.

*CONTROL_IMPLICIT_MODAL_DYNAMIC

Purpose: Activate implicit modal dynamic analysis. Eigenmodes are used to linearize the model by projecting the model onto the space defined by the eigenmodes. The eigenmodes can be computed or read from a file. All or some of the modes can be used in the linearization. Modal damping can be applied.

Card 1	1	2	3	4	5	6	7	8
Variable	MDFLAG	ZETA						
Type	I	F						

Optional Filename Card.

Card 2	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	A80							

VARIABLE	DESCRIPTION
MDFLAG	Modal Dynamic flag EQ.0: no modal dynamic analysis EQ.1: perform modal dynamic analysis.
ZETA	Modal Dynamic damping constant.
FILENAME	If specified the eigenmodes are read from the specified file. Otherwise the eigenmodes are computed as specified on *CONTROL_IMPLICIT_EIGENVALUE.

Remarks:

Modal Dynamic uses the space spanned by the eigenmodes of the generalized eigenvalue problem

$$\mathbf{K}\boldsymbol{\phi}_i = \lambda_i\mathbf{M}\boldsymbol{\phi}_i.$$

The matrix of eigenmodes, $\mathbf{\Phi}$, diagonalizes $\mathbf{K}$ and $\mathbf{M}$

$$\mathbf{\Phi}^{\mathbf{T}}\mathbf{K}\mathbf{\Phi} = \mathbf{\Lambda}$$

and

$$\mathbf{\Phi}^{\mathrm{T}}\mathbf{M}\mathbf{\Phi} = \mathbf{I}.$$

Multiplication by $\boldsymbol{\Phi}$ changes coordinates from amplitude space to displacement space as

$$\mathbf{u} = \mathbf{\Phi a}$$

where $\boldsymbol{a}$ is a vector of modal amplitudes. The equations of motion

$$\mathbf{M}\ddot{\mathbf{u}}^{n+1} + \mathbf{K}\mathbf{\Delta u} = \mathbf{F}(\mathbf{x}^{\boldsymbol{n}})$$

when multiplied on the left by $\boldsymbol{\phi}^{\mathrm{T}}$ and substituting $\mathbf{u} = \mathbf{\Phi a}$ become the linearized equations of motion in its spectral form as,

$$\mathbf{I}\ddot{\mathbf{a}}^{n+1} + \mathbf{\Lambda}(\mathbf{\Delta a}) = \mathbf{\Phi}^{\mathrm{T}}\mathbf{F}(\mathbf{x}^{n}).$$

The modal damping features adds a velocity dependent damping term,

$$\mathbf{I}\ddot{\mathbf{a}}^{n+1} + 2\mathbf{Z}\dot{\mathbf{a}}^{n} + \mathbf{\Lambda}(\mathbf{\Delta a}) = \mathbf{\Phi}^{\mathrm{T}}\mathbf{F}(\mathbf{x}^{\boldsymbol{n}})$$

Where $Z_{ii} = \zeta_i\omega_i$, $\omega_i = \sqrt{\lambda_i}$, and each ζ_i is a user specified damping coefficients.

The matrices in the reduced equations are diagonal and constant. So Modal Dynamics can quickly compute the acceleration of the amplitudes and hence the motion of the model. But the motion is restricted to the space spanned by the eigenmodes.

Eigenmodes are either computed based on *CONTROL_IMPLICIT_EIGENVALUE or read from file FILENAME. By default all modes are used in the projection. Selected modes can be specified via *CONTROL_IMPLICIT_MODAL_DYNAMIC_MODE to reduce the size of the projection. .

Stresses are computed only for linear shell formulation 18 and linear solid formulation 18.

Modal damping on all modes can be specified using ZETA. More options for specifying modal damping can be found on *CONTROL_IMPLICIT_MODAL_DYNAMIC_DAMPING.

Using MDFLAG = 1, ZETA = 0.0, and FILENAME = “ ” is the same as using IMASS = 2 with *CONTROL_IMPLICIT_DYNAMICS. Using MDFLAG = 1, ZETA = 0.0 and FILENAME = ’d3eigv’ is the same as IMASS = 3. The new keywords *CONTROL_IMPLICIT_MODAL_DYNAMIC_MODE and *CONTROL_IMPLICIT_MODAL_DYNAMIC_-DAMPING provide additional user options for mode selection and modal damping.

CONTROL_IMPLICIT_MODAL_DYNAMIC_DAMPING_*{OPTION}

Available options include:

BLANK

SPECIFIC

FREQUENCY_RANGE

Purpose: Define vibration modes to be used in implicit modal dynamic.

Damping Card. Card for option set to <BLANK>.

Card 1	1	2	3	4	5	6	7	8
Variable	ZETA1							
Type	F				I			

Specific Damping Cards. Cards for the SPECIFIC option. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	MID1	ZETA1	MID2	ZETA2	MID3	ZETA3	MID4	ZETA4
Type	I	F	I	F	I	F	I	F

Frequency Range Damping Cards. Cards for FREQUENCY_RANGE option. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	FREQ1	ZETA1	FREQ2	ZETA2	FREQ3	ZETA3	FREQ4	ZETA4
Type	F	F	F	F	F	F	F	F

VARIABLE	DESCRIPTION
ZETAn	Modal Dynamic damping coefficient n.
MIDn	Mode ID n.

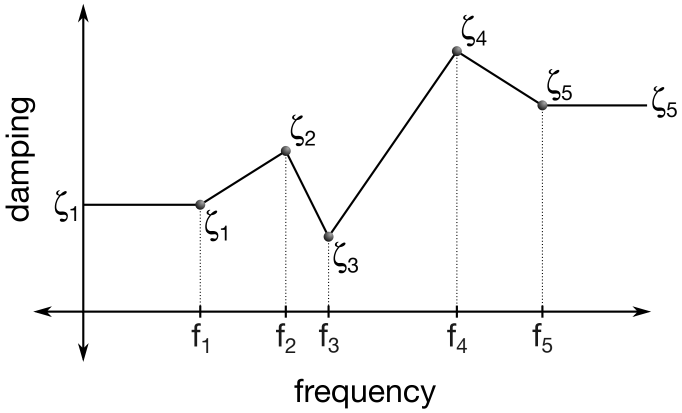


Figure 12-85. Schematic illustration of frequency range damping.

VARIABLE	DESCRIPTION
FREQn	Frequency value n.

Remarks:

1. If no option is specified the value of ZETA1 becomes the damping coefficient for all modes involved in implicit modal dynamic analysis. This value overrides the value on *CONTROL_IMPLICIT_MODAL_DYNAMIC.

2. If option SPECIFIC is specified the integers MIDn indicate which modes involved in *CONTROL_IMPLICIT_MODAL_DYNAMIC will have modal damping applied to them. The associated value ZETAn will be the modal damping coefficient for that mode.

3. If option FREQUENCY_RANGE is specified all modes involve will have modal damping applied. The damping coefficient will be computed by linear interpolation of the pairs (FREQi, ZETAi). If the modal frequency is less than FREQ1 then the modal damping coefficient will be ZETA1. If the modal frequency is greater than FREQn then the modal damping coefficient will be ZETAn. The values of FREQi must be specified in ascending order.

CONTROL_IMPLICIT_MODAL_DYNAMIC_MODE_*OPTION

Available options include:

LIST

GENERATE

Purpose: Define vibration modes to be used in implicit modal dynamic.

Mode ID Cards. Card 1 for the LIST keyword option. For each mode include an addition. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	MID1	MID2	MID3	MID4	MID5	MID6	MID7	MID8
Type	I	I	I	I	I	I	I	I

Mode Range Cards. Card 1 for the GENERATE keyword option. For each range of modes include an additional card. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	M1BEG	M1END	M2BEG	M2END	M3BEG	M3END	M4BEG	M4END
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
MID*n*	Mode ID *n*.
M*n*BEG	First mode ID in block *n*.
M*n*END	Last mode ID in block *n*. All mode ID's between and including M*n*BEG and M*n*END are added to the list.

Remarks:

1. User may use this keyword with *CONTROL_IMPLICIT_MODAL_DYNAMIC if some of the vibration modes have less contribution to the total structural response and can be removed from the implicit modal dynamic analysis.

CONTROL_IMPLICIT_MODES_*{OPTION}

Available options include:

<BLANK>

BINARY

Purpose: Request calculation of constraint, attachment, and/or eigenmodes for later use in modal analysis using *PART_MODES (see also *CONTROL_IMPLICIT_GENERAL) or *ELEMENT_DIRECT_MATRIX_INPUT.

Card 1	1	2	3	4	5	6	7	8
Variable	NSIDC	NSIDA	NEIG	IBASE	SE_MASS	SE_DAMP	SE_STIFF	SE_INERT
Type	I	I	I	I	C	C	C	C
Default	0	0						

Card 2	1	2	3	4	5	6	7	8
Variable	SE_FILENAME							
Type	C							

VARIABLE	DESCRIPTION
NSIDC	Node set ID for constraint modes EQ.0: no constraint modes will be generated
NSIDA	Node set ID for attachment modes EQ.0: no attachment modes will be generated
NEIG	Number of eigenmodes (normal modes) EQ.0: no eigenmodes will be generated
IBASE	Offset for numbering of the generalized internal degrees of freedom for the superelement

VARIABLE	DESCRIPTION
SE_MASS	Name of the superelement mass matrix. If left blank it is not generated.
SE_DAMP	Name of the superelement damping matrix. If left blank it is not generated.
SE_STIFF	Name of the superelement stiffness matrix. If left blank it is not generated.
SE_INERT	Name of the superelement inertia matrix, required for gravity loading applications of the superelement. If left blank it is not generated.
SE_FILENAME	If any of SE_MASS, SE_DAMP, SE_STIFF, or SE_INERT are not blank then the second line is required and contains the file name for the superelement.

Remarks:

To use this feature, an implicit analysis must be requested using IMFLAG = 1 on *CONTROL_IMPLICIT_GENERAL, and a non-zero termination time must be specified on *CONTROL_TERMINATION. A double precision version of LS-DYNA should be used for best accuracy. Care must be taken to apply a sufficient number of constraints to the model to eliminate static rigid body motion. Computed modes are written to binary output file d3mode, with the order of output being constraint modes, followed by attachment modes, and then eigenmodes. The d3mode file can be read and modes viewed using LS-PrePost. Eigenmodes are also written to binary output file d3eigv.

Constraint and attachment modes are generated by applying unit displacements and unit forces, respectively, to each specified degree of freedom. By default, modes are computed for all degrees of freedom for each node in sets NSIDC and NSIDA. The first and second node set attribute parameters can be optionally used to restrict the translational and rotational degrees of freedom for which modes are requested, respectively, according to the following syntax:

Node set attribute parameters DA1 and A1: translational degree of freedom codes

Node set attribute parameters DA2 and A2: rotational degree of freedom codes

code	modes computed
0	(See note below.)
1	X degree of freedom only
2	Y degree of freedom only

3	Z degree of freedom only
4	X, Y degrees of freedom only
5	Y, Z degrees of freedom only
6	X, Z degrees of freedom only
7	X, Y, Z degrees of freedom

Setting both node set attributes to zero is equivalent to setting both node set attributes to 7 (X, Y, and Z for translational and rotational degrees of freedom).

If one node set attribute is nonzero (codes 1 to 7) and the other node set attribute is zero, then the zero attribute means NO degrees of freedom are considered. For example, if DA1 = 2 and DA2 = 0, then only the Y-translational degree of freedom modes are calculated.

Eigenmodes are generated for the model with single point constraints applied on the constraint modes. The number of eigenmodes is specified here. If the user wants to compute eigenmodes other than the lowest ones, the controls on *CONTROL_IMPLICIT_EIGENVALUE can be used.

When the superelement is created an internal numbering must be applied to the attachment and eigen modes. This numbering starts at IBASE+1.

The user can create the superelement representation of the reduced model by specifying the SE_MASS, SE_DAMP, SE_STIFF, SE_INERT and SE_FILENAME fields. The inertia matrix is necessary if body forces, e.g., gravity loads, are applied to the superelement. The file, by default is written in the Nastran DMIG file format and can be used as input to *ELEMENT_DIRECT_MATRIX_INPUT. The BINARY keyword option can be used to create a binary representation for the superelement which can be used with *ELEMENT_DIRECT_MATRIX_INPUT_BINARY to reduce the file size.

The combination of constraint modes and eigenmodes form the Hurty-Craig-Bampton linearization for a model. Using only constraint modes is the same as static condensation.

Some broad guidelines for appropriate selection of constraint modes, attachment modes, and eigenmodes include:

1. Use constraint modes for the nodal degrees-of-freedom that are to be "constrained" with SPCs or prescribed motion.

2. Use attachment modes for nodal degrees-of-freedom that are under the influence of point loads.

3. Use eigenmodes in the construction of the superelement to capture the reaction of the part being modeled by the superelement and the associated feedback to the rest of the model.

*CONTROL_IMPLICIT_ROTATIONAL_DYNAMICS

Purpose: This keyword is used to model rotational dynamics using the implicit time integrator. Applications for this feature include the transient and vibration analysis of rotating parts such as turbine blades, propellers in aircraft, and rotating disks in hard disk drives. The current implementation *requires* a double-precession SMP version of LS-DYNA. An MPP implementation is under development.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	STYPE	OMEGA	VID	NOMEG	IREF	OMEGADR	
Type	I	I	F	I	I	I	F	
Default	none	0	none	none	0	0	0	

Additional Rotational Speed Cards. This card should be included only when NOMEG > 0. Include as many cards as needed to provide all NOMEG values. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	OMEG1	OMEG2	OMEG3	OMEG4	OMEG5	OMEG6	OMEG7	OMEG8
Type	F	F	F	F	F	F	F	F

VARIABLE **DESCRIPTION**

SID Set ID of the rotational components.

STYPE Set type:

EQ.0: Part;

EQ.1: Part set.

OMEGA Rotating speed.

GT.0: rotating speed.

LT.0: curve ID = (-OMEGA) gives rotating speed as a function of time.

VARIABLE **DESCRIPTION**

VID

Vector ID to define the rotating axis. It can be defined in *DEFINE_VECTOR and *DEFINE_VECTOR_NODES, and the tail of the vector should be set as the rotating center.

NOMEG

Number of rotating speeds. This feature is intended to automatically preform parameter studies with respect to the rotation speed. The keyword *CONTROL_IMPLICIT_EIGENVALUE must be included if NOMEG > 0.

IREF

Reference frame:

EQ.0: Rotating coordinate system and rotating parts will not rotate in visualization. Solid element and thick shell element will use IREF = 0.

EQ.1: Fixed coordinate system. Rotating parts rotates and the initial rotating velocity should be defined in *INITIAL_VELOCITY_GENERATION as well.

EQ.2: Rotating coordinate system, but rotate rotating parts for visualization purpose.

OMEG*n*

The *n*th rotating speed.

OMEGADR

Rotating speed defined in dynamic relaxation.

GT.0: rotating speed defined in dynamic relaxation.

LT.0: curve ID = (-OMEGA) gives rotating speed as a function of time.

Remarks:

The linearized equilibrium equation in the rotating coordinate system is given by

$$\mathbf{M}\ddot{\mathbf{u}} + (\mathbf{D} + 2\Omega\mathbf{C})\dot{\boldsymbol{u}} + (K - \Omega^2\mathbf{K}_G)\mathbf{u} = \mathbf{F}$$

Whereas, in a fixed coordinate system, the linearized equilibrium equation is

$$\mathbf{M}\ddot{\mathbf{u}} + (\mathbf{D} + \Omega\mathbf{C})\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{F}$$

with

$\mathbf{M}$ = lumped mass matrix

$\mathbf{D}$ = damping matrix

$\mathbf{K}$ = stiffness matrix

$\mathbf{C}$ = gyroscopic matrix

$\mathbf{K}_G$ = centrifugal stiffness matrix

$\mathbf{u}$ = nodal displacement vector

$\dot{\mathbf{u}}$ = nodal point velocities at time

$\ddot{\mathbf{u}}$ = nodal point acceleration at time

Ω = rotating speed

The chief difference between the equations for the rotating and fixed frames is the inclusion of the centrifugal stiffness matrices$\mathbf{K}_g$. Additionally, the coefficient on the gyroscopic matrix, **C**, as well as its content are modified in the rotating-frame case. Specifically, the rotating system includes an additional Coriolis contribution to **C**.

In many applications of rotational dynamics, the critical speed – the theoretical angular velocity that excites the natural frequency of a rotating object – is of particular concern. Therefore, the study of mode frequency response with the change of the rotating speed is very important. The Campbell diagram, which is defined to represent a system's eigen-frequencies as a function of rotating speeds, is introduced for this purpose. In order to do this, the user needs to define a set of rotating speeds on card 2, and LS-DYNA will do modal analysis for each of these speeds. NOMEG should be defined as the number of rotating speeds used in card 2. A keyword file example in this application can be set as follows:

```
*KEYWORD
*CONTROL_TERMINATION...
*CONTROL_IMPLICIT_EIGENVALUE
         5

*CONTROL_IMPLICIT_GENERAL
         1      0.05
*CONTROL_IMPLICIT_ROTATIONAL_DYNAMICS
$#     SID     STYPE     OMEGA       VID    NOMEGA      IREF
         1         0       0.0         1         4         1
$#   OMEG1     OMEG2     OMEG3     OMEG4
      50.0     100.0     150.0     200.0
*DEFINE_VECTOR
$#     VID        XT        YT        ZT        XH        YH        ZH       CID
         1       0.0       0.0       0.0       1.0       0.0       0.0
*DATABASE_...
*PART...
*SECTION...
*MAT...
*ELEMENT...
*NODE...
*END
```

Besides of modal analysis, transient analysis can also be done using this keyword. A keyword file example can be set as follows:

```
*KEYWORD
*CONTROL_TERMINATION...
*CONTROL_IMPLICIT_GENERAL
         1      0.05
*CONTROL_IMPLICIT_ROTATIONAL_DYNAMICS
$#     SID     STYPE     OMEGA       VID    NOMEGA      IREF
         1         0       0.0         1         0         0
```

```
*DEFINE_VECTOR
$#     VID        XT         YT         ZT         XH         YH         ZH        CID
         1       0.0        0.0        0.0        1.0        0.0        0.0
*DATABASE_...
*PART...
*SECTION...
*MAT...
*ELEMENT...
*NODE...
*END
```

CONTROL_IMPLICIT_SOLUTION_*{OPTION}

Available options include:

<BLANK>

DYN

SPR

Purpose: These optional cards apply to implicit calculations. Use these cards to specify whether a linear or nonlinear solution is desired. Parameters are also available to control the implicit nonlinear and arc length solution methods (see also *CONTROL_-IMPLICIT_GENERAL). The DYN option allows setting controls specifically for the dynamic relaxation phase. The SPR option allows setting controls specifically for the springback phase.

Card 1	1	2	3	4	5	6	7	8
Variable	NSOLVR	ILIMIT	MAXREF	DCTOL	ECTOL	RCTOL	LSTOL	ABSTOL
Type	I	I	I	F	F	F	F	F
Default	12	11	15	0.001	0.01	10^{10}	0.90	10^{-10}

Remaining cards are optional.†

Optional 2a	1	2	3	4	5	6	7	8
Variable	DNORM	DIVERG	ISTIF	NLPRINT	NLNORM	D3ITCTL	CPCHK	
Type	I	I	I	I	F/I	I	I	
Default	2	1	1	0	2	0	0	

Strict Tolerances Optional Card. Define this card if and only if DNORM.LT.0

Optional 2b	1	2	3	4	5	6	7	8
Variable	DMTOL	EMTOL	RMTOL		NTTOL	NRTOL	RTTOL	RRTOL
Type	F	F	F		F	F	F	F
Default	0.0	0.0	0.0		0.0	0.0	0.0	0.0

Arc Length Optional Card. The contents of this card are ignored unless an arc-length method is activated (6 ≤ NSOLVR ≤ 9, or NSOLVR = 12 and ARCMTH = 3).

Optional 3	1	2	3	4	5	6	7	8
Variable	ARCCTL	ARCDIR	ARCLEN	ARCMTH	ARCDMP	ARCPSI	ARCALF	ARCTIM
Type	I	I	F	I	I	F	F	F
Default	0	none	0	1	2	0.	0.	0.

Line Search Parameter Optional Card.

Optional 4	1	2	3	4	5	6	7	8
Variable	LSMTD	LSDIR	IRAD	SRAD	AWGT	SRED		
Type	I	I	F	F	F	F		
Default	4	2	0.0	0.0	0.0	0.0		

VARIABLE **DESCRIPTION**

NSOLVR

Solution method for implicit analysis:

EQ.1: Linear

EQ.12: Nonlinear with BFGS updates + optional arclength, (default) incorporating different line search and integration schemes compared to solver 2.

EQ.2: Nonlinear with BFGS updates (obsolete)

VARIABLE	DESCRIPTION
	EQ.3: Nonlinear with Broyden updates EQ.4: Nonlinear with DFP updates EQ.5: Nonlinear with Davidon updates EQ.6: Nonlinear with BFGS updates + arclength EQ.7: Nonlinear with Broyden updates + arclength EQ.8: Nonlinear with DFP updates + arclength EQ.9: Nonlinear with Davidon updates + arclength
ILIMIT	Iteration limit between automatic stiffness reformations
MAXREF	Stiffness reformation limit per time step. LT.0: If |MAXREF| matrix reformations occur convergence for that time step is forced, see REMARKS.
DCTOL	Displacement relative convergence tolerance
ECTOL	Energy relative convergence tolerance
RCTOL	Residual (force) relative convergence tolerance (DEFAULT = inactive)
LSTOL	Line search convergence tolerance
ABSTOL	Absolute convergence tolerance. LT.0: Convergence detected when the residual norm is less than –ABSTOL (**Tip**: To drive convergence based on –ABSTOL, set DCTOL and ECTOL to 1.0E-20)
DNORM	Displacement norm for convergence test EQ.1: Increment vs. displacement over current step EQ.2: Increment vs. total displacement (default) If DNORM.LT.0, this is to be interpreted as its absolute value, but activates reading of optional card 2b.
DIVERG	Divergence flag (force imbalance increase during equilibrium iterations) EQ.1: reform stiffness if divergence detected (default)

VARIABLE **DESCRIPTION**

EQ.2: ignore divergence

ISTIF

Initial stiffness formation flag

EQ.1: reform stiffness at start of each step (default)

EQ.n: reform stiffness at start of every "n"th step

NLPRINT

Nonlinear solver print flag

EQ.0: no nonlinear iteration information printed (new v970 default)

EQ.1: print iteration information to screen, message, d3hsp files

EQ.2: print extra norm information (NLNORM = 1)

EQ.3: same as 2, but also print information from line search

NOTE: during execution, interactive commands can be used:

interactive command	response
<ctrl-c> **nlpr**int	toggle NLPRINT between 0 and 1
<ctrl-c> **diag**nostic	toggle NLPRINT between 0 and 2
<ctrl-c> **info**rmation	set NLPRINT = 2 for one iteration

NLNORM

Nonlinear convergence norm type, input an integer if zero or a positive number is used and float if a negative value is used

LT.0: Same as 4, but rotational degrees of freedom are scaled appropriately with characteristic length |NLNORM| to account for units. See remarks.

EQ.1: consider translational and rotational degrees of freedom

EQ.2: consider translational degrees of freedom only (default)

EQ.4: consider sum of translational and rotational degrees of freedom, i.e., no separate treatment. See remarks.

D3ITCTL

Control d3iter database. If nonzero, the search directions for the nonlinear implicit solution are written to the d3iter database. To reduce the size of the d3iter database the database is reset every n time steps where n = d3itctl.

CPCHK

Contact penetration check flag. This flag does not apply to mortar contacts.

EQ.0: no contact penetration check is performed (default).

VARIABLE **DESCRIPTION**

EQ.1: check for contact penetration during the nonlinear solution procedure. If such penetration is found modify the line search to prevent unnecessary penetration.

DMTOL Maximum displacement convergence tolerance, convergence is detected when the *relative* maximum nodal or rigid body displacement is less than this value

EMTOL Maximum energy convergence tolerance, convergence is detected when the *relative* maximum nodal or rigid body energy increment is less than this value

RMTOL Maximum residual convergence tolerance, convergence is detected when the *relative* maximum nodal or rigid body residual is less than this value

NTTOL Nodal translational convergence tolerance, convergence is detected when the *absolute* maximum nodal translational residual is less than this value

NRTOL Nodal rotational convergence tolerance, convergence is detected when the *absolute* maximum nodal rotational residual is less than this value

RTTOL Rigid body translational convergence tolerance, convergence is detected when the *absolute* maximum rigid body translational residual is less than this value

RRTOL Rigid body rotational convergence tolerance, convergence is detected when the *absolute* maximum rigid body rotational residual is less than this value

ARCLEN Relative arc length size. See remarks below.

LE.0.0: use automatic size,

GT.0.0: use ARCLEN × (automatic step size).

ARCMTH Arc length method

EQ.1: Crisfield (default)

EQ.2: Ramm

EQ.3: Modified Crisfield (used with NSOLVR = 12 only)

ARCDMP Arc length damping option

EQ.2: off (default)

EQ.1: on, oscillations in static solution are suppressed

ARCPSI

Relative influence of load/time parameter in spherical arclength constraint, default value is 0 which corresponds to a cylindrical arclength constraint. Applies to ARCMTH = 3.

ARCALF

Relative influence of predictor step direction for positioning of the arc center, default is 0 which means that the center is at the origin. Applies to ARCMTH = 3.

ARCTIM

Optional time when arclength method is initiated. Applies to ARCMTH = 3.

LSMTD

Line search convergence method:

EQ.1: Energy method using only translational variables

EQ.2: Residual method

EQ.3: Energy method using both translational and rotational variables

EQ.4: Energy method using sum of translational and rotational degrees of freedom (default), i.e., no separate treatment

EQ.5: Same as 4, but account for residual norm growth to be extra conservative in step length

EQ.6: Same as 5, but minimizes the residual norm whenever convenient.

LSDIR

Line search direction method:

EQ.1: Search on all variables (traditional approach used in versions prior to 971)

EQ.2: Search only on the independent (unconstrained) variables

EQ.3: Use adaptive line search (see AWGT, SRED)

EQ.4: Use curved line search (see IRAD, SRAD)

IRAD

Normalized curvature factor for curved line search, where 0 indicates a straight line search and 1 indicates full curved line search.

SRAD

Radius of influence for determining curve in curved line search.

VARIABLE **DESCRIPTION**

For each independent node, all nodes within this radius are used for determining the curve. If 0, then all nodes connected to the same element as the independent node are used.

AWGT Adaptive line search weight factor between 0 and 1. A high value tends to restrict the motion of oscillating nodes during the implicit process.

SRED Initial step reduction between 0 and 1 for adaptive line search, use large number for conservative start in implicit procedure.

Remarks:

VARIABLE **REMARKS**

NSOLVR If a linear analysis is selected, equilibrium checking and iterations are not performed.

The Full Newton nonlinear solution method can be invoked by using the default BFGS solver, and selecting ILIMIT = 1 to form a new stiffness matrix every iteration.

In the neighborhood of limit points the Newton based iteration schemes often fail. The arc length method of Riks and Wempner (combined here with the BFGS method) adds a constraint equation to limit the load step to a constant "arc length" in load-displacement space. This method is frequently used to solve snap through buckling problems. When applying the arc-length method, the curves that define the loading should contain only two points, and the first point should be at the origin (0,0). LS-DYNA will extrapolate, if necessary, to determine the load. In this way, time and load magnitude are related by a constant. It is possible that time can become negative in case of load reversal. The arc length method cannot be used in a dynamic analysis.

ILIMIT In the default BFGS method, the global stiffness matrix is only reformed every ILIMIT iterations. Otherwise, an inexpensive stiffness update is applied. By setting ILIMIT = 1, a stiffness reformation is performed every iteration. This is equivalent to the Full Newton method (with line search). A higher value of ILIMIT (20-25) can reduce the number of stiffness matrix reformations and factorizations which may lead to a significant reduction in cost. Note that the storage requirements for

VARIABLE REMARKS

implicit include storing 2 vectors per iteration. Large values of ILIMIT will cause substantial increase in storage requirements.

MAXREF

The nonlinear equilibrium search will continue until the stiffness matrix has been reformed |MAXREF| times, with ILIMIT iterations between each reformation. If equilibrium has not been found and MAXREF > 0, control will be passed to the automatic time step controller if it is activated. If the automatic time step controller is not active error termination will result. When the auto time step controller is active, it is often efficient to choose MAXREF = 5 and try another stepsize quickly, rather than wasting too many iterations on a difficult step.

When MAXREF < 0 and |MAXREF| matrix reformations have occurred convergence for the current time step is declared, with a warning, and the simulation moves to the next time step. This option should be used with caution as the results for that particular time step may be wrong.

DCTOL

When the displacement norm ratio is reduced below DCTOL, this condition is satisfied. Smaller numbers lead to more accurate determination of equilibrium and, on the negative side, result in more iterations and higher costs. Use NLPRINT to display norm data each iteration.

ECTOL

When the energy norm ratio is reduced below ECTOL, this condition is satisfied. Smaller numbers lead to more strict determination of equilibrium and, on the negative side, result in more iterations and higher costs. Use NLPRINT to display norm data each iteration.

RCTOL

When the residual norm ratio is reduced below RCTOL, this condition is satisfied. Smaller numbers lead to more strict determination of equilibrium and, on the negative side, result in more iterations and higher costs. By default this convergence criterion is effectively disabled using RCTOL = 1.e10. Use NLPRINT to display norm data each iteration.

DMTOL, etc.

For all nonzero values of the strict tolerance parameters in optional card 2b, the associated criterion must be satisfied *in addition* to the ones defined through DCTOL, ECTOL and RCTOL. These criteria are based on the maximum norm, which is regarded as stronger than the Euclidian norm used for the other parameters, and using them will likely result in higher

VARIABLE **REMARKS**

accuracy at the price of more iterations. For NLPRINT.GE.2 a table is listed in the message and d3hsp for each iteration, providing the values associated with all the criteria activated. The first three (DMTOL, EMTOL and RMTOL) of these extra parameters are unitless and honor the meaning of both DNORM and NLNORM. The last four (NTTOL, NRTOL, RTTOL and RRTOL) are to be given in units force, torque, force and torque, respectively, and the values used should account for the representative loads in the problem as well as the discretization size.

LSTOL

A line search is performed on stiffening systems to guard against divergence of Newton-based nonlinear solvers. With the Full Newton method, it is sometimes helpful to define a large value (LSTOL = 9999.0) to effectively disable line search.

DNORM

When computing the displacement ratio, the norm of the incremental displacement vector is divided by the norm of “total” displacement. This “total” displacement may be either the total over the current step, or the total over the entire simulation. The latter tends to be more lax, and can be poor at the end of simulations where large motions develop. For these problems, an effective combination is DNORM = 1, and DCTOL = 0.01 or larger.

DIVERGE

By default, a new stiffness matrix is formed whenever divergence (growing out-of-balance force) is detected. This flag can be used to suppress this stiffness reformation.

ISTIF

By default, a new stiffness matrix is formed at the start of every time step. Suppressing this stiffness reformation can decrease the cost of simulations which have many tiny steps that are mostly linear, such as transient dynamics.

NLPRINT

This flag controls printing of displacement and energy convergence measures during the nonlinear equilibrium search. If convergence difficulty occurs, this information is helpful in determining the problem.

NLNORM

By default, only translational degrees of freedom are used in evaluating convergence norms. Use this flag to include rotational degrees of freedom, or to make additional data available for diagnosing convergence problems.

VARIABLE REMARKS

This additional data includes the worst offending node and degree of freedom contributing to each norm. Rotational degrees of freedom can be considered independently from the translational degrees of freedom, meaning that two separate scalar products are used for evaluating norms, $\langle \mathbf{u}, \mathbf{v} \rangle_{\mathrm{t}} = \mathbf{u}^{\mathrm{T}} \mathbf{J}_{\mathrm{t}} \mathbf{v}$ and $\langle \mathbf{u}, \mathbf{v} \rangle_{\mathrm{r}} = \mathbf{u}^{\mathrm{T}} \mathbf{J}_{\mathrm{r}} \mathbf{v}$. Here $\mathbf{J}_{\mathrm{t}}$ and $\mathbf{J}_{\mathrm{r}}$ are diagonal matrices with ones on the diagonal to extract the translational and rotational degrees of freedom, respectively. This is the option NLNORM = 1, and the convergence criteria must be satisfied for both translational and rotational degrees of freedom simultaneously.

Alternatively they can be included by defining the single scalar product $\langle \mathbf{u}, \mathbf{v} \rangle = \langle \mathbf{u}, \mathbf{v} \rangle_{\mathrm{t}} + \lambda_u \lambda_v \langle \mathbf{u}, \mathbf{v} \rangle_{\mathrm{r}}$, where λ_u and λ_v are scale factors to account for different units of the rotational degrees of freedom. For NLNORM = 4 these scale factors are equal to 1, but for NLNORM < 0 λ_u is equal to |NLNORM| if **u** is a displacement vector and $|\mathrm{NLNORM}|^{-1}$ if it is a force vector, and the same goes for the pair λ_v and **v**. So |NLNORM| is a characteristic length that appropriately weighs translational and rotational degrees of freedom together.

ARCCTL

The arc length method can be controlled based on the displacement of a single node in the model. For example, in dome reversal problems the node at the center of the dome can be used. By default, the generalized arc length method is used, where the norm of the global displacement vector controls the solution. This includes all nodes.

ARCLEN

In many cases the arc length method has difficulty tracking the load displacement curve through critical regions. Using 0 < ARCLEN < 1 will reduce the step size to assist tracking the load-displacement curve with more accuracy. Use of ARCLEN < 1 will cause more steps to be taken. Suggested values are 1.0 (the default), 0.5, 0.25, and 0.10.

ARCDMP

Some static problems exhibit oscillatory response near instability points. This option numerically suppresses these oscillations, and may improve the convergence behavior of the post-buckling solution.

LMSTD

The default method for determining convergence of the nonlinear line search is to find the minimum of the energy. This

VARIABLE **REMARKS**

parameter allows choosing the energy on only the translational variables, energy of both the translational and rotational variables, or for minimizing the residual (forces). The effect of using a residual based line search is not always positive, sometimes it is too restrictive and stops convergence. However, it is a more conservative approach than using the energy based method since it explicitly controls the norm of the residual. It should not be seen as a better strategy than the energy method but as an alternative to try in cases when the default method seems to be working poorly. Line search methods 5 and 6 are conservative line search methods to be used for highly nonlinear problems, these should not be used as default but as final resorts to potentially resolve convergence issues. The rule of thumb is that the LSMTD = 5 is slow but robust and LSMTD = 6 is even slower but more robust.

LSDIR

In Version 971 of LS-DYNA new line search options were added. The traditional approach (LSDIR = 1) computes the line search direction using all variables. The new (default) approach of LSIDR = 2 computes the line search direction only on the unconstrained variables. It has proven to be both robust and more efficient. We have also included two new approaches to try for problems where the default and traditional approach fail and the user is using Full Newton (ILIMIT = 1). See the next two remarks for more information on those methods.

IRAD / SRAD

The parameters IRAD and SRAD are for the curved line search (LSDIR = 4). The first parameter is a switch (0 or 1) to invoke this line search, an intermediate value is interpreted as weighted combination of a straight and curved line search (the curvature radius is decreased with increasing IRAD). A value of unit is recommended in situations with rather smooth responses, e.g. springback and similar problems. Also, IRAD = 1 seems to work best with full Newton iterations. The SRAD parameter should be equal to 0 for most cases, this means that the search curve for a node is determined from the search direction of nodes connected to the same elements as that node. SRAD > 0 is interpreted as a radius of influence,

meaning that the search curve for a node is determined from the search direction of nodes within a distance SRAD of this node. This option was introduced as an experiment to see if this had a smoothing and stabilizing effect. A value of 0.0 is

VARIABLE	REMARKS
	currently recommended.
AWGT / SRED	The parameters AWGT and SRED are for the adaptive line search. The intention is to improve robustness for problems that have tendencies to oscillate or diverge, indicated by the dnorm and enorm parameter outputs in the iterations (stdout). A value of 0.5 is recommended for AWGT as a starting point. With a nonzero value the motions of individual nodes are tracked. For nodes that are oscillating (going back and forth in space), the maximum step size for the next iteration is reduced in proportion to the parameter AWGT, and for nodes that are not oscillating but going nicely along a straight path, the maximum step size for the next iteration is increased in proportion to 1-AWGT. In test problems, the introduction of the adaptive line search has stabilized the implicit procedure in the sense that the dnorm and enorm values are more monotonically decreasing until convergence with virtually no oscillations. If a problem is still oscillating or diverging, the user should try to increase the AWGT parameter since this is a more restrictive approach but probably gives a slower convergence rate. An option for nasty problems is also to use SRED > 0 which is the initial step reduction factor (less than 1). This means that the initial step size is reduced by this value but the maximum step size will increase by an amount that is determined by the success in the iterative procedure, eventually it will reach unity. It can never decrease. Also here, it is intended to be used with full Newton method.

CONTROL_IMPLICIT_SOLVER_*{OPTION}

Available options include:

<BLANK>

DYN

SPR

Purpose: These optional cards apply to implicit calculations. The linear equation solver performs the CPU-intensive stiffness matrix inversion (see also *CONTROL_IMPLICIT_GENERAL). The DYN option allows setting controls specifically for the dynamic relaxation phase. The SPR option allows setting controls specifically for the springback phase.

Card 1	1	2	3	4	5	6	7	8
Variable	LSOLVR	LPRINT	NEGEV	ORDER	DRCM	DRCPRM	AUTOSPC	AUTOTOL
Type	I	I	I	I	I	F	I	F
Default	4	0	2	0	4	see below	1	see below

Card 2 is optional.

Card 2	1	2	3	4	5	6	7	8
Variable	LCPACK	MTXDMP						
Type	I	I						
Default	2	0						

VARIABLE	DESCRIPTION
LSOLVR	Linear equation solver method (see Remarks below). EQ.4: SMP parallel multi-frontal sparse solver (default). EQ.5: SMP parallel multi-frontal sparse solver, double precision EQ.6: BCSLIB-EXT, direct, sparse, double precision EQ.10: iterative, best of currently available iterative methods EQ.11: iterative, Conjugate Gradient method EQ.12: iterative, CG with Jacobi preconditioner EQ.13: iterative, CG with Incomplete Choleski preconditioner EQ.14: iterative, Lanczos method EQ.15: iterative, Lanczos with Jacobi preconditioner EQ.16: iterative, Lanczos with Incomplete Choleski preconditioner
LPRINT	Linear solver print flag controls screen and message file output (see Remarks below). EQ.0: no printing EQ.1: output summary statistics on memory, cpu requirements EQ.2: more statistics EQ.3: even more statistics and debug checking NOTE: during execution, use the interactive command "<ctrl-c> lprint" to toggle this print flag between 0 and 1.
NEGEV	Negative eigenvalue flag. Selects procedure when negative eigenvalues are detected during stiffness matrix inversion (see Remarks below). EQ.1: stop, or retry step if auto step control is active EQ.2: print warning message, try to continue (default)
ORDER	Ordering option (see Remarks below) EQ.0: method set automatically by LS-DYNA EQ.1: MMD, Multiple Minimum Degree. EQ.2: Metis

VARIABLE	DESCRIPTION
DRCM	Drilling rotation constraint method for shells (see Remarks below). EQ.1: add drilling stiffness (old Version 970 method) EQ.2: same as 4 below EQ.3: add no drilling stiffness EQ.4: add drilling stiffness (improved method) (default)
DRCPRM	Drilling rotation constraint parameter for shells. This parameter scales the drilling stiffness. For the old method (DRCM = 1) the default value of DRCPRM is 1.0 for linear analysis, 100.0 for nonlinear implicit analysis,; and either 1.E-12 or 1.E-8 for eigenvalue analysis depending on the shell element type. For eigenvalue analysis, the input value for DRCPRM is ignored. For the improved method (default, DRCM = 4), the default value of DRCPRM is as described above for the old method except default DRCPRM is 1.0 for nonlinear implicit analysis.
AUTOSPC	Automatic Constraint Scan flag EQ.1: scan the assembled stiffness matrix looking for unconstrained, unattached degrees of freedom. Generate additional constraints as necessary to avoid negative eigenvalues. EQ.2: do not add constraints.
AUTOTOL	AUTOSPC tolerance. The test for singularity is the ratio of the smallest singular value and the largest singular value. If this ratio is less than AUTOTOL, then the triple of columns are declared singular and a constraint is generated. Default value in single precision is 10^{-4} and in double precision, 10^{-8}.
LCPACK	Matrix assembly package. EQ.2: Use v970's LCPACK (default, only available option in 971) EQ.3: Same as 2, but incorporates a non-symmetric linear solver, see remark for LCPACK.
MTXDMP	Matrix and right-hand-side dumping. LS-DYNA has the option of dumping the globally assembled stiffness matrix and right-hand-side vectors files in Harwell-Boeing sparse matrix format. Such output may be useful for comparing to other linear equation solution packages.

VARIABLE **DESCRIPTION**

EQ.0: No dumping

GT.0: Dump all matrices and right-hand-side vectors every MTXDMP time steps. Output is written as ASCII text and the iinvolved filenames are of the following form:

K_*xxxx*_*yyy*.mtx.rb

This file contains the stiffness matrix at step *xxxx*, iteration *yyy*.

M_xxxx_yyy.mtx.rb

This file contains the mass matrix at step *xxxx*, iteration. Only for eigenvalue analysis._

MW_xxxx_yyy.mtx.rb

This file contains the damping matrix at step *xxxx*, iteration. Only for simulations with damping.

K_*xxxx*_*yyy*_*zzz*.rhs.rb

This file contains the right hand side at step *xxxx*, iteration *yyy*, where *yyy* is the iteration at which a stiffness matrix is formed; *zzz* is the cumulative iteration number for the step. The values of *yyy* and *zzz* don't always coincide because the stiffness matrix is not necessarily reformed every iteration.

Node_Data_*xxxx*_*yyy*

This file maps stiffness matrix to nodes and provides nodal coordinates.

LT.0: Like positive values of MTXDMP but dumped data is binary.

EQ.|9999|: Simulation is terminated after dumping matrices and right hand side prior to factorization.

Remarks:

VARIABLE **REMARKS**

LSOLVR The linear solver is used to compute the inverse of the global stiffness matrix, which is a costly procedure both in memory and cpu time. Direct solvers apply Gaussian elimination, while iterative solvers successively improve "guesses" at the correct solution. Iterative solvers require far less memory than direct

VARIABLE **REMARKS**

solvers, but may suffer from convergence problems. Generally, iterative solvers are poor for automotive applications, but can be superior for large brick element soil models in civil engineering.

Solvers 5 and 6 promote the global matrix to double precision before factoring to reduce numerical truncation error. Solvers 4 and 5 are equivalent if a double precision executable is used.

Solver 6 is the direct linear equation solver from BCSLIB-EXT, Boeing's Extreme Mathematical Library. This option should be used whenever the factorization is too large to fit into memory. It has extensive capabilities for out-of-core solution and can solve larger problems than any of the other direct factorization methods. Solver 6 also includes a sophisticated pivoting strategy which can be superior for nearly singular matrices.

Solver 5 is the only option supported in MPP.

LPRINT

Select printing of the timing and storage information (LPRINT = 1) if you are comparing performance of linear equation solvers, or if you are running out of memory for large models. Minimum memory requirements for in-core and out-of-core solution are printed. This flag can also be toggled using sense switch "<ctrl-c> lprint". *For best performance, increase available memory using “memory=“ on the command line until an IN-CORE solution is indicated.*

When using solver option 6, LPRINT = 2 and 3 will cause increased printed output of statistics and performance information.

NEGEV

Negative eigenvalues result from underconstrained models (rigid body modes), severely deformed elements, or non-physical material properties. This flag allows control to be passed directly to the automatic time step controller when negative eigenvalues are detected. Otherwise, significant numerical roundoff error is likely to occur during factorization, and equilibrium iterations may fail (see *CONTROL_IMPLICIT_AUTO).

VARIABLE **REMARKS**

ORDER

The system of linear equations is reordered to optimize the sparsity of the factorization when using direct methods. Metis is a ordering method from University of Minnesota which is very effective for larger problems and for 3D solid problems, but also very expensive. MMD is inexpensive, but may not produce an optimum reordering, leading to higher cost during numeric factorization. MMD is usually best for smaller problems (less than 100,000 degrees of freedom).

Reordering cost is included in the symbolic factorization phase of the linear solver (LSPRINT ≥ 1). For large models, if this cost exceeds 20% of the numeric factorization cost, it may be more efficient to select the MMD method.

Note that the values of LPRINT and ORDER also affect the eigensolution software. That is LPRINT and ORDER from this keyword card is applicable to eigensolution.

DRCPRM

To avoid a singular stiffness matrix in implicit analysis of flat shell topologies, some constraint on the drilling degree of freedom is needed. The default method of applying this constraint, DRCPRM = 4, adds the consistent force vector for consistency and improved convergence as compared to the old method, DRCPRM = 1.

In explicit analysis, an unconstrained drilling degree of freedom is usually not a concern since a stiffness matrix is not used. However, special situations may arise in which the user wishes to include additional resisting rotational force in the drilling degree of freedom for improved robustness and/or accuracy. To activate the consistent drilling constraint in explicit analysis, use the input variables DRCPSID and DRCPRM for *CONTROL_SHELL.

LCPACK

Certain features may break the symmetry of the stiffness matrix. Unless LCPACK is set to 3 these contributions are suppressed or symmetrized by the default symmetric linear solver. However, when LCPACK is set to 3 a more general linear solver lifting the symmetry requirement is used. The solver for non-symmetric matrices is more computationally expensive.

Keywords for which the non-symmetric contribution is implemented are listed below:

***CONTACT_..._MORTAR:**

VARIABLE	REMARKS

The mortar contact accounts for frictional non-symmetry in the resulting tangent stiffness matrix, the effects on convergence characteristics have not yet shown to be significant.

***LOAD_SEGMENT_NONUNIFORM:**

The non-symmetric contribution may be significant for the follower load option, LCID < 0.

***LOAD_SEGMENT_SET_NONUNIFORM:**

The non-symmetric contribution may be significant for the follower load option, LCID < 0.

***MAT_FABRIC_MAP:**

This stress map fabric model accounts for non-symmetry in the material tangent modulus, representing the non-linear Poisson effect due to complex interaction of yarns.

***SECTION_SHELL, *SECTION_SOLID:**

User defined resultant elements (ELFORM = 101, 102, 103, 104, 105 with NIP=0) support the assembly and solution of non-symmetric element matrices.

***SECTION_BEAM:**

Belytschko-Schwer beam (ELFORM=2) nonsymmetric geometric stiffness contribution is supported.

CONTROL_IMPLICIT_STABILIZATION_*{OPTION}

Available options include:

<BLANK>

DYN

SPR

Purpose: This optional card applies to implicit calculations. Artificial stabilization is required for multi-step unloading in implicit springback analysis (see also *CONTROL_IMPLICIT_GENERAL). The DYN option allows setting controls specifically for the dynamic relaxation phase. The SPR option allows setting controls specifically for the springback phase.

Card 1	1	2	3	4	5	6	7	8
Variable	IAS	SCALE	TSTART	TEND				
Type	I	F	F	F				
Default	2	1.0	see below	see below				

VARIABLE	DESCRIPTION
IAS	Artificial Stabilization flag EQ.1: active EQ.2: inactive (default)
SCALE	Scale factor for artificial stabilization. For flexible parts with large springback, like outer body panels, a value of 0.001 may be required. EQ.-n: curve ID = n gives SCALE as a function of time
TSTART	Start time. (Default: immediately upon entering implicit mode)
TEND	End time. (Default: termination time)

Remarks:

Artificial stabilization allows springback to occur over several steps. This is often necessary to obtain convergence during equilibrium iterations on problems with large springback deformation. Stabilization is introduced at the start time TSTART, and slowly removed as the end time TEND is approached. Intermediate results are not accurate representations of the fully unloaded state. The end time TEND must be reached exactly for total springback to be predicted accurately.

VARIABLE	REMARKS
IAS	The default for IAS depends on the analysis type in *CONTROL_-IMPLICIT_GENERAL. For “seamless” springback analysis, automatic time step control and artificial stabilization are activated by default. Otherwise, IAS is inactive by default.
SCALE	This is a penalty scale factor similar to that used in contact interfaces. If modified, it should be changed in order-of-magnitude increments at first. Large values suppress springback deformation until very near the termination time, making convergence during the first few steps easy. Small values may not stabilize the solution enough to allow equilibrium iterations to converge.

CONTROL_IMPLICIT_STATIC_CONDENSATION_*{OPTION}

Available options include:

<BLANK>

BINARY

Purpose: Request static condensation of a part to build a reduced linearized model for later computation with *ELEMENT_DIRECT_MATRIX_INPUT. Optionally the analysis can continue using the linearization for the current analysis.

Card 1	1	2	3	4	5	6	7	8
Variable	SC_FLAG	SC_NSID	SC_PSID	SE_MASS	SE_STIFF	SE_INERT		
Type	I	I	I	C	C	C		
Default	0	0	0					

Card 2	1	2	3	4	5	6	7	8
Variable	SE_FILENAME							
Type	A80							

VARIABLE	DESCRIPTION
SC_FLAG	Static Condensation Control Flag EQ.0: no static condensation will be performed EQ.1: create superelement representation based on static condensation. EQ.2: use static condensation to build a linearized representation for a part and use that linearized representation in the following analysis.
SC_NSID	Node set ID for nodes to be preserved in the static condensation procedure. Required when SC_FLAG = 1.

VARIABLE	DESCRIPTION
SC_PSID	Part set ID for parts to be included in the static condensation procedure. When SC_FLAG = 1, SC_PSID can be used to specify a subset of the model with the default being the entire model. When SC_FLAG = 2, SC_PSID is required. SC_PSID = 0 implies that the entire model is condensed.
SE_MASS	Name of the superelement mass matrix. If left blank it is not generated.
SE_STIFF	Name of the superelement stiffness matrix. If left blank it is not generated.
SE_INERT	Name of the superelement inertia matrix, required for gravity loading applications of the superelement. If left blank it is not generated.
SE_FILENAME	If any of SE_MASS, SE_STIFF, or SE_INERT is blank then the second line is required and contains the file name for the superelement.

Remarks:

To use this feature, an implicit analysis must be requested using IMFLAG = 1 on *CONTROL_IMPLICIT_GENERAL, and a non-zero termination time must be specified on *CONTROL_TERMINATION. A double precision version of LS-DYNA should be used for best accuracy. The superelement model is written to file SE_FILENAME.

Static condenstation is the reduction of the global stiffness and mass matrices to a specified sets of rows and columns associated with the nodes in the node set SC_NSID. The first and second node set attribute parameters can be optionally used to restrict the translational and rotational degrees of freedom for which modes are requested, respectively, according to the following syntax:

Node set attribute parameters DA1 and A1: translational degree of freedom codes

Node set attribute parameters DA2 and A2: rotational degree of freedom codes

Code	Modes Computed
0	(See note below.)
1	X degree of freedom only
2	Y degree of freedom only
3	Z degree of freedom only
4	X, Y degrees of freedom only
5	Y, Z degrees of freedom only
6	X, Z degrees of freedom only
7	X, Y, Z degrees of freedom

Setting both node set attributes to zero is equivalent to setting both node set attributes to 7 (X, Y, and Z for translational and rotational degrees of freedom).

If one node set attribute is nonzero (codes 1 to 7) and the other node set attribute is zero, then the zero attribute means NO degrees of freedom are considered. For example, if DA1 = 2 and DA2 = 0, then only the Y-translational degree of freedom modes are calculated.

The user can create the superelement representation of the reduced model by specifying the SE_MASS, SE_STIFF, SE_INERT and SE_FILENAME fields. This implementation does not include SE_DAMP. The file, by default is written in the Nastran DMIG file format and can be used as input to *ELEMENT_DIRECT_MATRIX_INPUT. The keyword option BINARY can be used to create a binary representation for the superelement which can be used with *ELEMENT_DIRECT_MATRIX_INPUT_BINARY to reduce the file size.

Static Condensation is equivalent to using only constraint modes with *CONTROL_IMPLICIT_MODES. Static Condensation does have the ability to continue the analysis using the linear representation for a part set.

*CONTROL_IMPLICIT_TERMINATION

Purpose: Specify termination criteria for implicit transient simulations.

Card 1	1	2	3	4	5	6	7	8
Variable	DELTAU	DELTA1	KETOL	IETOL	TETOL	NSTEP		
Type	F	F	F	F	F	I		
Default	0.0	0.0	0.0	0.0	0.0	3		

VARIABLE	DESCRIPTION
DELTAU	Terminate based on relative total displacement in the Euclidean norm. GT.0.0: terminate when displacement in the Euclidean norm for last time step relative to the total displacement in the Euclidean norm is less than DELTAU.
DELTA1	Terminate based on relative total displacement in the max norm. GT.0.0: terminate when displacement in the max norm for last time step relative to the total displacement in the max norm is less than DELTAU.
KETOL	Terminate based on kinetic energy GT.0.0: terminate when kinetic energy drops below KETOL for NSTEP consecutive implicit time steps.
IETOL	Terminate based on internal energy GT.0.0: terminate when internal energy drops below IETOL for NSTEP consecutive implicit time steps.
TETOL	Terminate based on total energy GT.0.0: terminate when total energy drops below TETOL for NSTEP consecutive implicit time steps.
NSTEP	Number of steps used in the early termination tests for kinetic, internal, and total energy.

Remarks:

For some implicit applications it is useful to terminate when there is no change in displacement or low energy. This keyword provides the ability to specify such a stopping criterias to terminate the simulation prior to ENDTIM.

*CONTROL_MAT

Purpose: Define global control parameters for material model related properties.

Card 1	1	2	3	4	5	6	7	8
Variable	MAEF							
Type	I							
Default	0							

VARIABLE **DESCRIPTION**

MAEF Failure options:

EQ.0: all *MAT_ADD_EROSION definitions are active.

EQ.1: switch off all *MAT_ADD_EROSION definitions globally. This feature is useful for larger models where removing the *MAT_ADD_EROSION cards is inconvient.

*CONTROL_MPP

Purpose: Set control parameters for MPP specific features.

*CONTROL_MPP_CONTACT_GROUPABLE

*CONTROL_MPP_DECOMPOSITION_ARRANGE_PARTS

*CONTROL_MPP_DECOMPOSITION_AUTOMATIC

*CONTROL_MPP_DECOMPOSITION_BAGREF

*CONTROL_MPP_DECOMPOSITION_CHECK_SPEED

*CONTROL_MPP_DECOMPOSITION_CONTACT_DISTRIBUTE

*CONTROL_MPP_DECOMPOSITION_CONTACT_ISOLATE

*CONTROL_MPP_DECOMPOSITION_DISABLE_UNREF_CURVES

*CONTROL_MPP_DECOMPOSITION_DISTRIBUTE_ALE_ELEMENTS

*CONTROL_MPP_DECOMPOSITION_DISTRIBUTE_SALE_ELEMENTS

*CONTROL_MPP_DECOMPOSITION_DISTRIBUTE_SPH_ELEMENTS

*CONTROL_MPP_DECOMPOSITION_ELCOST

*CONTROL_MPP_DECOMPOSITION_FILE

*CONTROL_MPP_DECOMPOSITION_METHOD

*CONTROL_MPP_DECOMPOSITION_NUMPROC

*CONTROL_MPP_DECOMPOSITION_OUTDECOMP

*CONTROL_MPP_DECOMPOSITION_PARTS_DISTRIBUTE

*CONTROL_MPP_DECOMPOSITION_PARTSET_DISTRIBUTE

*CONTROL_MPP_DECOMPOSITION_RCBLOG

*CONTROL_MPP_DECOMPOSITION_SCALE_CONTACT_COST

*CONTROL_MPP_DECOMPOSITION_SCALE_FACTOR_SPH

*CONTROL_MPP_DECOMPOSITION_SHOW

*CONTROL_MPP_DECOMPOSITION_TRANSFORMATION

*CONTROL_MPP_IO_LSTC_REDUCE

*CONTROL_MPP_IO_NOBEAMOUT

*CONTROL_MPP_IO_NOD3DUMP

*CONTROL_MPP_IO_NODUMP

*CONTROL_MPP_IO_NOFAIL

*CONTROL_MPP_IO_NOFULL

*CONTROL_MPP_IO_SWAPBYTES

*CONTROL_MPP_MATERIAL_MODEL_DRIVER

*CONTROL_MPP_PFILE

*CONTROL_MPP_CONTACT_GROUPABLE

Purpose: Allow for global specification that the GROUPABLE algorithm should be enabled/disabled for contacts when running MPP.

Card 1	1	2	3	4	5	6	7	8
Variable	GRP							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
GRP	The sum of these available options (in any combination that makes sense): 1: Turn on GROUPABLE for all non-tied contacts 2: Turn on GROUPABLE for all tied contacts 4: Turn off GROUPABLE for all non-tied contacts 8: Turn off GROUPABLE for all tied contacts

Remarks:

The GROUPABLE algorithm is an alternate MPP communication algorithm for SINGLE_SURFACE, NODE_TO_SURFACE, and SURFACE_TO_SURFACE contacts. This algorithm does not support all contact options, including SOFT = 2, as of yet, and is still under development. It can be significantly faster and scale better than the normal algorithm when there are more than two or three applicable contact types defined in the model. Its intent is to speed up the contact processing but not to change the behavior of the contact.

This keyword will override any setting of the GRPABLE parameter on the *CONTACT_…_MPP card, and is intended as a way to quickly experiment with this feature. The equivalent pfile option is "contact { groupable GRP }" where GRP is an integer as described above.

CONTROL_MPP_DECOMPOSITION_ARRANGE_PARTS_*OPTION

Purpose: Allow users to distribute certain part(s) to all processors or to isolate certain part(s) in a single processor. This keyword supports multiple entries. Each entry is be processed as a separate region for decomposition.

When this keyword is part of an included file and the **LOCAL** option is given, the decomposition will be done in the coordinate system of the included file, which may be different from the global system, if the file is included using the ***INCLUDE_TRANSFORM** keyword.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	TYPE	NPROC	FRSTP				
Type	I	I	I	I				
Default	none	none	None	None				

VARIABLE	DESCRIPTION
ID	Part ID/Part set ID
TYPE	EQ.0: Part ID to be distributed to all processors EQ.1: Part Set ID to be distributed to all processors EQ.10: Part ID to be lumped into one processor EQ.11: Part Set ID to be lumped into one processor.
NPROC	Used only for TYPE equal to 0 or 1. Evenly distributed Part ID/Part set ID to NPROC of processors.
FRSTP	Used only for TYPE equal to 0 or 1. Starting MPP rank ID.

Remarks:

There is no equivalent option under pfile.

*CONTROL_MPP_DECOMPOSITION_AUTOMATIC

Purpose: Instructs the program to apply a simple heuristic to try to determine the proper decomposition for the simulation.

There are no input parameters. The existence of this keyword triggers the automated decomposition. This option should not be used if there is more than one occurrence of any of the following options in the model:

*INITIAL_VELOCITY

*CHANGE_VELOCITY

*BOUNDARY_PRESCRIBED_MOTION

And the following control card must not be used:

*CONTROL_MPP_DECOMPOSITION_TRANSFORMATION

For the general case, it is recommended that you specify the proper decomposition using the command *CONTROL_MPP_DECOMPOSITION_TRANSFORMATION instead.

*CONTROL_MPP_DECOMPOSITION_BAGREF

Purpose: With this card LS-DYNA performs decomposition according to the airbag's reference geometry, rather than the folded geometry.

Other than BAGID values this card takes no input parameters. The initial geometry may lead to a poor decomposition once the bag is deployed. This option will improve load balancing for the fully deployed geometry.

Optional card(s) for selected reference geometry ID

Card 1	1	2	3	4	5	6	7	8
Variable	BAGID1	BAGID2	BAGID3	BAGID4	BAGID5	BAGID6	BAGID7	BAGID8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
BAGID*i*	ID defined in *AIRBAG_REFERENCE_GEOMETRY_ID or *AIRBAG_SHELL_REFERENCE_GEOMETRY_ID

Bags specified in the optional cards will be decomposed based on the reference geometry. If there is no card given, *all* bags will be decomposed by their reference geometry.

Remarks:

Command in partition file (pfile): BAGREF. The option for selecting particular airbags is only available when using keyword input.

*CONTROL_MPP_DECOMPOSITION_CHECK_SPEED

Purpose: Modifies the decomposition depending on the relative speed of the processors involved.

There are no input parameters. Use of this keyword activates a short floating point timing routine to be executed on each processor. The information gathered is used during the decomposition, with faster processors being given a relatively larger portion of the problem. This option is not recommended on homogeneous systems.

CONTROL_MPP_DECOMPOSITION_CONTACT_DISTRIBUTE_*OPTION

Purpose: Ensures that the indicated contact interfaces are distributed across all processors, which can lead to better load balance for large contact interfaces. If this appears in an included file and the **LOCAL** option is given, the decomposition will be done in the coordinate system of the included file, which may be different from the global system if the file is included via ***INCLUDE_TRANSFORM**.

Card 1	1	2	3	4	5	6	7	8
Variable	ID1	ID2	ID3	ID4	ID5			
Type	I	I	I	I	I			
Default	none	none	none	none	none			

VARIABLE	DESCRIPTION
ID1	First contact interface ID to distribute. If no contact ID's are specified, the number given here corresponds to the order of the interfaces as they appear in the input, with the first being 1.
ID2, ID3, ID4, ID5	Remaining interfaces ID's to distribute.

Remarks:

Up to 5 contact interface ID's can be specified. The decomposition is modified as follows: First, all the elements involved in the first contact interface are decomposed across all the processors. Then all the elements involved in the second contact interface (excluding any already assigned to processors) are distributed, and so on. After all the contact interfaces given are processed, the rest of the input is decomposed in the normal manner. This will result in each processor having possibly several disjoint portions of the input assigned to it, which will increase communications somewhat. However, this can be offset by improved load balance in the contact. It is generally recommended that at most one or two interfaces be specified, and then only if they are of substantial size relative to the whole problem.

*CONTROL_MPP_DECOMPOSITION_CONTACT_ISOLATE

Purpose: Ensures that the indicated contact interfaces are isolated on a single processor, which can lead to decreased communication.

Card 1	1	2	3	4	5	6	7	8
Variable	ID1	ID2	ID3	ID4	ID5			
Type	I	I	I	I	I			
Default	none	none	none	none	none			

VARIABLE	DESCRIPTION
ID1	First contact interface ID to isolate. If no contact ID's are specified, the number given here corresponds to the order of the interfaces as they appear in the input, with the first being 1.
ID2, ID3, ID4, ID5	Remaining interfaces ID's to isolate.

Remarks:

Up to 5 contact interfaces can be specified. The decomposition is modified as follows: First, all the elements involved in the first contact interface ID are assigned to the first processor. Then all the elements involved in the second contact interface ID (excluding any already assigned to processors) are assigned to the next processor, and so on. After all the contact interfaces given are processed, the rest of the input is decomposed in the normal manner. This will result in each of the interfaces being processed on a single processor. For small contact interfaces this can result in better parallelism and decreased communication.

*CONTROL_MPP_DECOMPOSITION_DISABLE_UNREF_CURVES

Purpose: Disable unreferenced time dependent load curves for the following keyword.

*BOUNDARY_PRESCRIBED_MOTION_NODE

*LOAD_NODE

*LOAD_SHELL_ELEMENT

*LOAD_THERMAL_VARIABLE_NODE

The details of this operation are reported in each processor's scratch "scr####" file. This will skip the curve evaluation on each cycle, and improve the parallel efficiency.

Remarks:

Command in partition file (pfile): DUNREFLC.

*CONTROL_MPP_DECOMPOSITION_DISTRIBUTE_ALE_ELEMENTS

Purpose: Ensures ALE elements are evenly distributed to all processors

There are no input parameters and the card below is optional. ALE elements usually have higher computational cost than other type of elements and it is better to distribute them to all CPU for better load balance. The existence of this keyword causes DYNA/MPP to extract ALE parts from input and then evenly distributed to all processors.

The card is optional

Card 1	1	2	3	4	5	6	7	8
Variable	OVERLAP							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
OVERLAP	For FSI models where structures are inside ALE meshes (see *CONSTRAINED_LAGRANGE_IN_SOLID), decompose the structure and ALE domains together instead of first the structure and then ALE (See Remark 2). Set type: EQ.0: Off EQ.1: On

Remarks:

1. Command in partition file (pfile): ALEDIST.

2. Most of the processors will have to deal with MPP subdomains from the structure and ALE meshes: a portion of the ALE computational domain and a portion of the structure meshes. The default decomposition (first divide the structures, then ALE) does not always overlap these subdomains. The more they overlap, the lesser the MPP communications in the coupling cost. Cutting the ALE and structure meshes together allows their MPP subdomains to be as inclusive as possible.

*CONTROL_MPP_DECOMPOSITION_DISTRIBUTE_SPH_ELEMENTS

Purpose: Ensures SPH elements are evenly distributed to all processors

There are no input parameters. SPH elements usually have higher computational cost than other type of elements and it is better to distribute them to all CPU for better load balance. The existence of this keyword causes DYNA/MPP to extract SPH parts from input and then evenly distributed to all processors.

Remarks:

Command in partition file (pfile): SPHDIST.

*CONTROL_MPP_DECOMPOSITION_ELCOST

Purpose: Instructs the program to use a hardware specific element cost weighting for the decomposition

Card 1	1	2	3	4	5	6	7	8
Variable	ITYPE							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
ITYPE	Hardware specific cost profile. EQ.1: Fujitsu PrimePower EQ.2: Intel IA 64, AMD Opteron EQ.3: Intel Xeon 64 EQ.4: General profile

Remarks:

Command in partition file (pfile): elcost itype.

*CONTROL_MPP_DECOMPOSITION_FILE

Purpose: Allow for pre-decomposition and a subsequent run or runs without having to do the decomposition.

Card 1	1	2	3	4	5	6	7	8
Variable	NAME							
Type	A80							
Default	none							

VARIABLE	DESCRIPTION
NAME	Name of a file containing (or to contain) a decomposition record.

Remarks:

If the indicated file does not exist, it is created with a copy of the decomposition information from this run. If the file exists, it is read and the decomposition steps can be skipped. The original run that created the file must be for a number of processors that is a multiple of the number of processors currently being used. Thus, a problem can be decomposed once for, say, 48 processors. Subsequent runs are then possible on any number that divides 48: 1, 2, 3, 4, 6, etc. Since the decomposition phase generally requires more memory than execution, this allows large models to be decomposed on one system and run on another (provided the systems have compatible binary formats). The file extension ".pre" is added automatically.

*CONTROL_MPP_DECOMPOSITION_METHOD

Purpose: Specify the decomposition method to use.

Card 1	1	2	3	4	5	6	7	8
Variable	NAME							
Type	A80							
Default	RCB							

VARIABLE	DESCRIPTION
NAME	Name of the decomposition method to use. There are currently two options: EQ.“RCB”: recursive coordinate bisection EQ.“GREEDY”: a simple heuristic method In almost all cases the RCB method is superior and should be used.

*CONTROL_MPP_DECOMPOSITION_NUMPROC

Purpose: Specify the number of processors for decomposition.

Card 1	1	2	3	4	5	6	7	8
Variable	N							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
N	Number of processors for decomposition.

Remarks:

This is used in conjunction with the CONTROL_MPP_DECOMPOSITION_FILE command to allow for later runs on different numbers of processors. By default, the decomposition is performed for the number of processors currently being used. However, a different value can be specified here. If N > 1 and only one processor is currently being used, the decomposition is done and then the program terminates. If N is *not* a multiple of the current number of processors, then it is ignored the execution proceeds with the current number of processors. Otherwise, the decomposition is performed for N processors, and the execution continues using the current number of processors.

*CONTROL_MPP_DECOMPOSITION_OUTDECOMP

Purpose: Instructs the program to output element's ownership data to file for post-processor to show state data from different processors

Card 1	1	2	3	4	5	6	7	8
Variable	TYPE							
Type	I							
Default	none							

VARIABLE **DESCRIPTION**

ITYPE Sets the format for the output file.

EQ.1: database in LS-PrePost format:

decomp_parts.lsprepost.

EQ.2: database in animator format:

decomp_parts.ses

Remarks:

Command in partition file (pfile): OUTDECOMP ITYPE.

When ITYPE is set to 1, the elements assigned to any particular core can be viewed and animated by LS-PrePost by (1) reading the d3plot data, and then (2) selecting Models > Views > MPP > Load > decomp_parts.lsprepost.

CONTROL_MPP_DECOMPOSITION_PARTS_DISTRIBUTE_*OPTION

Purpose: Distribute the parts given in this option to all processors before the decomposition for the rest of the model is performed. If this appears in an included file and the **LOCAL** option is given, the decomposition will be done in the coordinate system of the included file, which may be different from the global system if the file is included via ***INCLUDE_TRANSFORM**.

Card 1	1	2	3	4	5	6	7	8
Variable	ID1	ID2	ID3	ID4	ID5	ID6	ID7	ID8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
ID1, ID2, ID3, …	For each ID: GT.0: ID is a part number. LT.0: –ID is a part set number. All parts defined in this card will be treated as a single region to be decomposed.

Remarks:

Up to 16 parts/part sets can be specified. The decomposition is modified as follows: the elements involved in the given parts are put into a separate domain from rest of the model and then distributed to all processors to balance their computational cost. Then the remainder of the model will be distributed in the usual way.

This is equivalent to the pfile command (for example, if ID1-ID3 are part ids and ID4-ID6 are partset ids):

```
decomp { region { parts ID1 ID2 ID3 or partsets ID4 ID5 ID6 } }
```

(the partset ids are positive when used in the pfile).

CONTROL_MPP_DECOMPOSITION_PARTSET_DISTRIBUTE_*OPTION

Purpose: Distribute the part sets given in this option to all processors before the decomposition for the remainder of the model is performed. If this appears in an included file and the **LOCAL** option is given, the decomposition will be done in the coordinate system of the included file, which may be different from the global system if the file is included via ***INCLUDE_TRANSFORM**.

Card 1	1	2	3	4	5	6	7	8
Variable	ID1	ID2	ID3	ID4	ID5	ID6	ID7	ID8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
ID1, ID2, ID3, …	Partset ID to be distributed.. All parts in ID1 will be shared across all processors. Then all parts in ID2 will be distributed, and so on..

Remarks:

Any number of part sets can be specified. Each part set is distributed across all processors, in the order given. The order may be significant, in particular, if a part ID is in more than one set. Distribution of these parts is done before any decomposition specifications given in the pfile.

*CONTROL_MPP_DECOMPOSITION_RCBLOG

Purpose: Causes the program to record decomposition information in the indicated file, for use in subsequent analyses.

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	A80							
Default	none							

VARIABLE	DESCRIPTION
FILENAME	Name of output file where decomposition history will be recorded. This file can be used as the pfile for later analyses.

Remarks:

Command in parallel option file (pfile): rcblog filename.

*CONTROL_MPP_DECOMPOSITION_SCALE_CONTACT_COST

Purpose: Instructs the program to apply a scale factor to the list of contacts to change the partition weight for the decomposition.

Card 1	1	2	3	4	5	6	7	8
Variable	SF	ID1	ID2	ID3	ID4	ID5	ID6	ID7
Type	F	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
SF	Scale factor for the contact segments listed in the interface ID.
ID1, ID2, …	interfaces ID's to be considered for scaling. Include second card if necessary.

Remarks:

Up to 15 contact interfaces ID can be specified. The decomposition is modified by applying this scale factor to the default computational cost of elements for the given contact interface ID.

Command in partition file (pfile): CTCOST ID1, ID2, …, SF.

*CONTROL_MPP_DECOMPOSITION_SCALE_FACTOR_SPH

Purpose: Instructs the program to apply a scale factor to SPH elements to change the partition weight for the decomposition.

Card 1	1	2	3	4	5	6	7	8
Variable	SF							
Type	F							
Default	none							

VARIABLE	DESCRIPTION
SF	Scale factor

Remarks:

Command in partition file (pfile): SPHSF SF.

*CONTROL_MPP_DECOMPOSITION_SHOW

Purpose: The keyword writes the final decomposition to the d3plot database. There are no input parameters.

This keyword causes MPP LS-DYNA to terminate immediately after the decomposition phase without performing an analysis. The resulting d3plot database is designed to allow visualization of the decomposition by making each part correspond to the group of solids, shells, beams, thick shells, or SPH particles assigned to a particular processor. For example, in a model that includes various element types including solids, part 1 corresponds to the solid elements assigned to processor 1, part 2 corresponds to the solid elements assigned to processor 2, and so on.

This command can be used in conjunction with the *CONTROL_MPP_DECOMPOSITION_NUMPROC command to run on one processor and produce a d3plot file to visualize the resulting decomposition for the number of processors specified in *CONTROL_MPP_DECOMPOSITION_NUMPROC.

*CONTROL_MPP_DECOMPOSITION_TRANSFORMATION

Purpose: Specifies transformations to apply to modify the decomposition.

There are 10 different kinds of decomposition transformations available. For a detailed description of each, see Appendix O the LS-DYNA MPP user guide.

The data cards for this keyword consist of transformation operations. Each operation, depending on its type, involves either one or two additional cards. The input deck may include an arbitrary number of transformations with the next keyword, "*," card terminating this input.

Transformation Card 1. For each transformation this card is required.

Card 1	1	2	3	4	5	6	7	8
Variable	TYPE	V1	V2	V3	V4	V5	V6	
Type	A10	F	F	F	F	F	F	
Default	none	0.0	0.0	0.0	0.0	0.0	0.0	

Transformation Card 2. Additional card for TYPE set to one of VEC3, C2R, S2R, MAT.

Card 2	1	2	3	4	5	6	7	8
Variable	V7	V8	V9					
Type	F	F	F					
Default	0.0	0.0	0.0					

VARIABLE	DESCRIPTION
TYPE	Which transformation to apply. The allowed values are RX, RY, RZ, SX, SY, SZ, VEC3, C2R, S2R, and MAT.

VARIABLE	DESCRIPTION
V1 - V9	For type set to either RX, RY, RZ, SX, SY, or SZ: The parameter V1 gives either the angle of rotation (RX, RY, RZ) or the magnitude for the scaling (SX, SY, SZ). The remaining parameters are ignored. For type set to either VEC3, C2R, S2R, or MAT: All parameters are used. See the appendix for the "pfile."

*CONTROL_MPP_IO_LSTC_REDUCE

Purpose: Use LSTC's own reduce routine to get consistent summation of floating point data among processors. There are no input parameters.

Remarks:

Command in partition file (pfile): lstc_reduce.

*CONTROL_MPP_IO_NOBEAMOUT

Purpose: Suppress beam, shell, and solid element failure messages in the d3hsp and message files. There are no parameters for this keyword.

Remarks:

Command in parallel option file (pfile): nobeamout.

*CONTROL_MPP_IO_NOD3DUMP

Purpose: Suppresses the output of all dump files.

There are no input parameters. The existence of this keyword causes the d3dump and runrsf file output routines to be skipped.

*CONTROL_MPP_IO_NODUMP

Purpose: Suppresses the output of all dump files and full deck restart files.

There are no input parameters. The existence of this keyword causes the d3dump and runrsf file output routines to be skipped. It also suppresses output of the full deck restart file d3full.

*CONTROL_MPP_IO_NOFAIL

Purpose: Turn off failed element checking in MPP contact. If you know that no elements will fail, or that any such failure will not impact any of the contact calculations, turning on this option can increase the efficiency of the contact routines.

There are no input parameters.

*CONTROL_MPP_IO_NOFULL

Purpose: Suppresses the output of the full deck restart files.

There are no input parameters. The existence of this keyword suppresses the output of the full deck restart file d3full.

*CONTROL_MPP_IO_SWAPBYTES

Purpose: Swap bytes on some of the output files.

There are no input parameters. The existence of this keyword causes the d3plot file and the “interface component analysis” file to be output with bytes swapped. This is to allow further processing of data on a different machine that has big endian vs. little endian incompatibilities compared to the system on which the analysis is running.

*CONTROL_MPP_MATERIAL_MODEL_DRIVER

Purpose: Enable this feature in MPP mode. To allow MPP reader to pass the input phase even without any nodes and elements but using only one processor.

*CONTROL_MPP_PFILE

Purpose: Provide keyword support for the MPP “p=” pfile options

All lines of input up to the next keyword card will be copied to a temporary file which is effectively pre-pended to the “p=” file given on the command line (even if no such file is given). This allows all options available via the “p=” file to be specified in the keyword input. The only restriction is that pfile directives in the “directory” section are not available, as those must be processed before the keyword input file is read. See the “LS-DYNA MPP User Guide” in the appendix for details of the available pfile commands and their syntax.

*CONTROL_NONLOCAL

Purpose: Allocate additional memory for *MAT_NONLOCAL option.

Card 1	1	2	3	4	5	6	7	8
Variable	MEM							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
MEM	Percentage increase of memory allocated for *MAT_NONLOCAL option over that required initially. This is for additional storage that may be required due to geometry changes as the calculation proceeds. Generally, a value of 10 should be sufficient.

*CONTROL_OUTPUT

Purpose: Set miscellaneous output parameters. This keyword does not control the information, such as the stress and strain tensors, which is written into the binary databases. For the latter, see the keyword *DATABASE_EXTENT_BINARY.

Card 1	1	2	3	4	5	6	7	8
Variable	NPOPT	NEECHO	NREFUP	IACCOP	OPIFS	IPNINT	IKEDIT	IFLUSH
Type	I	I	I	I	F	I	I	I
Default	0	0	0	0	0.	0	100	5000

Remaining cards are optional.

Optional Card 2

Card 2	1	2	3	4	5	6	7	8
Variable	IPRTF	IERODE	TET10S8	MSGMAX	IPCURV	GMDT	IP1DBLT	EOCS
Type	I	I	I	I	I	F	I	I
Default	0	0	2	50	0	0.	0	0

Optional Card 3

Card 3	1	2	3	4	5	6	7	8
Variable	TOLEV	NEWLEG	FRFREQ	MINFO	SOLSIG	MSGFLG	CDETOL	
Type	I	I	I	I	I	I	F	
Default	2	0	1	0	0	0	10.0	

Optional Card 4

Card 4	1	2	3	4	5	6	7	8
Variable	PHSCHNG	DEMDEN						
Type	I	I						
Default	0	0						

VARIABLE	DESCRIPTION
NPOPT	Print suppression during input phase flag for the “d3hsp” file: EQ.0: no suppression, EQ.1: nodal coordinates, element connectivities, rigid wall definitions, nodal SPCs, initial velocities, initial strains, adaptive constraints, and SPR2/SPR3 constraints are not printed.
NEECHO	Print suppression during input phase flag for “echo” file: EQ.0: all data printed, EQ.1: nodal printing is suppressed, EQ.2: element printing is suppressed, EQ.3: both node and element printing is suppressed.
NREFUP	Flag to update reference node coordinates for beam formulations 1, 2, and 11. This option requires that each reference node is unique to the beam: EQ.0: Do not update reference node. EQ.1: Update reference node: This update is required for proper visualization of the beam cross-section orientation in LS-PrePost beyond the initial (t = 0) plot state. NREFUP does not affect the internal updating of the beam cross-section orientation in LS-DYNA.

VARIABLE **DESCRIPTION**

IACCOP

Flag to average or filter nodal accelerations output to file "nodout" and the time history database "d3thdt":

EQ.0: no average (default),

EQ.1: averaged between output intervals,

EQ.2: accelerations for each time step are stored internally and then filtered over each output interval using a filter from General Motors [Sala, Neal, and Wang, 2004] based on a low-pass Butterworth frequency filter. See also [Neal, Lin, and Wang, 2004]. DT2MS in *CONTROL_-TIMESTEP must be set to a negative value when IACCOP = 2 so that the maximum possible number of time steps for an output interval is known and adequate memory can be allocated. See Figure 12.15.

OPIFS

Output time interval for interface file written per *INTERFACE_-COMPONENT_option.

IPNINT

Flag controlling output of initial time step sizes for elements to "d3hsp":

EQ.0: 100 elements with the smallest time step sizes are printed.

EQ.1: Time step sizes for all elements are printed.

GT.1: IPNINT elements with the smallest time step sizes are printed.

IKEDIT

Problem status report interval steps to the "d3hsp" file. This flag is ignored if the "glstat" file is written, see *DATABASE_GLSTAT.

IFLUSH

Number of time steps interval for flushing I/O buffers. The default value is 5000. If the I/O buffers are not emptied and an abnormal termination occurs, the output files can be incomplete. The I/O buffers for restart files are emptied automatically whenever a restart file is written so these files are not affected by this option.

VARIABLE **DESCRIPTION**

IPRTF

Default print flag for “rbdout” and “matsum” files. This flag defines the default value for the print flag which can be defined in the part definition section, see *PART. This option is meant to reduce the file sizes by eliminating data which is not of interest.

EQ.0: write part data into both matsum and rbdout

EQ.1: write data into rbdout file only

EQ.2: write data into matsum file only

EQ.3: do not write data into rbdout and matsum

IERODE

Output eroded internal and kinetic energy into the “matsum” file. Also, output to the “matsum” file under the heading of part ID 0 is the kinetic energy from nonstructural mass, lumped mass elements and lumped inertia elements.

EQ.0: do not output extra data

EQ.1: output the eroded internal and kinetic energy

TET10S8

Output ten connectivity nodes for the 10-node solid tetrahedral and the eight connectivity nodes for the 8-node shell into “d3plot” database. The current default is set to 2 since this change in the database may make the data unreadable for many popular post-processors and older versions of LS-PrePost. The default will change to 1 later.

EQ.1: write the full node connectivity into the “d3plot” database

EQ.2: write only the corner nodes of the elements into the “d3plot” database

MSGMAX

Maximum number of each error/warning message

GT.0: number of messages to screen output, all messages written to d3hsp/messag

LE.0: number of messages to screen output and d3hsp/messag

EQ.0: default, 50

IPCURV

Flag to output digitized curve data to “messag” and d3hsp files.

EQ.0: off

EQ.1: on

VARIABLE **DESCRIPTION**

GMDT

Output interval for recorded motions from *INTERFACE_SSI_-AUX

IP1DBLT

Output information of 1D (bar-type) seatbelt created for 2D (shell-type) seatbelt to sbtout.

EQ.0: the analysis results of internally created 1D seatbelts are extracted and processed to yield the 2D belt information. The 2D belt information is stored in sbtout,

EQ.1: the analysis results of internally created 1D retractors and slip rings are stored in sbtout. Belt load can be yielded by *DATABASE_CROSS_SECTION.

EOCS

Elout Coordinate System: controls the coordinate system to be used when writing out shell data to the “elout” file:

EQ.0: default

EQ.1: local element coordinate system

EQ.2: global coordinate system

TOLEV

Timing Output Levels: controls the # of levels output in the timing summary at termination. The default is 2.

NEWLEG

New Legends: controls the format of the LEGEND section of various ASCII output files.

EQ.0: use the normal format

EQ.1: use the optional format with extra fields.

FRFREQ

Output frequency for failed element report, in cycles. The default is to report the summary every cycle on which an element fails. If > 1, the summary will be reported every FRFREQ cycles whether an element fails that cycle or not, provided some element has failed since the last summary report. Individual element failure is still reported as it occurs.

MINFO

Output penetration information for mortar contact after each implicit step, not applicable in explicit analysis. See remarks on mortar contact on *CONTACT card.

EQ.0: No information

EQ.1: Penetrations reported for each contact interface.

VARIABLE **DESCRIPTION**

SOLSIG

Flag to extrapolate stresses and other history variables for multi-integration point solids from integration points to nodes. These extrapolated nodal values replace the integration point values normally stored in d3plot. When a nonzero SOLSIG is invoked, NINTSLD in *DATABASE_EXTENT_BINARY should be set to 8 as any other value of NINTSLD will result in only one value being reported for each element. Supported solid formulations are: -1, -2, 2, 3, 4, 18, 16, 17, and 23.

NOTE: Do not use "Setting - Extrapolate" in LS-PrePost when this field, SOLSIG, is nonzero.

EQ.0: No extrapolation.

EQ.1: Extrapolate the stress for linear materials only.

EQ.2: Extrapolate the stress if plastic strain is zero.

EQ.3: Extrapolate the stress always.

EQ.4: Extrapolate all history variables.

MSGFLG

Flag for writing detailed error/warning messages to d3msg. MSGFLG has no affect on output of standard length error/warning messages; such messages are written to messag or mes****. NOTE: Most errors/warnings offer only standard length messages. Only a few also offer optional, detailed messages.

EQ.0: Do not write detailed messages to d3msg.

EQ.1: Write detailed messages to d3msg at the conclusion of the run. Each detailed message is written only once even in cases where the associated error or warning occurs multiple times.

CDETOL

Tolerance for output of *DEFINE_CURVE discretization warnings. After each curve is discretized, the resulting curve is evaluated at each of the original definition points, and the values compared. A warning will be issued for any curve where this comparison results in an error of more than CDETOL/100 × M, where the curve specific value M is computed as the median of the absolute values of the non-zero curve values.

VARIABLE **DESCRIPTION**

PHSCHNG Message to messag file when materials 216, 217, and 218 change phase..

EQ.0: (default) no message.

EQ.1: The time and element ID are written.

DEMDEN Output DEM density data to d3plot database..

EQ.0: (default) no output.

EQ.1: output data.

*CONTROL_PARALLEL

Purpose: Control parallel processing usage by defining the number of processors and invoking the optional consistency of the global vector assembly. This command applies only to shared memory parallel (SMP) LS-DYNA. It does not apply to distributed memory parallel (MPP) LS-DYNA.

Card	1	2	3	4	5	6	7	8
Variable	NCPU	NUMRHS	CONST	PARA				
Type	I	I	I	I				
Default	1	0	2	0				
Remarks		1	2	3				

VARIABLE **DESCRIPTION**

NCPU Number of cpus used.

(This parameter is disabled in 971 R5 and later versions. Set number of cpus using "ncpu=" on the execution line — see Execution Syntax section of Getting Started — or on the *KEYWORD line of the input.)

NUMRHS Number of right-hand sides allocated in memory:

EQ.0: same as NCPU, always recommended,

EQ.1: allocate only one.

CONST Consistency flag. (Including "ncpu=*n*" on the execution line or on the *KEYWORD line of input overrides CONST. The algebraic sign of *n* determines the consistency setting.)

EQ.1 or $n < 0$: on (recommended)

EQ.2 or $n > 0$: off, for a faster solution (default).

VARIABLE	DESCRIPTION
PARA	Flag for parallel force assembly if CONST=1. (Including "para=" on the execution line overrides PARA.) EQ.0: off EQ.1: on EQ.2: on

Remarks:

1. It is recommended to always set NUMRHS = NCPU since great improvements in the parallel performance are obtained since the force assembly is then done in parallel. Setting NUMRHS to one reduces storage by one right hand side vector for each additional processor after the first. If the consistency flag is active, i.e., CONST = 1, NUMRHS defaults to unity.

2. For any given problem with the consistency option off, i.e., CONST = 2, slight differences in results are seen when running the same job multiple times with the same number of processors and also when varying the number of processors. Comparisons of nodal accelerations often show wide discrepancies; however, it is worth noting that the results of accelerometers often show insignificant variations due to the smoothing effect of the accelerometers which are generally attached to nodal rigid bodies.

   The accuracy issues are not new and are inherent in numerical simulations of automotive crash and impact problems where structural bifurcations under compressive loads are common. This problem can be easily demonstrated by using a perfectly square thin-walled tubular beam of uniform cross section under a compressive load. Typically, every run on one processor that includes a minor input change (i.e., element or hourglass formulation) will produces dramatically different results in terms of the final shape, and, likewise, if the same problem is again run on a different brand of computer. If the same problem is run on multiple processors the results can vary dramatically from run to run WITH NO INPUT CHANGE. The problem here is due to the randomness of numerical round-off which acts as a trigger in a "perfect" beam.

   Since summations with (CONST=2) occur in a different order from run to run, the round-off is also random. The consistency flag, CONST=1, provides for identical results (or nearly so) whether one, two, or more processors are used while running in the shared memory parallel (SMP) mode. This is done by requiring that all contributions to global vectors be summed in a precise order independently of the number of processors used. When checking for consistent results, nodal displacements or element stresses should be compared. The

NODOUT and ELOUT files should be digit to digit identical. However, the GLSTAT, SECFORC, and many of the other ASCII files will not be identical since the quantities in these files are summed in parallel for efficiency reasons and the ordering of summation operations are not enforced. The biggest drawback of this option is the CPU cost penalty which is at least 15 percent if PARA=0 and is much less if PARA=1 and 2 or more processors are used. Unless the PARA flag is on (for non-vector processors), parallel scaling is adversely affected. The consistency flag does not apply to MPP parallel.

3. PARA set to 1 or 2 will cause the force assembly for the consistency option to be performed in parallel for the SMP version, so better scaling will be obtained. However, PARA = 1 will increase memory usage while PARA = 2 will not. This flag does not apply to the MPP version. If PARA = CONST = 0 and NUMRHS = NCPU the force assembly by default is done in parallel, but without consistency. The value of the flag may also be given by including “para=<value>” on the execution line, and the value given in this manner will override the value of PARA in *CONTROL_PARALLEL.

*CONTROL_PORE_FLUID

Purpose: Set parameters for pore water pressure calculations.

This control card is intended for soil analysis. However, other materials containing pore fluid could be treated by the same methods. The pore pressure capabilities invoked by this card are available in SMP and MPP versions of LS-DYNA, but are not available for implicit solutions. Furthermore, pore pressure capabilities are limited to a subset of 3-D solid Lagrangian element formulations, including solid formulations 1, 2, 4, 10, and 15.

LS-DYNA uses Terzaghi's Effective Stress to model materials with pore pressure. The pore fluid and soil skeleton are assumed to occupy the same volume and to carry loads in parallel. Thus, the total stress in an element is the sum of the "effective stress" in the soil skeleton, plus the hydrostatic stress in the pore fluid. LS-DYNA calculates the "effective stress" with standard material models. The pore fluid treatment, then, is independent of material model. The pore pressure is calculated at nodes, and interpolated onto the elements. The pore fluid's hydrostatic stress is equal to the negative of the element pore pressure.

Card 1	1	2	3	4	5	6	7	8
Variable	ATYPE	(blank)	WTABLE	PF_RHO	GRAV	PF_BULK	OUTPUT	TMF
Type	I	F	F	F	F	F	I	F
Default	0	0.0	0.0	(none)	(none)	(none)	0	1.0

Card 2	1	2	3	4	5	6	7	8
Variable	TARG	FMIN	FMAX	FTIED	CONV	CONMAX	ETERM	THERM
Type	F	F	F	F	F	F	F	F
Default	0.0	0.0	0.0	0.0	1.E-4	1.E20	0.0	0.0

Card 3 is optional

Card 3	1	2	3	4	5	6	7	8
Variable	ETFLAG							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
ATYPE	Analysis type for pore water pressure calculations: EQ.0: No pore water pressure calculation. EQ.1: Undrained analysis, EQ.2: Drained analysis, EQ.3: Time dependent consolidation (coupled), EQ.4: Consolidate to steady state (uncoupled), EQ.5: Drained in dynamic relaxation, undrained in transient, EQ.6: As 4 but do not check convergence, continue to end time.
WTABLE	Default z-coordinate of water table (where pore pressure is zero).
PF_RHO	Default density for pore water.
GRAV	Gravitational acceleration used to calculate hydrostatic pore water pressure.
PF_BULK	Default bulk modulus of pore fluid (stress units).
OUTPUT	Output flag controlling stresses to D3PLOT and D3THDT binary files: EQ.0: total stresses are output EQ.1: effective stresses are output, see notes

VARIABLE **DESCRIPTION**

TMF

Initial Time Magnification factor on seepage (ATYPE = 3,4 only).

GT.0: Factor (can be used with automatic control, see TARG, FMIN, FMAX).

LT.0: Load Curve ID (see *DEFINE_CURVE) giving Time Magnification Factor versus analysis time.

TARG

Target for maximum change of excess pore pressure at any node, per timestep. If the actual change falls below the target, the time factor on the seepage calculation will be increased (see notes). If zero, the constant value of TMF is used. If non-zero, TMF is taken as the initial factor.

FMIN

Minimum time factor on seepage calculation

FMAX

Maximum time factor on seepage calculation

FTIED

Analysis type for pore water pressure calculations:

EQ.0.0: Tied contacts act as impermeable membranes,

EQ.1.0: Fluid may flow freely through tied contacts.

CONV

Convergence tolerance for ATYPE = 4. Maximum head change per time step at any node as measured in units of characteristic length, l.

$$l = \frac{p}{\rho g}$$

where,

ρ = pore fluid density, PF_RHO

g = gravitational acceleration.

CONMAX

Maximum factor on permeability with ATYPE = -4

ETERM

Event time termination (ATYPE = 3)

THERM

Thermal expansion: Volumetric strain per degree increase for undrained soil.

ETFLAG

Flag for interpretation of time etc (see notes):

EQ.0: Time means analysis time,

EQ.1: Time means event time.

Analysis Types (see ATYPE):

Undrained

For analyses of the "undrained" type the pore fluid is trapped within the material. Volume changes result in pore pressure changes. This approximation is used to simulate the effect of rapidly-applied loads on relatively impermeable soil.

Drained

For analyses of the "drained" type the pore fluid is free to move within the material such that the user-defined pressure-versus-z-coordinate relationship is always maintained. This approximation is used to model high-permeability soils.

Time-dependent consolidation

For the analysis type "time dependent consolidation," pressure gradients cause pore fluid to flow through the material according to Darcy's law:

$$v = \kappa \nabla (p + z)$$

where,

v = fluid velocity vector

κ = permeability

p = pressure head

z = z-coordinate.

Net inflow or outflow at a node leads to a theoretical volume gain or loss. The analysis is coupled, i.e. any difference between actual and theoretical volume leads to pore pressure change, which in turn affects the fluid flow. The result is a prediction of response-versus-time.

Steady-state consolidation

For the analysis type "steady-state consolidation," an iterative method is used to calculate the steady-state pore pressure. The analysis is uncoupled, i.e. only the final state is meaningful, not the response-versus-time.

Time factoring:

Consolidation occurs over time intervals of days, weeks or months. To simulate this process using explicit time integration, a time factor is used. The permeability of the soil is increased by the time factor so that consolidation occurs more quickly. The output times in the D3PLOT and D3THDT files are modified to reflect the time factor. The factored time ("Event Time") is intended to represent the time taken in the real-life consolidation process and will usually be much larger than the analysis time (the analysis time is the sum of the LS-DYNA timesteps). The time factor may be chosen explicitly (using TMF) but it is recommended to use automatic factoring instead. The automatic scheme adjusts the time factor according to how quickly the pore pressure is changing; usually at the start of consolidation the pore pressure changes quickly and the time factor is low; the time factor increases gradually as the rate of pore pressure change reduces. Automatic time factoring is input by setting TARG (the target pore pressure head change per timestep) and maximum and minimum allowable time factors, for example TARG = 0.001 to 0.01m head, FMIN = 1.0, FMAX = 1.0e6. Optimum settings for these are model-dependent.

Loading, other input data from loadcurves, and output time-intervals on *DATABASE cards by default use the analysis time (for example, the x-axis of a loadcurve used for pressure loading is analysis time). When performing consolidation with automatic time-factoring, the relationship between analysis time and event time is unpredictable. Termination based on event time may be input using ETERM.

It may also be desired to apply loads as functions of event time rather than analysis time, since the event time is representative of the real-life process. By setting ETFLAG = 2, the time axis of all load curves used for any type of input-versus-time, and output intervals on *DATABASE cards, will be interpreted as event time. This method also allows consolidation to be used as part of a staged construction sequence – when ETFLAG = 2, the stages begin and end at the “real time” stage limits and input curves of pore pressure analysis type vs. time may be used to enforce, for example, consolidation in some stages, and undrained behavior in others.

Output:

Extra variables for solid elements are automatically written to the d3plot and d3thdt files when the model contains *CONTROL_PORE_FLUID. In LS971 R4 onwards, 5 additional extra variables are written, of which the first is the pore pressure in stress units. In LS971 R3, 15 additional extra variables are written, of which the seventh is pore pressure in stress units. These follow any extra variables requested by the user, e.g. if the user requested 3 extra variables, then in LS971 there will be a total of 8 extra variables of which the fourth is pore pressure.

Further optional output to d3plot and d3thdt files is available for nodal pore pressure variables – see *DATABASE_PWP_OUTPUT.

For time-dependent and steady-state consolidation, information on the progress of the analysis is written to d3hsp file.

Remarks:

1. Tied Contacts. By default, the mesh discontinuity at a tied contact will act as a barrier to fluid flow. If the flag FTIED is set to 1, then pore fluid will be trans-

mitted across tied nodes in tied contacts (*CONTACT_TIED_SURFACE_TO_-SURFACE and *CONTACT_TIED_NODES_TO_SURFACE, including_OFFSET and non-_OFFSET types). This algorithm has an effect only when the analysis type of at least one of the contacting parts is 3, 4 or 6.

2. Thermal. Note that this property is for VOLUMETRIC strain increase. Typical thermal expansion coefficients are linear; the volumetric expansion will be three times the linear thermal expansion coefficient. Regular thermal expansion coefficients (e.g. on *MAT or *MAT_ADD_THERMAL_EXPANSION) apply to the soil skeleton and to drained parts. Pore pressure can be generated due to the difference of expansion coefficients of the soil skeleton and pore fluid.

3. Part Associativity. Pore pressure is a nodal variable, but analysis type and other pore pressure related inputs are properties of parts. When a node is shared by elements of different parts, and those parts have different pore pressure inputs, the following rules are followed to determine which part's properties should be applied to the node.

   a) Dry parts (i.e. parts without a *BOUNDARY_PORE_FLUID card) will never be used (lowest priority).

   b) If a part is initially dormant (due to staged construction inputs), it has next-lowest priority

   c) Parts with analysis type = drained have highest priority.

   d) Next, higher permeability gives higher priority

   e) If two or more parts have equal-highest priority at a node, the part with lowest ID will win.

4. Related Cards:

*BOUNDARY_PORE_FLUID.

(This card is essential since without this card, no parts will have pore fluid.)

BOUNDARY_PWP_*OPTION

*DATABASE_PWP_OUTPUT

*DATABASE_PWP_FLOW

*MAT_ADD_PERMEABILITY

*CONTROL_PORE_AIR

Purpose: Set parameters for pore air pressure calculations.

Card 1	1	2	3	4	5	6	7	8
Variable	AIR_RO	AIR_P	ETERM	ANAMSG				
Type	F	F	F	I				
Default	(none)	(none)	endtim	0				

VARIABLE **DESCRIPTION**

PA_RHO Density of atmospheric air, = 1.184 kg/m^3 at 25°C

PA_PATM Pressure of atmospheric air, = 101.325 kPa at 25°C

ETERM Event termination time, default to ENDTIME of *CONTROL_-TERMINATION

ANAMSG Flag to turn off the printing of pore air analysis status message, including the analysis time, the node with the highest pressure change.

EQ.0: Status messages are printed, the default value.

EQ.1: Status messages are not printed

*CONTROL_REFINE_ALE

Purpose: Refine ALE hexahedral solid elements locally. Each parent element is replaced by 8 child elements with a volume equal to 1/8th the parent volume. If only the 1st card is defined, the refinement occurs during the initialization. The 2nd card defines a criterion CRITRF to automatically refine the elements during the run. If the 3rd card is defined, the refinement can be removed if a criterion CRITRM is reached: the child elements can be replaced by their parents.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	TYPE	NLVL	MMSID	IBOX	IELOUT		
Type	I	I	I	I	I	I		
Default	none	0	1	0	0	0		

Remaining cards are optional.†

Automatic refinement card. Optional card for activating automatic refinement whereby each element satisfying certain criteria is replaced by a cluster of 8 child elements

Card 2	1	2	3	4	5	6	7	8
Variable	NTOTRF	NCYCRF	CRITRF	VALRF	BEGRF	ENDRF	LAYRF	DELAYRF
Type	I	F	I	F	F	F	I	F
Default	0	0.0	0	0.0	0.0	0.0	0	0.0

Automatic Refinement Remove Card. Optional card for activating automatic refinement removal whereby, when, for a cluster of 8 child elements, certain criteria are satisfied the clusters is replaced by its parent.

Card 3	1	2	3	4	5	6	7	8
Variable	MAXRM	NCYCRM	CRITRM	VALRM	BEGRM	ENDRM	MMSRM	DELAYRM
Type	I	F	I	F	F	F	I	F
Default	0	0.0	0	0.0	0.0	0.0	0	0.0

VARIABLE	DESCRIPTION
ID	Set ID.
TYPE	Set type: EQ.0: ALE Part Set, EQ.1: ALE Part, EQ.2: Lagrangian Part Set coupled to ALE (see Remarks 1 and 2), EQ.3: Lagrangian Part coupled to ALE (see Remarks 1 and 2), EQ.4: Lagrangian Shell Set coupled to ALE (see Remarks 1 and 2), EQ.5: ALE Solid Set.
NLVL	Number of refinement levels (see Remark 3).
MMSID	Multi-Material Set ID (see Remark 4): LT.0: only ALE elements with all the multi-material groups listed in *SET_MULTI-MATERIAL_GROUP_LIST can be refined (or removed otherwise) GT.0: ALE elements with at least one of the multi-material groups can be refined (or removed)
IBOX	Box ID (See *DEFINE_BOX) defining a region in which the ALE elements are refined.
IELOUT	Flag to handle child data in elout (see Remarks 10 and 11).

VARIABLE **DESCRIPTION**

NTOTRF Total number of ALE elements to refine (see Remark 5):

GT.0: Number of elements to refine

EQ.0: NTOTRF = number of solid elements in IBOX (see Remark 2)

LT.0: |NTOTRF| is the id of *CONTROL_REFINE_MPP_DISTRIBUTION that computes the number of extra elements required by processors.

NCYCRF Number of cycles between each refinement.

LT.0: |NCYCRF| is the time interval

CRITRF Refinement criterion:

EQ.0: static refinement (as if only the 1st card is defined),

EQ.1: Pressure (if pressure > VALRF),

EQ.2: Relative Volume (if V/Vo < VALRF) ,

EQ.3: Volume Fraction (if Volume fraction > VALRF),

EQ.5: User defined criterion. The fortran routine alerfn_criteria5 in the file dynrfn_user.f should be used to develop the criterion. The file is part of the general package usermat.

VALRF Criterion value to reach for the refinement.

BEGRF Time to begin the refinement.

ENDRF Time to end the refinement.

LAYRF Number of element layers to refine around a element reaching the refinement criterion (see Remark 6).

DELAYRF Period of time after removing the refinement of an element, during which this element will not be refined again.

MAXRM Maximum number of child clusters to remove (see Remark 9):

LT.0: for the whole run,

GT.0: every NCYCRM cycles

VARIABLE DESCRIPTION

NCYCRM

Number of cycles between each check for refinement removal:

LT.0: |NCYCRM| is the time interval

CRITRM

Criterion for refinement removal:

EQ.0: no refinement removal (as if only the 1st and 2nd card are defined),

EQ.1: Pressure (if pressure < VALRM),

EQ.2: Relative Volume (if V/Vo > VALRM) ,

EQ.3: Volume Fraction (if Volume fraction < VALRM),

EQ.5: User defined criterion. The fortran routine alermv_criteria5 in the file dynrfn_user.f should be used to develop the criterion. The file is part of the general package usermat.

VALRM

Criterion value to reach in each child element of a cluster for its removal (child elements replaced by parent element)..

BEGRM

Time to begin the check for refinement removal:

LT.0: |BEGRM| represents a critical percent of NTOTRF below which the check for refinement removal should begin (0.0 < |BEGRM| < 1.0). (See Remark 8).

ENDRM

Time to end the check for refinement removal.

MMSRM

Multi-Material Set ID for the refinement removal. (See Remark 7)

LT.0: | MMSRM | represents the radius of a sphere centered on a newly refined element, in which the refinement can not be removed.

DELAYRM

Period of time after refining an element, during which this refinement will not be removed.

Remarks:

1. If only the 1st card is defined, only TYPE = 0, 1, 5 can be defined.

2. *CONSTRAINED_LAGRANGE_IN_SOLID needs to be defined for TYPE = 2, 3, 4. If an ALE element has at least one coupling point (see NQUAD in *CONSTRAINED_LAGRANGE_IN_SOLID), this element will be selected to be re-

fined (or removed). The number of elements to refine is computed during the initialization. NTOTRF can be zero. Otherwise it can used to add more elements.

3. If NLVL = 1, there is only one level of refinement: the ALE elements in *ELEMENT_SOLID are the only ones to be replaced by clusters of 8 child elements. If NLVL > 1, there are several levels of refinement: not only the initial ALE elements in *ELEMENT_SOLID are refined but also their child elements.

4. If only the 1st card is defined, a multi-material set id is not used. It can be left to zero. For the 2nd and 3rd cards, MMSID is the ID of *SET_MULTI-MATERIAL_-GROUP_LIST in which the multi-material group ids (as defined in *ALE_MULTI-MATERIAL_GROUP) are listed to select the ALE elements to be refined (or removed). If MMSID < 0, only mixed ALE elements containing all the multi-material groups can be refined. Otherwise clusters of 8 elements without a mix of the listed multi-material groups can be removed.

5. NTOTRF defines the total number of ALE elements to be refined. So for example NTOTRF = 100 with NLVL = 1 means that only 100 ALE elements can be replaced by 800 ALE finer elements (or 100 clusters of 8 child elements). For NLVL = 2, these 800 elements can be replaced by 6400 finer elements.

6. If an element is refined, it is possible to refine the neighbor elements as well. LAYRF defines the number of neighbor layers to refine. For example, LAYRF = 2 for an element at the center of a block of 5 × 5 × 5 elements will refine these 125 elements.

7. If MMSRM = 0, MMSID defines the multi-material region where the check for refinement removal should occur. If MMSRM is defined, only ALE child elements fully filled by the multi-material groups listed by the set MMSRM can be removed (if the refinement removal criterion is reached).

8. If BEGRM < 0, the check for refinement removal is activated when the number of 8-element clusters for the refinement is below a limit defined by |BEGRM|*NTOTRF. If |BEGRM| = 0.1, it means that the check for refinement removal starts when 90% of the stock of clusters is used for the refinement.

9. MAXRM < 0 defines a total number of child clusters to remove for the whole run. If positive, MAXRM defines an upper limit for the number of child clusters to remove every NCYCRM cycles.

10. If only the 1st card is defined, the code for IELOUT is always activated. Since the refinement occurs during the initialization, every refined element is replaced by its 8 children in the set defined for *DATABASE_ELOUT.

11. If there are more than 1 line, the code for IELOUT is activated if the flag is equal to 1. Since the refinement occurs during the run, the parent ids in the set defined for *DATABASE_ELOUT are duplicated 8^{NLVL} times. The points of integration in the elout file are incremented to differentiate the child contributions to the database.

*CONTROL_REFINE_ALE2D

Purpose: Refine ALE quadrilateral shell elements locally. Each parent element is replaced by 4 child elements with a volume equal to 1/4th the parent volume. If only the 1st card is defined, the refinement occurs during the initialization. The 2nd card defines a criterion CRITRF to automatically refine the elements during the run. If the 3rd card is defined, the refinement can be removed if a criterion CRITRM is reached: the child elements can be replaced by their parents.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	TYPE	NLVL	MMSID	IBOX	IELOUT		
Type	I	I	I	I	I	I		
Default	none	0	1	0	0	0		

Remaining cards are optional.†

Automatic refinement card. Optional card for activating automatic refinement whereby each element satisfying certain criteria is replaced by a cluster of 4 child elements

Card 2	1	2	3	4	5	6	7	8
Variable	NTOTRF	NCYCRF	CRITRF	VALRF	BEGRF	ENDRF	LAYRF	
Type	I	F	I	F	F	F	I	
Default	0	0.0	0	0.0	0.0	0.0	0	

Automatic Refinement Remove Card. Optional card for activating automatic refinement removal whereby, when, for a cluster of 4 child elements, certain criteria are satisfied the clusters is replaced by its parent.

Card 2	1	2	3	4	5	6	7	8
Variable	MAXRM	NCYCRM	CRITRM	VALRM	BEGRM	ENDRM	MMSRM	
Type	I	F	I	F	F	F	I	
Default	0	0.0	0	0.0	0.0	0.0	0	

VARIABLE	DESCRIPTION
ID	Set ID.
TYPE	Set type: EQ.0: ALE Part Set, EQ.1: ALE Part, EQ.2: Lagrangian Part Set coupled to ALE (see Remarks 1 and 2), EQ.3: Lagrangian Part coupled to ALE (see Remarks 1 and 2), EQ.4: Lagrangian Shell Set coupled to ALE (see Remarks 1 and 2), EQ.5: ALE Shell Set.
NLVL	Number of refinement levels (see Remark 3).
MMSID	Multi-Material Set ID (see Remark 4): LT.0: only ALE elements with all the multi-material groups listed in *SET_MULTI-MATERIAL_GROUP_LIST can be refined (or removed otherwise) GT.0: ALE elements with at least one of the multi-material groups can be refined (or removed)
IBOX	Box ID (See *DEFINE_BOX) defining a region in which the ALE elements are refined.
IELOUT	Flag to handle child data in elout (see Remarks 10 and 11).

VARIABLE	DESCRIPTION
NTOTRF	Total number of ALE elements to refine (see Remark 5): GT.0: Number of elements to refine EQ.0: NTOTRF = number of shell elements in IBOX (see Remark 2)
NCYCRF	Number of cycles between each refinement. LT.0: |NCYCRF| is the time interval
CRITRF	Refinement criterion: EQ.0: static refinement (as if only the 1st card is defined), EQ.1: Pressure (if pressure > VALRF), EQ.2: Relative Volume (if V/Vo < VALRF) , EQ.3: Volume Fraction (if Volume fraction > VALRF), EQ.5: User defined criterion: The fortran routine al2rfn_ criteria5 in the file dynrfn_user.f should be used to develop the criterion. The file is part of the general package usermat.
VALRF	Criterion value to reach for the refinement.
BEGRF	Time to begin the refinement.
ENDRF	Time to end the refinement.
LAYRF	Number of element layers to refine around a element reaching the refinement criterion (see Remark 6).
MAXRM	Maximum number of child clusters to remove (see Remark 9): LT.0: for the whole run, GT.0: every NCYCRM cycles
NCYCRM	Number of cycles between each check for refinement removal: LT.0: |NCYCRM| is the time interval

VARIABLE	DESCRIPTION
CRITRM	Criterion for refinement removal: EQ.0: no refinement removal (as if only the 1st and 2nd card are defined), EQ.1: Pressure (if pressure < VALRM), EQ.2: Relative Volume (if V/Vo > VALRM) , EQ.3: Volume Fraction (if Volume fraction < VALRM), EQ.5: User defined criterion: The fortran routine al2rmv_criteria5 in the file dynrfn_user.f should be used to develop the criterion. The file is part of the general package usermat.
VALRM	Criterion value to reach in each child element of a cluster for its removal (child elements of a cluster replaced by parent element).
BEGRM	Time to begin the check for refinement removal: LT.0: |BEGRM| represents a critical percent of NTOTRF below which the check for refinement removal should begin (0.0 < |BEGRM| < 1.0). (See Remark 8).
ENDRM	Time to end the check for refinement removal.
MMSRM	Multi-Material Set ID for the refinement removal. (See Remark 7)

Remarks:

1. If only the 1st card is defined, only TYPE = 0,1,5 can be defined.

2. *CONSTRAINED_LAGRANGE_IN_SOLID needs to be defined for TYPE = 2,3,4. If an ALE element has at least one coupling point (see NQUAD in *CONSTRAINED_LAGRANGE_IN_SOLID), this element will be selected to be refined (or removed).

3. If NLVL = 1, there is only one level of refinement: the ALE elements in *ELEMENT_SHELL are the only ones to be replaced by clusters of 4 child elements. If NLVL > 1, there are several levels of refinement: not only the initial ALE elements in *ELEMENT_SHELL are refined but also their child elements. If NLVL = 2 for example, the initial ALE elements can be replaced by clusters of 16 child elements.

4. If only the 1st card is defined, a multi-material set id is not used. It can be left to zero. For the 2nd and 3rd cards, MMSID is the ID of *SET_MULTI-MATERIAL_-

GROUP_LIST in which the multi-material group ids (as defined in *ALE_MULTI-MATERIAL_GROUP) are listed to select the ALE elements to be refined (or removed). If MMSID < 0, only mixed ALE elements containing all the multi-material groups can be refined. Otherwise clusters of 4 elements without a mix of the listed multi-material groups can be removed.

5. NTOTRF defines the total number of ALE elements to be refined. So for example NTOTRF = 100 means that only 100 ALE elements will be replaced by 400 ALE finer elements (or 100 clusters of 4 child elements). For NLVL = 2, these 400 elements can be replaced by 1600 finer elements.

6. If an element is refined, it is possible to refine the neighbor elements as well. LAYRF defines the number of neighbor layers to refine. For example, LAYRF = 2 for an element at the center of a block of 5 × 5 elements will refine these 25 elements.

7. If MMSRM = 0, MMSID defines the multi-material region where the check for refinement removal should occur. If MMSRM is defined, only ALE child elements fully filled by the multi-material groups listed by the set MMSRM can be removed (if the refinement removal criterion is reached).

8. If BEGRM < 0, the check for refinement removal is activated when the number of 4-element clusters for the refinement is below a limit defined by |BEGRM|*NTOTRF. If |BEGRM| = 0.1, it means that the check for refinement removal starts when 90% of the stock of clusters is used for the refinement.

9. MAXRM < 0 is the exact opposite of NTOTRF > 0 and it defines a total number of child clusters to remove for the whole run. If positive, MAXRM defines an upper limit for the number of child clusters to remove every NCYCRM cycles

10. If only the 1st card is defined, the code for IELOUT is always activated. Since the refinement occurs during the initialization, every refined element is replaced by its 4 children in the set defined for *DATABASE_ELOUT.

11. If there are more than 1 line, the code for IELOUT is activated if the flag is equal to 1. Since the refinement occurs during the run, the parent ids in the set defined for *DATABASE_ELOUT are duplicated 4^{NLVL} times. The points of integration in the elout file are incremented to differentiate the child contributions to the database.

*CONTROL_REFINE_MPP_DISTRIBUTION

Purpose: Distribute the elements for the refinement over the MPP processes. This keyword addresses to the following situation:

> If TYPE = 2, 3, 4 in *CONTROL_REFINE_ALE, the refinement occurs around a structure. The number of elements for this refinement is computed for each process according the initial position of the structure in each MPP subdomain (after the MPP decomposition of the ALE mesh during the phase 3 of the initialization, each process has a subdomain that is a portion of the ALE mesh). If the structure is not in a subdomain, the related process receives no extra element for the refinement. If the structure moves into this subdomain during the computation, the refinement around the structure can not occur. To avoid this problem, the structure can be considered within a box (the structure maxima and minima give the box dimensions and positions). This box moves and expands during the computation to keep the structure inside. An estimation of the maximal displacement and expansion will allow the code to evaluate which subdomains the structure will likely cross and how many extra elements a process may need to carry out the refinement.

The computation of the number of extra elements per process occurs in 2 steps:

- If a file called "refine_mpp_distribution" does not exist in the working directory, it will be created to list the number of elements by process. Each line in this file matches a process rank (starting from 0). After the phase 3 of the MPP decomposition, the run terminates as if *CONTROL_MPP_DECOMPOSITION_SHOW was activated.
- The model can be run again and the file "refine_mpp_distribution" will be read to allocate the memory for the extra elements and distribute them across the processes.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	DX	DY	DZ	EX	EY	EZ	
Type	I	F	F	F	F	F	F	
Default	none	0.0	0.0	0.0	1.0	1.0	1.0	

VARIABLE	DESCRIPTION
ID	ID = -NTOTRF in *CONTROL_REFINE_ALE

VARIABLE	DESCRIPTION
DX	Dimensionless *x*-displacement of the box. (see Remark 1).
DY	Dimensionless *y*-displacement of the box. (see Remark 1).
DZ	Dimensionless *z*-displacement of the box. (see Remark 1).
EX	Dimensionless *x*-expansion of the box. (see Remark 2).
EY	Dimensionless *y*-expansion of the box. (see Remark 2).
EZ	Dimensionless *z*-expansion of the box. (see Remark 2).

Remarks:

1. **Box Displacements.** DX, DY and DZ are the maximal displacements of the box center. These displacements are ratio of the box dimensions. If, for example, the largest length of the structure in the x-direction is 10m and the maximal displacement in this direction is 2m, DX should be equal to 0.2

2. **Maximal Box Dilations.** EX, EY and EZ represent the maximal dilatations of the box in each direction. These expansions are ratio of the box dimensions. The box expands around its center. If, for example, the maximal thickness of a structure along z is 1cm and the structure deforms 30 times the thickness in z-direction, EZ should be equal to 30 and DZ=15 accounts for the box center motion. The x-y plane is a plane of symmetry for this deformation, DZ can be zero.

*CONTROL_REFINE_SHELL

Purpose: Refine quadrilateral shell elements locally. Each parent element is replaced by 4 child elements with a volume equal to 1/4th the parent volume. If only the 1st card is defined, the refinement occurs during the initialization. The 2nd card defines a criterion CRITRF to automatically refine the elements during the run. If the 3rd card is defined, the refinement can be removed if a criterion CRITRM is reached: the child elements can be replaced by their parents.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	TYPE	NLVL	IBOX	IELOUT			
Type	I	I	I	I	I			
Default	none	0	1	0	0			

Remaining cards are optional.†

Automatic refinement card. Optional card for activating automatic refinement whereby each element satisfying certain criteria is replaced by a cluster of 4 child elements

Card 2	1	2	3	4	5	6	7	8
Variable	NTOTRF	NCYCRF	CRITRF	VALRF	BEGRF	ENDRF	LAYRF	
Type	I	F	I	F	F	F	I	
Default	0	0.0	0	0.0	0.0	0.0	0	

Automatic Refinement Remove Card. Optional card for activating automatic refinement removal whereby, when, for a cluster of 4 child elements, certain criteria are satisfied the clusters is replaced by its parent.

Card 3	1	2	3	4	5	6	7	8
Variable	MAXRM	NCYCRM	CRITRM	VALRM	BEGRM	ENDRM		
Type	I	F	I	F	F	F		
Default	0	0.0	0	0.0	0.0	0.0		

VARIABLE	DESCRIPTION
ID	Set ID. LT.0: parent elements can be hidden in lsprepost as they are replaced by their children.
TYPE	Set type:: EQ.0: Part Set, EQ.1: Part, EQ.2: Shell Set.
NLVL	Number of refinement levels (see Remark 1).
IBOX	Box ID (See *DEFINE_BOX) defining a region in which the elements are refined.
IELOUT	Flag to handle child data in the elout file (see Remarks 6 and 7).
NTOTRF	Total number of elements to refine (see Remark 2): GT.0: Number of elements to refine EQ.0: NTOTRF = number of shell elements in IBOX
NCYCRF	Number of cycles between each refinement. LT.0: |NCYCRF| is the time interval

VARIABLE **DESCRIPTION**

CRITRF

Refinement criterion:

EQ.0: static refinement (as if only the 1st card is defined),

EQ.1: Pressure (if pressure > VALRF),

EQ.2: undefined ,

EQ.3: Von Mises criterion,

EQ.4: Criterion similar to ADPOPT = 4 in *CONTROL_ADAPTIVE (VALRF = ADPTOL),

EQ.5: User defined criterion: The fortran routine shlrfn_ criteria5 in the file dynrfn_user.f should be used to develop the criterion. The file is part of the general package usermat.

VALRF

Criterion value to reach for the refinement.

BEGRF

Time to begin the refinement.

ENDRF

Time to end the refinement.

LAYRF

Number of element layers to refine around a element reaching the refinement criterion (see Remark 3).

MAXRM

Maximum number of child clusters to remove (see Remark 5):

LT.0: for the whole run,

GT.0: every NCYCRM cycles

NCYCRM

Number of cycles between each check for refinement removal:

LT.0: |NCYCRM| is the time interval

VARIABLE DESCRIPTION

CRITRM

Criterion for refinement removal:

EQ.0: no refinement removal (as if only the 1st and 2nd card are defined),

EQ.1: Pressure (if pressure < VALRM),

EQ.2: undefined,

EQ.3: Von Mises criterion,

EQ.4: Criterion similar to ADPOPT = 4 in *CONTROL_ADAPTIVE (VALRF = ADPTOL),

EQ.5: User defined criterion: The fortran routine shlrmv_criteria5 in the file dynrfn_user.f should be used to develop the criterion. The file is part of the general package usermat.

VALRM

Criterion value to reach in each child elements of a cluster for its removal (child elements replaced by parent element).

BEGRM

Time to begin the check for refinement removal.

LT.0: |BEGRM| represents a critical percent of NTOTRF below which the check for refinement removal should begin (0.0 < |BEGRM| < 1.0). (See Remark 4).

ENDRM

Time to end the check for refinement removal.

Remarks:

1. If NLVL = 1, there is only one level of refinement: the elements in *ELEMENT_SHELL are the only ones to be replaced by clusters of 4 child elements. If NLVL > 1, there are several levels of refinement: not only the initial elements in *ELEMENT_SHELL are refined but also their child elements. If NLVL = 2 for example, the initial elements can be replaced by clusters of 16 child elements.

2. NTOTRF defines the total number of elements to be refined. So for example NTOTRF = 100 with NLVL = 1 means that only 100 elements can be replaced by 400 finer elements (or 100 clusters of 4 child elements). For NLVL = 2, these 400 elements can be replaced by 1600 finer elements.

3. If an element is refined, it is possible to refine the neighbor elements as well. LAYRF defines the number of neighbor layers to refine. For example, LAYRF = 2 for an element at the center of a block of 5 × 5 elements will refine these 25 elements.

4. If BEGRM < 0, the check for refinement removal is activated when the number of 4-element clusters for the refinement is below a limit defined by |BEGRM| × NTOTRF. If |BEGRM| = 0.1, it means that the check for refinement removal starts when 90% of the stock of clusters is used for the refinement.

5. MAXRM < 0 defines a total number of child clusters to remove for the whole run. If positive, MAXRM defines an upper limit for the number of child clusters to remove every NCYCRM cycles.

6. If only the 1st card is defined, the code for IELOUT is always activated. Since the refinement occurs during the initialization, every refined element is replaced by its 4 children in the set defined for *DATABASE_ELOUT.

7. If there are more than 1 line, the code for IELOUT is activated if the flag is equal to 1. Since the refinement occurs during the run, the parent ids in the set defined for *DATABASE_ELOUT are duplicated 4^{NLVL} times. The points of integration in the elout file are incremented to differentiate the child contributions to the database.

*CONTROL_REFINE_SOLID

Purpose: Refine hexahedral solid elements locally. Each parent element is replaced by 8 child elements with a volume equal to 1/8th the parent volume. If only the 1st card is defined, the refinement occurs during the initialization. The 2nd card defines a criterion CRITRF to automatically refine the elements during the run. If the 3rd card is defined, the refinement can be removed if a criterion CRITRM is reached: the child elements can be replaced by their parents.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	TYPE	NLVL	IBOX	IELOUT			
Type	I	I	I	I	I			
Default	none	0	1	0	0			

Remaining cards are optional.†

Automatic refinement card. Optional card for activating automatic refinement whereby each element satisfying certain criteria is replaced by a cluster of 8 child elements

Card 2	1	2	3	4	5	6	7	8
Variable	NTOTRF	NCYCRF	CRITRF	VALRF	BEGRF	ENDRF	LAYRF	
Type	I	F	I	F	F	F	I	
Default	0	0.0	0	0.0	0.0	0.0	0	

Automatic Refinement Remove Card. Optional card for activating automatic refinement removal whereby, when, for a cluster of 8 child elements, certain criteria are satisfied the clusters is replaced by its parent.

Card 2	1	2	3	4	5	6	7	8
Variable	MAXRM	NCYCRM	CRITRM	VALRM	BEGRM	ENDRM		
Type	I	F	I	F	F	F		
Default	0	0.0	0	0.0	0.0	0.0		

VARIABLE	DESCRIPTION

ID

Set ID.

LT.0: parent elements can be hidden in lsprepost as they are replaced by their children.

TYPE

Set type:

EQ.0: Part Set,

EQ.1: Part,

EQ.2: Solid Set.

NLVL

Number of refinement levels. See Remark 1.

IBOX

Box ID (See *DEFINE_BOX) defining a region in which the elements are refined.

IELOUT

Flag to handle child data in elout. See Remarks 6 and 7.

NTOTRF

Total number of elements to refine. See Remark 2.

GT.0: Number of elements to refine

EQ.0: NTOTRF = number of solid elements in IBOX

NCYCRF

Number of cycles between each refinement.

LT.0: |NCYCRF| is the time interval

VARIABLE **DESCRIPTION**

CRITRF Refinement criterion:

EQ.0: static refinement as if only the 1st card is defined,

EQ.1: Pressure, if pressure > VALRF,

EQ.2: undefined ,

EQ.3: Von Mises criterion.

EQ.5: User defined criterion. The fortran routine sldrfn_criteria5 in the file dynrfn_user.f should be used to develop the criterion. The file is part of the general package usermat.

VALRF Criterion value to reach for the refinement.

BEGRF Time to begin the refinement.

ENDRF Time to end the refinement.

LAYRF Number of element layers to refine around an element reaching the refinement criterion. See Remark 3.

MAXRM Maximum number of child clusters to remove. See Remark 5.

LT.0: for the whole run,

GT.0: every NCYCRM cycles

NCYCRM Number of cycles between each check for refinement removal:

LT.0: |NCYCRM| is the time interval

CRITRM Criterion for removal of refinement:

EQ.0: no removal of refinement as if only the 1st and 2nd card are defined,

EQ.1: Pressure if pressure < VALRM,

EQ.2: undefined,

EQ.3: Von Mises criterion.

EQ.5: User defined criterion. The FORTRAN routine sldrmv_criteria5 in the file dynrfn_user.f should be used to develop the criterion. The file is part of the general package usermat.

VARIABLE	DESCRIPTION
VALRM	Criterion value to reach in each child element of a cluster for its removal (replace child elements with parent element).
BEGRM	Time to begin check for refinement removal: LT.0: |BEGRM| represents a critical percent of NTOTRF below which the check for refinement removal should begin (0.0 < |BEGRM| < 1.0). See Remark 4.
ENDRM	Time to end the check for refinement removal.

Remarks:

1. **Number of Refinement Levels.** If NLVL=1, there is only one level of refinement: the elements in *ELEMENT_SOLID are the only ones to be replaced by clusters of 8 child elements. If NLVL > 1, there are several levels of refinement: not only the initial elements in *ELEMENT_SOLID are refined but also their child elements. If NLVL = 2 for example, the initial elements can be replaced by clusters of 64 child elements.

2. **Maximum Number of Elements to Refine.** NTOTRF defines the total number of elements to be refined. So for example NTOTRF=100 with NLVL=1 means that only 100 elements can be replaced by 800 finer elements (or 100 clusters of 8 child elements). For NLVL=2, these 800 elements can be replaced by 6400 finer elements.

3. **Number of Layers to Refine.** If an element is refined, it is possible to refine the neighbor elements as well. LAYRF defines the number of neighbor layers to refine. For example, LAYRF=2 for an element at the center of a block of 5 × 5 × 5 elements will refine these 125 elements.

4. **Onset of Refinement Removal.** If BEGRM < 0, the check for refinement removal is activated when the number of 8-element clusters for the refinement is below a limit defined by |BEGRM| × NTOTRF. If |BEGRM| = 0.1, it means that the check for refinement removal starts when 90% of the stock of clusters are used for the refinement.

5. **Maximum Refinement Removal.** MAXRM < 0 defines a total number of child clusters to remove for the whole run. If positive, MAXRM defines an upper limit for the number of child clusters to remove every NCYCRM cycles.

6. **The "elout" Database and Initial Refinement.** If only the 1st card is defined, the code for IELOUT is always activated. Since the refinement occurs during

the initialization, every refined element is replaced by its 8 children in the set defined for *DATABASE_ELOUT.

7. **The “elout” Database and Refinement at Run Time.** If there are more than 1 line, the code for IELOUT is activated if the flag is equal to 1. Since the refinement occurs during the run, the parent ids in the set defined for *DATABASE_-ELOUT are duplicated 8^{NLVL} times. The points of integration in the elout file are incremented to differentiate the child contributions to the database.

CONTROL_REMESHING_*{OPTION}

Available options include:

<BLANK>

EFG

Purpose: Provide control over the remeshing of solids which are meshed with the solid tetrahedron element type 13 and mesh-free solid types 41, 42. The element size for three-dimensional adaptivity can be set on the surface mesh of the solid part, and adaptivity can be activated based on the criteria of volume loss, mass increase, or minimum time step size. In addition, so-called interactive adaptivity triggers can be invoked using the EFG option.

There are two types of 3-D solid adaptivity affected by *CONTROL_REMESHING:

1. General tetrahedral adaptivity for which the EFG option of *CONTROL_-REMESHING may be invoked. See ADPOPT = 2 in *PART.

2. Axisymmetric adaptivity, sometimes called orbital adaptivity, in which remeshing is done with hexahedral and pentahedral elements. See AD-POPT = 3 in *PART. The EFG option of *CONTROL_REMESHING does not apply for this type of adaptivity.

Card 1	1	2	3	4	5	6	7	8
Variable	RMIN	RMAX	VF_LOSS	MFRAC	DT_MIN	ICURV	CID	SEGANG
Type	F	F	F	F	F	I	I	F
Default	none	none	1.0	0.0	0.	4	0	0.0

Additional card for EFG option.

Card 2	1	2	3	4	5	6	7	8
Variable	IVT	IAT	IAAT	IER	MM			
Type	I	I	I	I	I			
Default	1	0	0	0	0			

Second additional card for EFG option. This card is optional.

Card 3	1	2	3	4	5	6	7	8
Variable	IAT1	IAT2	IAT3					
Type	F	F	F					
Default	10^{20}	10^{20}	10^{20}					

VARIABLE	DESCRIPTION
RMIN	Minimum edge length for the surface mesh surrounding the parts which should be remeshed.
RMAX	Maximum edge length for the surface mesh surrounding the parts which should be remeshed.
VF_LOSS	Volume fraction loss required in a type 13 tetrahedral elements to trigger a remesh. In the type 13 solid elements, the pressures are computed at the nodal points; therefore, it is possible for volume to be conserved but for individual tetrahedrons to experience a significant volume loss or gain. The volume loss can lead to numerical problems. Recommended values for VF_LOSS in the range of 0.10 to 0.30 may be reasonable.
MFRAC	Mass ratio gain during mass scaling required for triggering a remesh. For a one percent increase in mass, set MFAC = 0.010. This variable applies to both to general three dimensional tetrahedral remeshing and to three dimensional axisymmetric remeshing.
DT_MIN	Time step size required for triggering a remesh. This option applies only to general three dimensional tetrahedral remeshing and is checked before mass scaling is applied and the time step size reset.
ICURV	Define number of element along the radius in the adaptivity. See remark 3.

VARIABLE DESCRIPTION

CID

Coordinate system ID for three dimensional axisymmetric remeshing. The z-axis in the defined coordinate system is the orbital axis, and has to be parallel to the global z-axis in the current axisymmetric remesher.

EQ.0: use global coordinate, and the global *z*-axis is the orbital axis (default)

SEGANG

For Axisymmetric 3-D remeshing: Angular element size (degrees).

For General (tet) 3-D remeshing: Critical angle specified in radians to preserve feature lines.

IVT

Internal variable transfer in adaptive EFG.

EQ.1: Moving Least square approximation with Kronecker-delta property (recommended in general case).

EQ.-1: Moving Least square approximation without Kronecker-delta property.

EQ.2: Partition of unity approximation with Kronecker-delta property.

EQ.-2: Partition of unity approximation without Kronecker-delta property.

EQ.-3: Finite element approximation.

IAT

Flag for interactive adaptivity.

EQ.0: No interactive adaptivity.

EQ.1: Interactive adaptivity combined with predefined adaptivity.

EQ.2: Purely interactive adaptivity. The time interval between two successive adaptive steps is bounded by ADPFREQ.

EQ.3: Purely interactive adaptivity.

IAAT

Interactive adaptivity adjustable tolerance.

EQ.0: The tolerance to trigger interactive adaptivity is not adjusted.

EQ.1: The tolerance is adjusted in run-time to avoid over-activation.

VARIABLE	DESCRIPTION
IER	Interactive adaptive remeshing with element erosion for metal cutting. EQ.1: The failed elements are eroded and the cutting surface is reconstructed before adaptive remeshing. The current implementation only supports SMP and IAT = 1, 2, 3.
MM	Interactive adaptive remeshing with monotonic resizing. EQ.1: The adaptive remeshing can not coarsen a mesh. The current implementation only supports IAT = 1, 2, 3 and IER = 0.
IAT1	Shear strain tolerance for interactive adaptivity. If the shear strain in any formulation 42 EFG tetrahedral element exceeds IAT1, remeshing is triggered. (0.1 ~ 0.5 is recommended).
IAT2	$L_{\mathrm{max}}/L_{\mathrm{min}}$ tolerance where L_{max} and L_{min} are the maximum and minimum edge lengths of any given formulation 42 EFG tetrahedral element. If this ratio in any element exceeds IAT2, remeshing is triggered. (RMAX/RMIN is recommended.)
IAT3	Volume change tolerance. If the normalized change in volume of any formulation 42 tetrahedral element, defined as $\lvert v_1 - v_0\rvert/\lvert v_0\rvert$ where v_1 is the current element volume and v_0 is the element volume immediately after the most recent remeshing, exceeds IAT3, remeshing is triggered. (0.5 is recommended.)

Remarks:

1. The value of RMIN and RMAX should be of the same order. The value of RMAX can be set to 2-5 times greater than RMIN.

2. When interactive adaptivity is invoked (IAT > 0), even if none of the tolerances IAT1, IAT2, and IAT3 for the three respective indicators (shear strain, edge length ratio, normalized volume change) are exceeded, remeshing will still be triggered if any of the three indicators over a single explicit time step changes by more than 50%, that is, if

$$\frac{\lvert[\mathrm{value}]_n - [\mathrm{value}]_{n-1}\rvert}{\lvert[\mathrm{value}]_{n-1}\rvert} > 0.5$$

where $[\text{value}]_n$ denotes value of indicator in n^{th} (current) time step and $[\text{value}]_{n-1}$ denotes value of indicator in previous time step . This condition is checked only if $[\text{value}]_{n-1}$ is nonzero.

3. ICURV represents a number of elements and applies only when ADPENE > 0 in *CONTROL_ADAPTIVE. The "desired element size" at each point on slave contact surface is computed based on the tooling radius of curvature (see the description of ADPENE in *CONTROL_ADAPTIVE), so that ICURV elements would be used to resolve a hypothetical 90 degree arc at the tooling radius of curvature. The value of ICURV is (internally) limited to be >=2 and <=12. The final adapted element size is adjusted as necessary to fall within the size range set forth by RMIN and RMAX.

*CONTROL_REQUIRE_REVISION

Purpose: To prevent the model from being run in old versions of LS-DYNA. This might be desirable due to known improvements in the program, required capability, etc.

Card 1	1	2	3	4	5	6	7	8
Variable	RELEASE	REVISION						
Type	C	I						
Default	none	none						

VARIABLE	DESCRIPTION
RELEASE	The release of code required. This should be a string such as "R6.1.0" or "R7.0"
REVISION	The minimum revision required. This corresponds to the "SVN Version" field in the d3hsp file.

Remarks:

1. Any number of lines can appear, indicating for example that a particular feature was introduced in different release branches at different times.

2. If the RELEASE field is left empty, then any executable whose development split from the main SVN trunk after the given REVISION will be allowed.

Example:

```
*CONTROL_REQUIRE_REVISION
R6.1    79315
R7.0    78310
        78304
```

This would prevent execution by any R6.1 executable before r79315, any R7.0 before r78310, and all other executables whose development split from the main trunk before r78304. Note that no versions of R6.0, R6.0.0, or R6.1.0 are allowed: R6.1 does NOT imply R6.1.0, no matter what the revision of R6.1.0 – R6.1.0 would have to be explicitly listed. Similarly, R7.0.0 would not be allowed because it is not listed, and it split from the trunk in r76398. Any future R8.X executable would be allowed, since it will have split from the trunk after r78304.

*CONTROL_RIGID

Purpose: Special control options related to rigid bodies and to linearized flexible bodies, see *PART_MODES.

Card 1	1	2	3	4	5	6	7	8
Variable	LMF	JNTF	ORTHMD	PARTM	SPARSE	METALF	PLOTEL	RBSMS
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

Remaining cards are optional.†

Card 2	1	2	3	4	5	6	7	8
Variable	NORBIC							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
LMF	Switch the explicit rigid body joint treatment to an implicit formulation which uses Lagrange multipliers to impose prescribed kinematic boundary conditions and joint constraints. There is a slight cost overhead due to the assembly of sparse matrix equations which are solved using standard procedures for nonlinear problems in rigid multi-body dynamics. Lagrange multiplier flag: EQ.0: explicit penalty formulation EQ.1: implicit formulation with Lagrange multipliers

VARIABLE **DESCRIPTION**

JNTF

Generalized joint stiffness formulation; see Remark 1 below:

EQ.0: incremental update

EQ.1: total formulation (exact)

EQ.2: total formulation intended for implicit analysis

ORTHMD

Orthogonalize modes with respect to each other:

EQ.0: true

EQ.1: false, the modes are already orthogonalized

PARTM

Use global mass matrix to determine part mass distribution. This mass matrix may contain mass from other parts that share nodes. See Remark 2 below.

EQ.0: true

EQ.1: false

SPARSE

Use sparse matrix multiply subroutines for the modal stiffness and damping matrices. See Remark 3.

EQ.0: false, do full matrix multiplies (frequently faster).

EQ.1: true

MATELF

Metal forming option, which should not be used for crash and other applications involving rigid bodies. Use fast update of rigid body nodes. If this option is active the rotational motion of all rigid bodies should be suppressed.

EQ.0: full treatment is used

EQ.1: fast update for metal forming applications

PLOTEL

Automatic generation of *ELEMENT_PLOTEL for *CONSTRAINED_NODAL_RIGID_BODY.

EQ.0: no generation

EQ.1: one part is generated for all nodal rigid bodies with the PID set to 1000000.

EQ.2: one part is generated for each nodal rigid body in the problem with a part ID of 1000000 + PID, where PID is the nodal rigid body ID.

VARIABLE	DESCRIPTION
RBSMS	Flag to apply consistent treatment of rigid bodies in selective and conventional mass scaling, Remark 4. EQ.0: Off EQ.1: On
NORBIC	Circumvent rigid body inertia check, see Remark 5. EQ.0: Off EQ.1: On

Remarks:

1. **JNTF.** The default behavior is for the relative angles between the two coordinate systems to be done incrementally. This is an approximation, in contrast to the total formulation where the angular offsets are computed exactly. The disadvantage of the latter approach is that a singularity exists when an offset angle equals 180 degrees. In most applications, the stop angles exclude this possibility and JNTF=1 should not cause a problem. JNTF=2 is implemented with smooth response and especially intended for implicit analysis.

2. **PARTM.** If the determination of the normal modes included the mass from both connected bodies and discrete masses, or if there are no connected bodies, then the default is preferred. When the mass of a given part ID is computed, the resulting mass vector includes the mass of all rigid bodies that are merged to the given part ID, but does not included discrete masses. See the keyword: *CONSTRAINED_RIGID_BODIES. A lumped mass matrix is always assumed.

3. **SPARSE.** Sparse matrix multipliers save a substantial number of operations if the matrix is truly sparse. However, the overhead will slow the multipliers for densely populated matrices.

4. **RBSMS.** In selective mass scaling, rigid bodies connected to deformable elements can result in significant addition of inertia due missing terms in the SMS mass matrix. This problem has been observed in automotive applications where spotwelds are modeled using constrained nodal rigid bodies. By applying consistent rigid body treatment significant improvement in accuracy and robustness are observed at the expense of increased CPU intensity. This flag also applies to conventional mass scaling as it has been observed that inconsistencies for various reasons may result in unstable solution schemes even for this case.

5. **NORBIC.** During initialization, the determinant of the rigid body inertia tensor is checked. If it falls below a tolerance value of 10^{-30}, LS-DYNA issues an error message and the calculation stops. In some rare cases (e.g. with an adverse system of units), such tiny values would still be valid. In this case, NORBIC should be set to 1 to circumvent the implied inertia check.

*CONTROL_SHELL

Purpose: Provide controls for computing shell response.

Card 1	1	2	3	4	5	6	7	8
Variable	WRPANG	ESORT	IRNXX	ISTUPD	THEORY	BWC	MITER	PROJ
Type	F	I	I	I	I	I	I	I
Default	20.	0	-1	0	2	2	1	0

Remaining cards are optional.†

Card 2	1	2	3	4	5	6	7	8
Variable	ROTASCL	INTGRD	LAMSHT	CSTYP6	THSHEL			
Type	F	I	I	I	I			
Default	1.	0	0	1	0			

Card 3	1	2	3	4	5	6	7	8
Variable	PSTUPD	SIDT4TU	CNTCO	ITSFLG	IRQUAD	W-MODE	STRETCH	ICRQ
Type	I	I	I	I	I	F	F	I
Default	0	0	0	0	0	inactive	inactive	0

Card 4	1	2	3	4	5	6	7	8
Variable	NFAIL1	NFAIL4	PSNFAIL	KEEPCS	DELFR	DRCPSID	DRCPRM	INTPERR
Type	I	I	I	I	I	I	F	
Default	inactive	inactive	0	0	0	0	1.0	

VARIABLE **DESCRIPTION**

WRPANG Shell element warpage angle in degrees. If a warpage greater than this angle is found, a warning message is printed. Default is 20 degrees.

ESORT Sorting of triangular shell elements to automatically switch degenerate quadrilateral shell formulations to more suitable triangular shell formulations.

EQ.0: Do not sort (default).

EQ.1: Sort (switch to C0 triangular shell formulation 4, or if a quadratic shell, switch to shell formulation 24, or if a shell formulation with thickness stretch, switch to shell formulation 27).

EQ.2: Sort (switch to DKT triangular shell formulation 17, or if a quadratic shell, switch to shell formulation 24). The DKT formulation will be unstable if used to model an uncommonly thick, triangular shell.

IRNXX Shell normal update option. This option affects the Hughes-Liu, Belytschko-Wong-Chiang, and the Belytschko-Tsay shell formulations (including fully integrated shells -16 and 16). The latter is affected if and only if the warping stiffness option is active, i.e., BWC = 1.

EQ.-2: unique nodal fibers which are incrementally updated based on the nodal rotation at the location of the fiber,

EQ.-1: recomputed fiber directions each cycle,

EQ.0: default set to -1,

EQ.1: compute on restarts,

EQ.n: compute every n cycles (Hughes-Liu shells only).

VARIABLE	DESCRIPTION
ISTUPD	Shell thickness change option for deformable shells. The parameter, PSTUPD, on the second optional card allows this option to be applied by part ID. For crash analysis, neglecting the elastic component of the strains, ISTUPD = 4, may improve energy conservation and stability. EQ.0: no thickness change. EQ.1: membrane straining causes thickness change in 3 and 4 node shell elements. This option is very important in sheet metal forming or whenever membrane stretching is important. EQ.2: membrane straining causes thickness change in 8 node thick shell elements, types 1 and 2. This option is not recommended for implicit or explicit solutions which use the fully integrated type 2 elements. Types 3 and 5 thick shells are continuum based and thickness changes are always considered. EQ.3: options 1 and 2 apply. EQ.4: option 1 applies, but the elastic strains are neglected for the thickness update. This option only applies to shells (not thick shells) and the most common elastic-plastic materials for which the elastic response is isotropic. See SIDT4TU for selective application of this option.
THEORY	Default shell formulation. For a complete list of shell formulations, refer to *SECTION_SHELL. For remarks on overriding this default and how THEORY may affect contact behavior, see Remark 2. EQ.1: Hughes-Liu EQ.2: Belytschko-Tsay (default) EQ.3: BCIZ triangular shell (not recommended) EQ.4: C0 triangular shell EQ.5: Belytschko-Tsay membrane EQ.6: S/R Hughes Liu EQ.7: S/R co-rotational Hughes Liu EQ.8: Belytschko-Leviathan shell EQ.9: fully integrated Belytschko-Tsay membrane

VARIABLE **DESCRIPTION**

EQ.10: Belytschko-Wong-Chiang

EQ.11: Fast (co-rotational) Hughes-Liu

EQ.12: Plane stress (*x*-*y* plane)

EQ.13: Plane strain (*x*-*y* plane)

EQ.14: Axisymmetric solid (*y*-axis of symmetry) – area weighted. See Remark 5

EQ.15: Axisymmetric solid (*y*-axis of symmetry) – volume weighted. See Remark 5

EQ.16: Fully integrated shell element (very fast)

EQ.17: Discrete Kirchhoff triangular shell (DKT)

EQ.18: Discrete Kirchhoff linear shell either quadrilateral or Triangular with 6DOF per node

EQ.20: C0 linear shell element with 6 DOF per node

EQ.21: C0 linear shell element with 5 DOF per node with the Pian-Sumihara membrane hybrid quadrilateral membrane

EQ.25: Belytschko-Tsay shell with thickness stretch

EQ.26: Fully integrated shell with thickness stretch

EQ.27: C0 triangular shell with thickness stretch

BWC

Warping stiffness for Belytschko-Tsay shells:

EQ.1: Belytschko-Wong-Chiang warping stiffness added.

EQ.2: Belytschko-Tsay (default).

MITER

Plane stress plasticity option (applies to materials 3, 18, 19, and 24):

EQ.1: iterative plasticity with 3 secant iterations (default),

EQ.2: full iterative plasticity,

EQ.3: radial return noniterative plasticity. May lead to false results and has to be used with great care.

PROJ

Projection method for the warping stiffness in the Belytschko-Tsay shell (the BWC option above) and the Belytschko-Wong-Chiang elements (see Remark 1 below). This parameter applies to explicit calculations since the full projection method is always used if the solution is implicit and this input parameter is

VARIABLE **DESCRIPTION**

ignored.

EQ.0: drill projection,

EQ.1: full projection.

ROTASCL

Define a scale factor for the rotary shell mass. This option is not for general use. The rotary inertia for shells is automatically scaled to permit a larger time step size. A scale factor other than the default, i.e., unity, is not recommended.

INTGRD

Default through thickness numerical integration rule for shells and thick shells. If more than 10 integration points are requested, a trapezoidal rule is used unless a user defined rule is specified.

EQ.0: Gauss integration:If 1-10 integration points are specified, the default rule is Gauss integration.

EQ.1: Lobatto integration: If 3-10 integration points are specified, the default rule is Lobatto. For 2 point integration, the Lobatto rule is very inaccurate, so Gauss integration is used instead. Lobatto integration has an advantage in that the inner and outer integration points are on the shell surfaces.

LAMSHT

Laminated shell theory flag. Except for those using the Green-Lagrange strain tensor, laminated shell theory is available for all thin shell and thick shell materials. It is activated when LAMSHT = 3, 4, or 5 and by using *PART_COMPOSITE or *INTEGRATION_SHELL to define the integration rule. See Remark 6.

EQ.0: do not update shear corrections,

EQ.1: activate laminated shell theory,

EQ.3: activate laminated thin shells,

EQ.4: activate laminated shell theory for thick shells,

EQ.5: activate laminated shell theory for thin and thick shells.

CSTYP6

Coordinate system for the type 6 shell element. The default system computes a unique local system at each in plane point. The uniform local system computes just one system used throughout the shell element. This involves fewer calculations and is therefore more efficient. The change of systems has a slight effect on results; therefore, the older, less efficient method is the

VARIABLE	DESCRIPTION
	default. EQ.1: variable local coordinate system (default), EQ.2: uniform local system.
THSHEL	Thermal shell option (applies only to thermal and coupled structural thermal analyses). See parameter THERM on DATABASE_EXTENT_BINARY keyword. EQ.0: No temperature gradient is considered through the shell thickness (default). EQ.1: A temperature gradient is calculated through the shell thickness.
PSTUPD	|PSTUPD| is the optional shell part set ID specifying which part ID's have or do not have their thickness updated according to ISTUPD. The shell thickness update as specified by ISTUPD by default applies to all shell elements in the mesh. LT.0: these shell parts are excluded from the shell thickness update EQ.0: all deformable shells have their thickness updated GT.0: these shell parts are included in the shell thickness update
SIDT4TU	Shell part set ID for parts which use the type 4 thickness update where elastic strains are ignored. The shell parts in part set SIDT4TU must also be included in the part set defined by PSTUPD. SIDT4TU has no effect unless ISTUPD is set to 1 or 3.
CNTCO	Flag affecting location of contact surfaces for shells when NLOC is nonzero in *SECTION_SHELL or in *PART_COMPOSITE, or when OFFSET is specified using *ELEMENT_SHELL_OFFSET. CNTCO is not supported for the slave side of NODES_TO_SURFACE type contacts, neither has it any effect for Mortar contacts. For Mortar contacts NLOC of OFFSET completely determines the location of the contact surfaces, as if CNTCO = 1 would be set. EQ.0: NLOC and OFFSET have no effect on location of shell contact surfaces. EQ.1: Contact reference plane (see comments below) coincides with shell reference surface. EQ.2: Contact reference plane (see comments below) is affected

VARIABLE **DESCRIPTION**

by contact thickness. This is typically not physical.

For automatic contact types, the shell contact surfaces are always, regardless of CNTCO, offset from a contact reference plane by half a contact thickness. Contact thickness is taken as the shell thickness by default but can be overridden, for example, with input on Card 3 of *CONTACT.

The parameter CNTCO affects how the location of the contact reference plane is determined. When CNTCO = 0, the contact reference plane coincides with the plane of the shell nodes. Whereas when CNTCO = 1, the contact reference plane coincides with the shell reference surface as determined by NLOC or by OFFSET. For CNTCO = 2, the contact reference plane is offset from the plane of the nodes by

$$-\frac{\mathrm{NLOC}}{2}\times \text{contact thickness}$$

or by

$$\mathrm{OFFSET}\times\left(\frac{\text{contact thickness}}{\text{shell thickness}}\right)$$

whichever applies.

ITSFLG

Flag to activate/deactivate initial transverse shear stresses (local shell stress components σ_{yz} and σ_{zx}) from *INITIAL_STRESS_SHELL:

EQ.0: keep transverse shear stresses

EQ.1: set transverse shear stresses to zero

IRQUAD

In plane integration rule for the 8-node quadratic shell element (shell formulation 23):

EQ.2: 2 × 2 Gauss quadrature (default),

EQ.3: 3 × 3 Gauss quadrature.

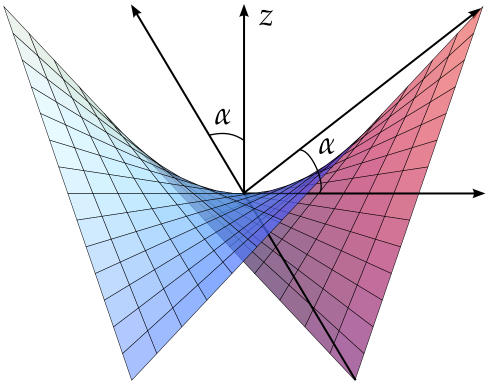


Figure 12-86. Illustration of an element in a *W-Mode*. One pair of opposite corners go up, and the other pair goes down. The angle, α, is formed by the plane of the flat element and by the vector connecting the center to the corner. See Remark 4.

VARIABLE	DESCRIPTION
W-MODE	W-Mode amplitude for element deletion, specified in degrees. See Figure 12-86 and Remark 4 for the definition of the angle.
STRETCH	Stretch ratio of element diagonals for element deletion. This option is activated if and only if either NFAIL1 or NFAIL4 are nonzero and STRETCH > 0.0.
ICRQ	Continuous treatment across element edges for some specified result quantities. See Remark 7. EQ.0: not active EQ.1: thickness and plastic strain
NFAIL1	Flag to check for highly distorted under-integrated shell elements, print a message, and delete the element or terminate. Generally, this flag is not needed for one point elements that do not use the warping stiffness. A distorted element is one where a negative Jacobian exist within the domain of the shell, not just at integration points. The checks are made away from the CPU requirements for one point elements. If nonzero, NFAIL1 can be changed in a restart. EQ.1: print message and delete element. EQ.2: print message, write d3dump file, and terminate GT.2: print message and delete element. When NFAIL1 elements are deleted then write d3dump file and terminate. These NFAIL1 failed elements also include all shell elements that failed for other reasons than distortion.

VARIABLE **DESCRIPTION**

Before the d3dump file is written, NFAIL1 is doubled, so the run can immediately be continued if desired.

NFAIL4

Flag to check for highly distorted fully-integrated shell elements, print a message and delete the element or terminate. Generally, this flag is recommended. A distorted element is one where a negative Jacobian exist within the domain of the shell, not just at integration points. The checks are made away from the integration points to enable the bad elements to be deleted before an instability leading to an error termination occurs. If nonzero, NFAIL4 can be changed in a restart.

EQ.1: print message and delete element.

EQ.2: print message, write d3dump file, and terminate

GT.2: print message and delete element. When NFAIL4 elements are deleted then write d3dump file and terminate. These NFAIL4 failed elements also include all shell elements that failed for other reasons than distortion. Before the d3dump file is written, NFAIL4 is doubled, so the run can immediately be continued if desired.

PSNFAIL

Optional shell part set ID specifying which part ID's are checked by the NFAIL1, NFAIL4, and W-MODE options. If zero, all shell part ID's are included.

KEEPCS

Flag to keep the contact segments of failed shell elements in the calculation. The contact segments of the failed shells remain active until a node shared by the segments has no active shells attached. Only then are the segments deleted.

EQ.0: Inactive

EQ.1: Active

DELFR

Flag to delete shell elements whose neighboring shell elements have failed; consequently, the shell is detached from the structure and moving freely in space. This condition is checked if NFAIL1 or NFAIL4 are nonzero.

EQ.0: Inactive

EQ.1: Isolated elements are deleted.

EQ.2: Isolated quadrilateral elements and triangular elements connected by only one node are deleted.

VARIABLE **DESCRIPTION**

EQ.3: Elements that are either isolated or connected by only one node are deleted.

DRCPSID Part set ID for drilling rotation constraint method (see Remark 3 below).

DRCPRM Drilling rotation constraint parameter (default = 1.0).

INTPERR Flag for behavior in case of unwanted interpolation/extrapolation of initial stresses from *INITIAL_STRESS_SHELL.

EQ.0: Only warning is written, calculation continues (default).

EQ.1: Error exit, calculation stops.

Remarks:

1. **Drill versus Full Projections for Warping Stiffness.** The drill projection is used in the addition of warping stiffness to the Belytschko-Tsay and the Belytschko-Wong-Chiang shell elements. This projection generally works well and is very efficient, but to quote Belytschko and Leviathan:

> "The shortcoming of the drill projection is that even elements that are invariant to rigid body rotation will strain under rigid body rotation if the drill projection is applied. On one hand, the excessive flexibility rendered by the 1-point quadrature shell element is corrected by the drill projection, but on the other hand the element becomes too stiff due to loss of the rigid body rotation invariance under the same drill projection".

They later went on to add in the conclusions:

> "The projection of only the drill rotations is very efficient and hardly increases the computation time, so it is recommended for most cases. However, it should be noted that the drill projection can result in a loss of invariance to rigid body motion when the elements are highly warped. For moderately warped configurations the drill projection appears quite accurate".

In crashworthiness and impact analysis, elements that have little or no warpage in the reference configuration can become highly warped in the deformed configuration and may affect rigid body rotations if the drill projection is used, i.e., DO NOT USE THE DRILL PROJECTION. Of course it is difficult to define what is meant by "moderately warped". The full projection circumvents these problems but at a significant cost. The cost increase of the drill projection versus no projection as reported by Belytschko and Leviathan is 12 percent and by timings in LS-DYNA, 7 percent, but for the full projection they report a 110 percent increase and in LS-DYNA an increase closer to 50 percent is observed.

In Version 940 of LS-DYNA the drill projection was used exclusively, but in one problem the lack of invariance was observed; consequently, the drill projection was replaced in the Belytschko-Leviathan shell with the full projection and the full projection is now optional for the warping stiffness in the Belytschko-Tsay and Belytschko-Wong-Chiang elements. Starting with version 950 the Belytschko-Leviathan shell, which now uses the full projection, is somewhat slower than in previous versions. In general, in light of these problems, the drill projection cannot be recommended. For implicit calculations, the full projection method is used in the development of the stiffness matrix.

2. **THEORY, ELFORM, and Contact with Tapered Shells.** All shell parts need not share the same element formulation. A nonzero value of ELFORM, given either in *SECTION_SHELL or *PART_COMPOSITE, overrides the element formulation specified by THEORY in *CONTROL_SHELL.

   When using MPP, THEORY = 1 in *CONTROL_SHELL has special meaning when dealing with non-uniform-thickness shells, that is, it serves to set the nodal contact thickness equal to the average of the nodal thicknesses from the shells sharing that node. Thus when a contact surface is comprised of non-uniform-thickness shells, THEORY = 1 is recommended and the user still has the option of setting the actual shell theory using ELFORM in *SECTION_SHELL.

3. **Drilling Rotation Constraint Method.** The drilling rotation constraint method which is used by default in implicit calculations (see parameter DRCM=4 on *CONTROL_IMPLICIT_SOLVER) can be used in explicit calculations as well by defining an appropriate part set DRCPSID. This might be helpful in situations where single constraints (e.g. spotwelds) are connected to flat shell element topologies. The additional drill force can by scaled with DRCPRM (default value is 1.0), where a moderate value should be chosen to avoid excessive stiffening of the structure. A speed penalty of max. 15 % may be observed with this option.

4. **W-Mode Failure Criterion.** The w-mode failure criteria depends on the magnitude of the w-mode, w, compared to the approximate side-length ℓ. The magnitude, w, is defined as

   $$w = \frac{1}{4}[(\mathbf{x}_1 - \mathbf{x}_2) + (\mathbf{x}_3 - \mathbf{x}_4)] \cdot \mathbf{n}$$

   where $\mathbf{x}_i$ is the position vector for node i, and $\mathbf{n}$ is the element normal vector evaluated at the centroid. The element normal is the unit vector obtained from the cross product of the diagonal vectors $\mathbf{a}$ and $\mathbf{b}$ as,

$$\mathbf{a} = \mathbf{x}_3 - \mathbf{x}_1$$

$$\mathbf{b} = \mathbf{x}_4 - \mathbf{x}_2$$

$$\mathbf{n} = \frac{\mathbf{a} \times \mathbf{b}}{\|\mathbf{a} \times \mathbf{b}\|}.$$

The failure criteria depends on the ratio of w to ℓ, where ℓ is defined as,

$$\ell = \frac{1}{2} \underbrace{\left[\sqrt{2} \underbrace{\sqrt{\frac{1}{2} \|\mathbf{a} \times \mathbf{b}\|}}_{\sim\sqrt{\text{area}}} \right]}_{\sim\text{diagonal length}}$$

such that the element is deleted when

$$\frac{|w|}{\ell} \geq \tan(\text{WMODE}).$$

The angle α in the figure may be identified as,

$$\alpha = \arctan\left(\frac{|w|}{\ell}\right).$$

5. **2D Axisymmetric Solid Elements.** The 2D axisymmetric solid elements come in two types: area weighted (type 14) and volume weighted (type 15).

   a) High explosive applications work best with the area weighted approach and structural applications work best with the volume weighted approach. The volume weighted approach can lead to problems along the axis of symmetry under very large deformations. Often the symmetry condition is not obeyed, and the elements will kink along the axis.

   b) The volume weighted approach must be used if 2D shell elements are used in the mesh. Type 14 and 15 elements cannot be mixed in the same calculation**.**

6. **Lamination Theory.** Lamination theory should be activated when the assumption that shear strain through the shell is uniform and constant becomes violated. Unless this correction is applied, the stiffness of the shell can be grossly incorrect if there are drastic differences in the elastic constants from ply to ply, especially for sandwich type shells. Generally, without this correction the results are too stiff. For the discrete Kirchhoff shell elements, which do not consider transverse shear, this option is ignored. For thin shells of material types, *MAT_COMPOSITE_DAMAGE, *MAT_ENHANCED_COMPOSITE_DAMAGE, and *MAT_GENERAL_VISCOELASTIC, laminated shell theory may also be done by stiffness correction by setting LAMSHT=1.

7. **Continuous Result Quantities**. A nodal averaging technique is used to achieve continuity for some quantities across element edges. Applying this approach to the thickness field and plastic strains (ICRQ=1) can reduce alternating weak localizations sometimes observed in metal forming applications when shell elements get stretch-bended over small radii. This option currently works with shell element types 2, 4, and 16. A maximum number of 9 through thickness integration points is allowed for this method. A speed penalty of max. 15 % may be observed with this option.

*CONTROL_SOLID

Purpose: Provide controls for solid element response.

Card 1	1	2	3	4	5	6	7	8
Variable	ESORT	FMATRX	NIPTETS	SWLOCL	PSFAIL	T10JTOL	ICOH	TET13K
Type	I	I	I	I	I	F	I	I
Default	0	0	4	1	0	0.	0	0

This card is optional.

Card 2	1	2	3	4	5	6	7	8	9	10
Variable	PM1	PM2	PM3	PM4	PM5	PM6	PM7	PM8	PM9	PM10
Type	I	I	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none	none	none

VARIABLE **DESCRIPTION**

ESORT

Automatic sorting of tetrahedral and pentahedral elements to avoid use of degenerate formulations for these shapes. See *SECTION_SOLID.

EQ.0: no sorting (default)

EQ.1: sort tetrahedron to type 10; pentahedron to type 15; cohesive pentahedron types 19 and 20 to types 21 and 22, respectively.

EQ.2: sort tetrahedron to type 10; 1-point integrated pentahedron to type 115; fully integrated pentahedron to type 15; cohesive pentahedron types 19 and 20 to types 21 and 22, respectively.

EQ.3: same as EQ.1 but also print switched elements in messag file

EQ.4: same as EQ.2 but also print switched elements in messag file

VARIABLE	DESCRIPTION
FMATRX	Default method used in the calculation of the deformation gradient matrix. EQ.1: Update incrementally in time. This is the default for explicit. EQ.2: Directly compute F: This is the default for implicit and implicit/explicit switching.
NIPTETS	Number of integration points used in the quadratic tetrahedron elements. Either 4 or 5 can be specified. This option applies to the types 4, 16, and 17 tetrahedron elements.
SWLOCL	Output option for stresses in solid elements used as spot welds with material *MAT_SPOTWELD. Affects elout and d3plot/d3part/etc. EQ.1: Stresses in global coordinate system (default), EQ.2: Stresses in element coordinate system.
PSFAIL	A nonzero PSFAIL has the same effect as setting ERODE = 1 in *CONTROL_TIMESTEP except that solid element erosion due to negative volume is limited to *only* the solid elements in part set PSFAIL. In other words, when PSFAIL is nonzero, the time-step-based criterion for erosion (TSMIN) applies to all solid elements (except formulations 11 and 12) while the negative volume criterion for erosion applies only to solids in part set PSFAIL.
T10JTOL	Tolerance for jacobian in 4-point 10-noded quadratic tetrahedra (type 16). If the quotient between the minimum and maximum jacobian value falls below this tolerance, a warning message is issued in the messag file. This is useful for tracking badly shaped elements in implicit analysis that deteriorates convergence, a value of 1.0 indicates a perfectly shaped element.

VARIABLE **DESCRIPTION**

ICOH

Global flag for cohesive element options, interpreted digit-wise as follows:

$$\text{ICOH} = [LK] = K + 10 \times L$$

K.EQ.1: Solid elements having ELFORM = 19-22 will be eroded when neighboring shell or solid elements fail. Only works for nodewise connected parts, not tied contacts.

K.EQ.0: No cohesive element deletion due to neighbor failure.

L.EQ.0: Default critical time step estimate.

L.EQ.1: Most conservative (smallest) critical time step estimate.

L.EQ.2: Intermediate critical time step estimate.

TET13K

Set to 1 to invoke a consistent tangent stiffness matrix for the pressure averaged tetrahedron (type 13). This feature is available only for the implicit integrator *and it is not supported in the MPP/MPI version*. This element type averages the volumetric strain over adjacent elements to alleviate volumetric locking, therefore, the corresponding material tangent stiffness should be treated accordingly. In contrast to a hexahedral mesh where a node usually connects to fewer than 8 elements, tetrahedral meshes offer no such regularity. Consequently, for nonlinear implicit analysis matrix assembly is computationally expensive and this option is recommended only for linear or eigenvalue analysis to exploit the stiffness characteristics of the type 13 tetrahedron.

PM1 – PM10

Components of a permutation vector for nodes that define the 10-node tetrahedron. The nodal numbering of 10-node tetrahedron elements is somewhat arbitrary. The permutation vector allows other numbering schemes to be used. Unless defined, this permutation vector is not used. PM1 – PM10 are unique numbers between 1 to 10 inclusive that reorders the input node ID's for a 10-node tetrahedron into the order used by LS-DYNA.

*CONTROL_SOLUTION

Purpose: To specify the analysis solution procedure if thermal only or coupled thermal analysis is performed. Other solutions parameters including the vector length and NaN (not a number) checking can be set.

Card 1	1	2	3	4	5	6	7	8
Variable	SOLN	NLQ	ISNAN	LCINT	LCACC			
Type	I	I	I	I	I			
Default	0	0	0	100	0			

VARIABLE	DESCRIPTION
SOLN	Analysis solution procedure: EQ.0: Structural analysis only, EQ.1: Thermal analysis only, EQ.2: Coupled structural thermal analysis.
NLQ	Define the vector length used in solution. This value must not exceed the vector length of the system which varies based on the machine manufacturer. The default vector length is printed at termination in the messag file.
ISNAN	Flag to check for a NaN in the force and moment arrays after the assembly of these arrays is completed. This option can be useful for debugging purposes. A cost overhead of approximately 2% is incurred when this option is active. EQ.0: No checking, EQ.1: Checking is active.
LCINT	Number of equally spaced points used in curve (*DEFINE_CURVE) rediscretization. Curve rediscretization applies only to curves used in material models. Curves defining loads, motion, etc. are not rediscretized.

VARIABLE	DESCRIPTION
LCACC	Flag to truncate curves to 6 significant figures for single precision and 13 significant figures for double precision. The truncation is done after applying the offset and scale factors specified in *DEFINE_CURVE. Truncation is intended to prevent curve values from deviating from the input value, e.g., 0.7 being stored as 0.69999999. This small deviation was seen to have an adverse effect in a particular analysis using *MAT_083. In general, curve truncation is not necessary and is unlikely to have any effect on results. EQ.0: No truncation. NE.0: Truncate.

*CONTROL_SPH

Purpose: Provide controls relating to SPH (Smooth Particle Hydrodynamics).

Card 1	1	2	3	4	5	6	7	8
Variable	NCBS	BOXID	DT	IDIM	MEMORY	FORM	START	MAXV
Type	I	I	F	I	I	I	F	F
Default	1	0	10^{20}	none	150	0	0.0	10^{15}

Optional Card.

Card 2	1	2	3	4	5	6	7	8
Variable	CONT	DERIV	INI	ISHOW	IEROD	ICONT	IAVIS	ISYMP
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	100

VARIABLE	DESCRIPTION
NCBS	Number of time steps between particle sorting.
BOXID	SPH approximations are computed inside a specified BOX. When a particle has gone outside the BOX, it is deactivated. This will save computational time by eliminating particles that no longer interact with the structure.
DT	Death time. Determines when the SPH calculations are stopped.
IDIM	Space dimension for SPH particles: EQ.3: for 3D problems EQ.2: for 2D plane strain problems EQ.-2: for 2D axisymmetric problems
MEMORY	Defines the initial number of neighbors per particle (see Remark 1 below).

VARIABLE **DESCRIPTION**

FORM

Particle approximation theory (Remark 2):

EQ.0: default formulation

EQ.1: renormalization approximation

EQ.2: symmetric formulation

EQ.3: symmetric renormalized approximation

EQ.4: tensor formulation

EQ.5: fluid particle approximation

EQ.6: fluid particle with renormalization approximation

EQ.7: total Lagrangian formulation

EQ.8: total Lagrangian formulation with renormalization

EQ.15: enhanced fluid formulation

EQ.16: enhanced fluid formulation with renormalization

START

Start time for particle approximation. Particle approximations will be computed when time of the analysis has reached the value defined in START.

MAXV

Maximum value for velocity for the SPH particles. Particles with a velocity greater than MAXV are deactivated. A negative MAXV will turn off the velocity checking.

CONT

Defines the computation of the particle approximation between different SPH parts:

EQ.0: Particle approximation is defined (default)

EQ.1: Particle approximation is not computed. Different SPH materials will not interact with each other and penetration is allowed unless *DEFINE_SPH_TO_SPH_COUPLING is defined. Combined with *SECTION_SPH_INTERACTION, a partial interaction between SPH parts through normal interpolation method and partially interact through the contact option can be realized. See *SECTION_SPH_INTERACTION.

DERIV

Time integration type for the smoothing length:

EQ.0: $\frac{d}{dt}[h(t)] = \frac{1}{d}h(t)\nabla \cdot \mathbf{v}$, (default),

EQ.1: $\frac{d}{dt}[h(t)] = \frac{1}{d}h(t)(\nabla \cdot \mathbf{v})^{1/3}$

VARIABLE **DESCRIPTION**

INI

Computation of the smoothing length during the initialization:

EQ.0: Bucket sort based algorithm (default, very fast).

EQ.1: Global computation on all the particles of the model.

EQ.2: Based on the mass of the SPH particle.

ISHOW

Display option for deactivated SPH particles:

EQ.0: No distinction in active SPH particles and deactivated SPH particles when viewing in LS-PrePost.

EQ.1: Deactivated SPH particles are displayed only as points and active SPH particles are displayed as spheres when Setting → SPH → Style is set to “smooth” in LS-PrePost.

IEROD

Deactivation control for SPH particles:

EQ.0: Particles remain active.

EQ.1: SPH particles are partially deactivated and stress states are set to 0 when erosion criteria are satisfied. See Remark 3.

EQ.2: SPH particles are totally deactivated and stress states are set to 0 when erosion criteria are satisfied. See Remark 3.

ICONT

Controls contact behavior for deactivated SPH particles:

EQ.0: Any contact defined for SPH remains active for deactivated particles.

EQ.1: Contact is inactive for deactivated particles.

IAVIS

Defines artificial viscosity formulation for SPH elements (Remark 4):

EQ.0: Monaghan type artificial viscosity formulation is used.

EQ.1: Standard type artificial viscosity formulation from solid element is used (this option is not supported in SPH 2D and 2D axisymmetric elements).

ISYMP

Defines the percentage of original SPH particles used for memory allocation of SPH symmetric planes ghost nodes generation process (default is 100%). Recommended for large SPH particles models (value range 10~20) to control the memory allocation for SPH ghost particles with *BOUNDARY_SPH_SYMMETRY_-PLANE keyword.

Remark:

1. **Memory.** MEMORY is used to determine the initial memory allocation for the SPH arrays. Its value can be positive or negative. If MEMORY is positive, memory allocation is dynamic such that the number of neighboring particles is initially equal to MEMORY but that number is subsequently allowed to exceed MEMORY as the solution progresses. If MEMORY is negative, memory allocation is static and |MEMORY| is the maximum allowed number of neighboring particles for each particle throughout the entire solution. Using this static memory option can avoid memory allocation problems.

2. **Form.** Some guidelines for selecting form variable: for most solid structure applications, form = 1 is recommended for more accurate results around the boundary area; for fluid or fluid-like material applications, form = 15, 16 with fluid formulation are recommended (form=16 usually has better accuracy but requires more CPU time); form = 2, 3 are not recommended for any case; all SPH formulations with Eulerian kernel (form = 0 to 6, 15 and 16) can be used for large or extreme large deformation applications but will have tensile instability issue; all SPH formulations with Lagrangian kernel (form = 7,8) can be used to avoid tensile instability issue but they can not endure very large deformation, user has to be careful to pick up the right one based on the applications. Only formulations 0, 1, 15 and 16 are implemented for 2D axisymmetric problems (dim=-2). Also note that forms 15 and 16 include a smoothing of the pressure field, and are therefore not recommended for materials with failure or problems with important strain localization.

3. **Erosion.** The erosion criteria, which triggers particle deactivation when IEROD=1 or 2, may come from either the material model with *MAT_ADD_-EROSION or from the ERODE parameter in *CONTROL_TIMESTEP. For IEROD=1, SPH particles are partially deactivated (i.e. the stress states of the deactivated SPH particles will be set to 0, but those particles still remain in the domain integration for more stable results); For IEROD=2, SPH particles are totally deactivated: stress states will be set to 0 and the deactivated particles no more remain in the domain integration. Deactivated particles can be distinguished from active particles by setting ISHOW=1. To disable contact for deactivated particles, set ICONT=1.

4. **Artificial Viscosity.** The artificial viscosity for standard solid elements, which is active when AVIS=1, is given by:

$$
\begin{aligned}
q &= \rho l \left(Q_1 l \dot{\varepsilon}_{kk}^2 - Q_2 a \dot{\varepsilon}_{kk} \right) && \dot{\varepsilon}_{kk} < 0 \\
q &= 0 && \dot{\varepsilon}_{kk} \geq 0
\end{aligned}
$$

where Q_1 and Q_2 are dimensionless input constants which default to 1.5 and .06, respectively, and l is a characteristic length given as the square root of the area in two dimensions and as the cube root of the volume in three, a is the local sound speed. This formulation, which is consistent with solid artificial viscosity, has better energy balance for SPH elements. For general applications, Monaghan type artificial viscosity is recommended since this type of artificial viscosity is specifically designed for SPH particles.

The Monaghan type artificial viscosity, which is active when AVIS = 0, is defined as follows:

$$q = \begin{cases} \dfrac{-Q_2\bar{c}_{ij}\phi_{ij} + Q_1\phi_{ij}^2}{\bar{\rho}_{ij}} & v_{ij}x_{ij} < 0 \\ 0 & v_{ij}x_{ij} \geq 0 \end{cases}$$

Where,

$$\phi_{ij} = \frac{h_{ij}v_{ij}x_{ij}}{\left|x_{ij}\right|^2 + \varphi^2}$$

$$\bar{c}_{ij} = 0.5(c_i + c_j)$$

$$\bar{\rho}_{ij} = 0.5(\rho_i + \rho_j)$$

$$h_{ij} = 0.5(h_i + h_j)$$

$$\varphi = 0.1h_{ij}$$

Q_1, Q_2 are input constants. When using Monaghan type artificial viscosity, it is recommended that the user set both Q1 and Q2 to 1.0 on either the *CONTROL_BULK_VISCOSITY or *HOURGLASS keywords; see for example G. R. Liu.

*CONTROL_SPOTWELD_BEAM

Purpose: Provides factors for scaling the failure force resultants of beam spot welds as a function of their parametric location on the contact segment and the size of the segment. Also, an option is provided to replace beam welds with solid hexahedron element clusters.

Card 1	1	2	3	4	5	6	7	8
Variable	LCT	LCS	T_ORT	PRTFLG	T_ORS	RPBHX	BMSID	ID_OFF
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

VARIABLE	DESCRIPTION
LCT	Load curve ID for scaling the response in tension based on the shell element size.
LCS	Load curve ID for scaling the response in shear based on the shell element size.
T_ORT	Table ID for scaling the tension response (and shear response if T_ORS = 0) based on the location of the beam node relative to the centroid of the shell.
PRTFLG	Set this flag to 1 to print for each spot weld attachment: the beam, node, and shell ID's, the parametric coordinates that define the constraint location, the angle used in the table lookup, and the three scale factors obtained from the load curves and table lookup. See Figure 12-87.

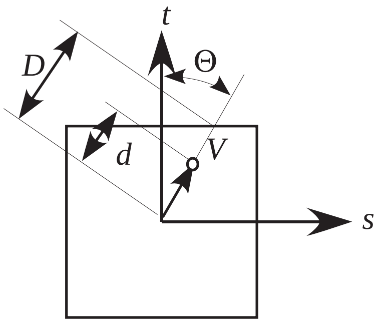


Figure 12-87. Definition of parameters for table definition.

VARIABLE **DESCRIPTION**

T_ORS

Optional table ID for scaling the shear response based on the location of the beam node relative to the centroid of the shell.

RPBHX

Replace each spot weld beam element with a cluster of RPBHX solid elements. The net cross-section of the cluster of elements is dimensioned to have the same area as the replaced beam. RPBHX may be set to 1, 4, or 8. When RPBHX is set to 4 or 8, a table is generated to output the force and moment resultants into the SWFORC file, if this file is active. This table is described by the keyword: *DEFINE_HEX_SPOTWELD_ASSEMBLY. The ID's of the beam elements are used as the cluster spot weld ID's so the ID's in the SWFORC file are unchanged. The beam elements are automatically deleted from the calculation, and the section and material data is automatically changed to be used with solid elements. See Figure 11-24.

BMSID

Optional beam set ID defining the beam element ID's that are to be converted to hex assemblies. If zero, all spot weld beam elements are converted to hex assemblies.

ID_OFF

This optional ID offset applies if and only if BMSID is nonzero. Beams, which share part ID's with beams that are converted to hex assemblies, will be assigned new part ID's by adding to the original part ID the value of ID_OFF. If ID_OFF, is zero the new part ID for such beams will be assigned to be larger than the largest part ID in the model.

Remarks:

The load curves and table provide a means of scaling the response of the beam spot welds to reduce any mesh dependencies for failure model 6 in *MAT_SPOTWELD. Figure 12-88 shows such dependencies that can lead to premature spot weld failure. Separate scale factors are calculated for each of the beam's nodes. The scale factors s_T, s_S, s_{OT} , and s_{OS} are calculated using the load curves LCT, LCS, table T_ORT, and table T_ORS, respectively, and are introduced in the failure criteria,

$$\left[\frac{s_T s_{OT} \sigma_{rr}}{\sigma_{rr}^F\left(\dot{\varepsilon}_{eff}\right)}\right]^2 + \left[\frac{s_S s_{OS} \tau}{\tau^F\left(\dot{\varepsilon}_{eff}\right)}\right]^2 - 1 = 0$$

If a curve or table is given an ID of 0, its scale factor is set to 1.0. The load curves LCT and LCS are functions of the characteristic size of the shell element used in the time step calculation at the start of the calculation. The orientation table is a function of the spot weld's isoparametric coordinate location on the shell element. A vector $V=(s,t)$ is defined from the centroid of the shell to the contact point of the beam's node. The arguments for the orientation table are the angle:

$$\Theta = \tan^{-1}\left[\frac{\min(|s|,|t|)}{\max(|s|,|t|)}\right],$$

and the normalized distance $\bar{d} = {}^{d}\!/_{D} = \max(|s|,|t|)$. See Figure 12-87 The table is

periodic over a range of 0 (V aligned with either the s or t axis) to 45 degrees (V is along the diagonal of the element). The table is specified by the angle of V in degrees, ranging from 0 to 45, and the individual curves give the scale factor as a function of the normalized distance of the beam node, $\overline{d}$, for a constant angle.

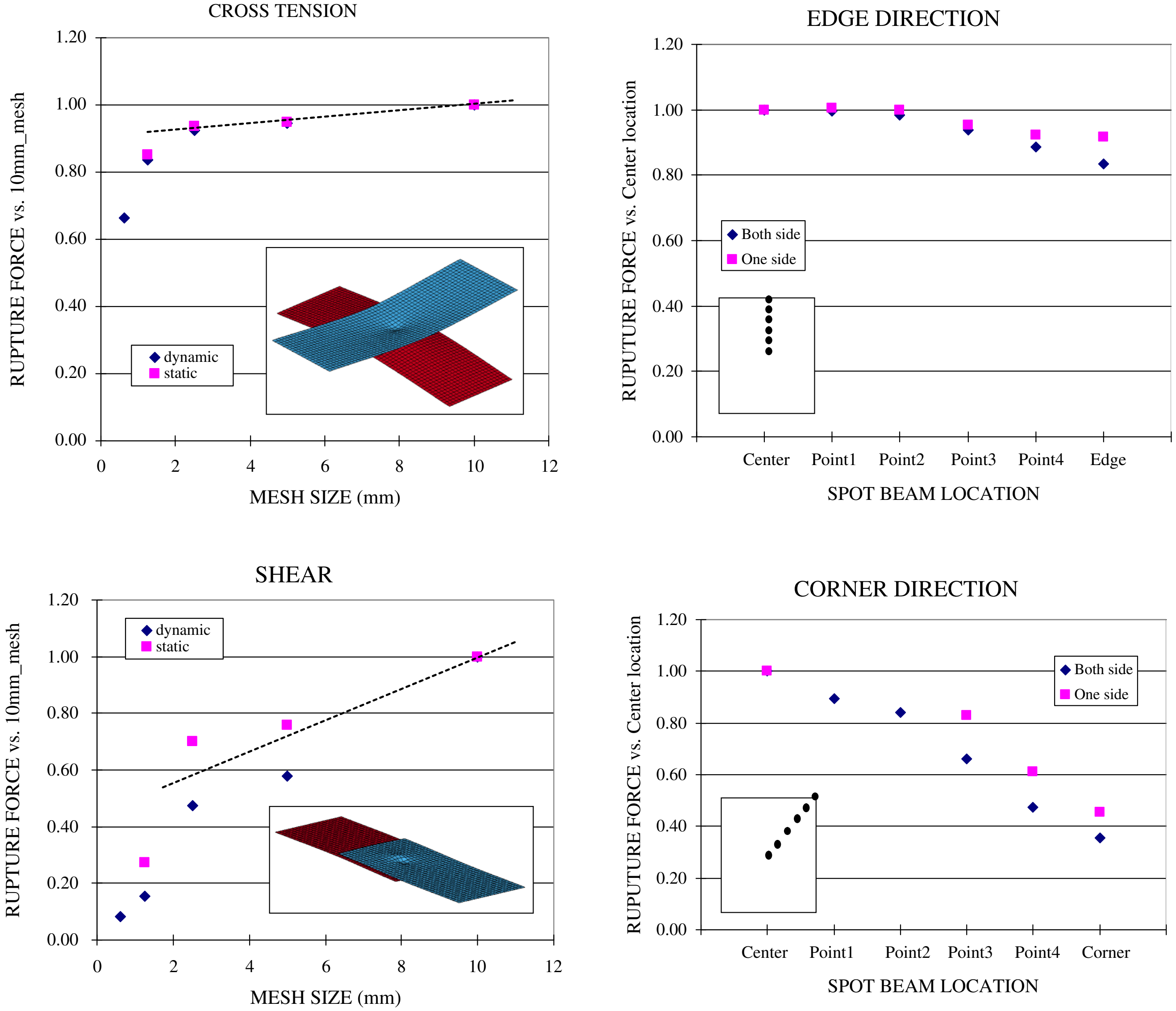


Figure 12-88 The failure force resultants can depend both on mesh size and the location of weld relative to the center of the contact segment

*CONTROL_START

Purpose: Define the start time of analysis.

Card 1	1	2	3	4	5	6	7	8
Variable	BEGTIM							
Type	F							

VARIABLE	DESCRIPTION
BEGTIM	Start time of analysis (default = 0.0). *Load curves are not shifted to compensate for the time offset.* Therefore, this keyword will change the results of any calculation involving time-dependent load curves.

*CONTROL_STAGED_CONSTRUCTION

This control card is used to help break down analyses of construction processes into stages.

Card 1	1	2	3	4	5	6	7	8
Variable	TSTART	STGS	STGE	ACCEL	FACT	STREF	DORDEL	NOPDEL
Type	F	I	I	F	F	I	I	I
Default	0	0	0	0.0	1.e-6	0	0	0

VARIABLE **DESCRIPTION**

TSTART Time at start of analysis (normally leave blank)

STGS Construction stage at start of analysis

STGE Construction stage at end of analysis

ACCEL Default acceleration for gravity loading

FACT Default stiffness and gravity factor for parts before they are added

STREF Reference stage for displacements in d3plot file

DORDEL Dormant part treatment in d3plot file, see notes.

EQ.0: Parts not shown when dormant (flagged as “deleted”),

EQ.1: Parts shown normally when dormant.

NOPDEL Treatment of pressure loads on deleted elements, see notes.

EQ.0: Pressure loads automatically deleted,

EQ.1: No automatic deletion.

Remarks:

See also *DEFINE_CONSTRUCTION_STAGES and *DEFINE_STAGED_CONSTRUCTION_PART.

The staged construction options offer flexibility to carry out the whole construction simulation in one analysis, or to run it stage by stage. Provided that at least one construction stage is defined (*DEFINE_CONSTRUCTION_STAGES), a dynain file will be written at the end of each stage (file names are end_stage001_dynain, etc). These contain node and element definitions and the stress state; the individual stages can then be re-run without re-running the whole analysis. To do this, make a new input file as follows:

- Copy the original input file, containing *DEFINE_CONSTRUCTION_-STAGES and *DEFINE_STAGED_CONSTRUCTION_PART.

- Delete node and element definitions as these will be present in the dynain file (*NODE, *ELEMENT_SOLID, *ELEMENT_SHELL, and *ELEMENT_BEAM).

- Delete any *INITIAL cards; the initial stresses in the new analysis will be taken from the dynain file.

- On *CONTROL_STAGED_CONSTRUCTION set STGS to start at the desired stage

- Add an *INCLUDE statement referencing, for example, end_stage002_dynain if starting the new analysis from Stage 3.

- Move or copy the dynain file into the same directory as the new input file.

When STGS is > 1 the analysis starts at a non-zero time (the start of stage STGS). In this case a dynain file must be included to start the analysis from the stress state at the end of the previous stage. The end time for stage STGE overrides the termination time on *CONTROL_TERMINATION. A new dynain file will be written at the end of all stages from STGS to STGE.

ACCEL and FACT are used with *STAGED_CONSTRUCTION_PART for simpler input definition of the parts present at different construction stages.

If STGS > 1 and elements have been deleted in a previous stage, these elements will be absent from the new analysis and should not be referred to (e.g. *DATABASE_HISTORY_SOLID) in the new input file.

TSTART can be used to set a non-zero start time (again, assuming a compatible dynain file is included). This option is used only if construction stages have not been defined.

STREF allows the user to set a construction stage at the start of which displacements are considered to be zero – e.g. so that initial analysis stages that achieve a pre-construction equilibrium do not contribute to contour plots of displacement. The current coordinates are not modified, only the "initial geometry" coordinates in the d3plot file. If this analysis starts from a stage later than STREF, the reference geometry will be taken from the dynain file that was written at the end of the stage previous to STREF – this dynain file must be in the same directory as the current model for this process to occur. This feature is not available in MPP.

DORDEL: By default, parts for which *DEFINE_STAGED_CONSTRUCTION_PART is defined are flagged as "deleted" in the d3plot file at time-states for which the part is not active (i.e. STGA has not yet been reached). Parts that are deleted because STGR has been reached are also flagged as "deleted". When animating the results, the parts should appear as they become active and disappear as they are deleted. If DORDEL is non-zero, inactive parts (before STGA) are shown normally. The parts are still shown as deleted after STGR is reached.

NOPDEL: By default, pressure load "segments" are automatically deleted by LS-DYNA if they share all four nodes with a deleted solid or shell element. In staged construction, the user may want to apply pressure load to the surface of an element (A) that is initially shared with an element (B), where B is deleted during the calculation. For example, B may be in a layer of soil that is excavated, leaving A as the new top surface. The default scheme would delete the pressure segment when B is removed, despite the fact that A is still present. NOPDEL instructs LS-DYNA to skip the automatic deletion of pressure segments, irrespective of whether the elements have been deleted due to staged construction or material failure. The user must then ensure that pressure loads are not applied to nodes no longer supported by an active element.

*CONTROL_STEADY_STATE_ROLLING

Card 1	1	2	3	4	5	6	7	8
Variable	IMASS	LCDMU	LCDMUR	IVEL	SCL_K			
Type	I	I	I	I	I			
Default	0	0	0	0				

VARIABLE	DESCRIPTION
IMASS	Inertia switching flag EQ.0: include inertia during an implicit dynamic simulation. EQ.1: treat steady state rolling subsystems as quasi-static during implicit dynamic simulations.
LCDMU	Optional load curve for scaling the friction forces in contact.
LCDMUR	Optional load curve for scaling the friction forces in contact during dynamic relaxation. If LCDMUR isn't specified, LCDMU is used.
IVEL	Velocity switching flag. EQ.0: eliminate the steady state rolling body forces and set the velocities of the nodes after dynamic relaxation. EQ.1: keep the steady state rolling body forces after dynamic relaxation instead of setting the velocities.
SCL_K	Scale factor for the friction stiffness during contact loading and unloading. The default values are 1.0 and 0.01 for explicit and implicit, respectively. Any scaling applied here applies only to contact involving the subsystem of parts defined for steady state rolling.

Remarks:

1. Treating the steady state rolling subsystems as quasi-static during an implicit simulation may eliminate vibrations in the system that are not of interest and is generally recommended.

2. Ramping up the friction by scaling it with LCDMU and LCDMUR may improve the convergence behavior of implicit calculations. The values of the load curves should be 0.0 at initial contact and ramp up smoothly to a value of 1.0.

3. After dynamic relaxation, the default behavior is to initialize the nodes with the velocities required to generate the body forces on elements and remove the body forces. This initialization is skipped, and the body forces retained, after dynamic relaxation if IVEL = 1.

4. The friction model in contact is similar to plasticity, where there is an elastic region during the loading and unloading of the friction during contact. The elastic stiffness is scaled from the normal contact stiffness. For implicit calculations, the default scale factor is 0.01, which results in long periods of time being required to build the friction force, and, in some cases, oscillations in the contact forces. A value between 10 and 100 produces smoother solutions and a faster build-up and decay of the friction force as the tire velocity or slip angle is varied, allowing a parameter study to be performed in a single run.

CONTROL_STRUCTURED_*{OPTION}

Available options include:

<BLANK>

TERM

Purpose: Write out an LS-DYNA structured input deck that is largely or wholly equivalent to the keyword input deck. This option may be useful in debugging errors that occur during processing of the input file, particularly if error messages of the type "*** ERROR ##### (STR + ###)" are written. The name of the structured input deck is "dyna.str".

Not all LS-DYNA features are supported in structured input format. Some data such as load curve numbers will be output in an internal numbering system.

If the TERM option is activated, termination will occur after the structured input deck is written.

Adding "outdeck = s" to the LS-DYNA execution line serves the same purpose as including *CONTROL_STRUCTURED in the keyword input deck.

CONTROL_SUBCYCLE_*{K}_{L} or
***CONTROL_SUBCYCLE**_{OPTION}*

Available options for subcycling first form with K and L

$K, L \in \{$<BLANK>$, 1, 2, 4, 8, 16, 32, 64\}$

Available options for multiscale (*OPTION*) include:

<BLANK>

MASS_SCALED_PART

MASS_SCALED_PART_SET

Purpose: This keyword is used to activate subcycling or mass scaling (multiscale). The common characteristic of both methods is that the time step varies from element to element, thereby eliminating unnecessary stepping on more slowly evolving portions of the model. These techniques are suited for reducing the computational cost for models involving large spatial variation in mesh density and/or material characteristics.

Subcycling is described in the LS-DYNA Theory Manual and in detail in Borrvall et.al. [2014] and may be seen as an alternative to using selective mass scaling, see the keyword *CONTROL_TIMESTEP.

This keyword comes in two variations:

1. **Subcycling.** Plain subcycling is activated by the *CONTROL_SUBCYCLE_*{K}*_*{L}* variant of this keyword. This form of the card should *not* be included more than once. It may be used in conjunction with mass scaling to limit the time step characteristics.

   For subcycling, time steps for integration are determined automatically from the characteristic properties of the elements in the model, with the restriction that the ratio between the largest and smallest time step is limited by K. Furthermore, L determines the relative time step at which external forces such as contacts and loads are calculated

   For example, *CONTROL_SUBCYCLE_16_4 limits the largest explicit integration time step to at most 16 times the smallest. Contact forces are evaluated every 4 time steps. The defaults are $K = 16$ and $L = 1$, and L cannot be specified larger than K. This option may be used without mass scaling activated but internally elements may still be slightly mass scaled to maintain computational efficiency.

2. **Mass Scaling/Multiscale.** For a multiscale simulation, mass scaling is mandatory and the time steps are directly specified in the input. The specified

parts (see the PID field) or part sets (see the PSID field) run at the time step specified in the TS field. All other elements evolve with a time step set by |DT2MS|, which is set on *CONTROL_TIMESTEP card.

This feature was motivated by automotive crash simulation, wherein it is common for a small subset of *solid* elements to limit the time step size. With this card the finely meshed parts (consisting of solid elements) can be made to run with a smaller time step through mass scaling so that the rest of the vehicle can run with a time step size of |DT2MS|.

Part Card. Additional card for the MASS_SCALED_PART and MASS_SCALED_-PART_SET keyword options. Provide as many cards as necessary. Input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID/PSID	TS						
Type	I	F						
Default	none	none						

VARIABLE	DESCRIPTION
PID/PSID	Part ID or part set ID if the SET option is specified.
TS	Time step size at which mass scaling is invoked for the PID or PSID

*CONTROL_TERMINATION

Purpose: Stop the job.

Card 1	1	2	3	4	5	6	7	8
Variable	ENDTIM	ENDCYC	DTMIN	ENDENG	ENDMAS	NOSOL		
Type	F	I	F	F	F	I		
Default	0.0	0	0.0	0.0	1.0E+08	0		
Remarks	1		2					

VARIABLE	DESCRIPTION
ENDTIM	Termination time. Mandatory.
ENDCYC	Termination cycle. The termination cycle is optional and will be used if the specified cycle is reached before the termination time. Cycle number is identical with the time step number.
DTMIN	Reduction (or scale) factor to determine minimum time step, *tsmin*, where $tsmin = dtstart \times \text{DTMIN}$ and *dtstart* is the initial step size determined by LS-DYNA. When the time step drops to *tsmin*, LS-DYNA terminates with a restart dump. See the exception described in Remark 2.
ENDENG	Percent change in energy ratio for termination of calculation. If undefined, this option is inactive.
ENDMAS	Percent change in the total mass for termination of calculation. This option is relevant if and only if mass scaling is used to limit the minimum time step size, see *CONTROL_TIMESTEP variable name "DT2MS".
NOSOL	Flag for a non-solution run, i.e. normal termination directly after initialization. EQ.0: off (default), EQ.1: on.

Remarks:

1. Termination by displacement may be defined in the *TERMINATION section.

2. If the erosion flag on *CONTROL_TIMESTEP is set (ERODE = 1), then solid elements and thick shell elements whose time step falls below *tsmin* will be eroded and the analysis will continue. This time-step-based failure option is not recommended when solid formulations 11 or 12 are included in the model. Furthermore, when PSFAIL in *CONTROL_SOLID is nonzero, regardless of the value of ERODE, then all solid elements excepting those with formulation 11 or 12, whose time step falls below *tsmin* will be eroded and the analysis will continue. This time-step-based erosion of solids due to a nonzero PSFAIL is not limited to solids in part set PSFAIL. Only the negative-volume-based erosion criterion is limited to solids in part PSFAIL.

*CONTROL_THERMAL_EIGENVALUE

Purpose: Compute eigenvalues of thermal conductance matrix for model evaluation purposes.

Card 1	1	2	3	4	5	6	7	8
Variable	NEIG							
Type	I							
Default	0						.	.

VARIABLE	DESCRIPTION
NEIG	Number of eigenvalues to compute. EQ.0: No eigenvalues are computed. GT.0: Compute NEIG eigenvalues of each thermal conductance matrix.

Remarks:

1. Computes NEIG eigenvalues for each thermal conductance matrix. This is a model evaluation tool and it is recommended that only a small number, such as 1, thermal time steps are used when using this feature.

*CONTROL_THERMAL_NONLINEAR

Purpose: Set parameters for a nonlinear thermal or coupled structural/thermal analysis. The control card, *CONTROL_SOLUTION, is also required.

Card 1	1	2	3	4	5	6	7	8
Variable	REFMAX	TOL	DCP	LUMPBC	THLSTL	NLTHPR	PHCHPN	
Type	I	F	F	I	F	I	F	
Default	10	1.e-04	1.0 or 0.5	0	0.	0	100.	

VARIABLE	DESCRIPTION

REFMAX

Maximum number of matrix reformations per time step:

EQ.0: set to 10 reformations.

TOL

Convergence tolerance for temperature:

EQ.0.0: set to 1000 * machine roundoff.

DCP

Divergence control parameter:

steady state problems	$0.3 \leq DCP \leq 1.0$	default 1.0
transient problems	$0.0 < DCP \leq 1.0$	default 0.5

LUMPBC

Lump enclosure radiation boundary condition. LUMPBC = 1 activates a numerical method to damp out anomalous temperature oscillations resulting from very large step function boundary conditions. This option is not generally recommended.

EQ.0: off (default)

EQ.1: on

THLSTL

Line search convergence tolerance:

EQ.0.0: No line search

GT.0.0: Line search convergence tolerance

VARIABLE	DESCRIPTION
NLTHPR	Thermal nonlinear print out level: EQ.0: No print out EQ.1: Print convergence parameters during solution of nonlinear system
PHCHPN	Phase change penalty parameter: EQ.0.0: Set to default value 100. GT.0.0: Penalty to enforce constant phase change temperature

*CONTROL_THERMAL_SOLVER

Purpose: Set options for the thermal solution in a thermal only or coupled structural-thermal analysis. The control card, *CONTROL_SOLUTION, is also required.

Card 1	1	2	3	4	5	6	7	8
Variable	ATYPE	PTYPE	SOLVER	CGTOL	GPT	EQHEAT	FWORK	SBC
Type	I	I	I	F	I	F	F	F
Default	0	0	3	$10^{-4}/10^{-6}$	8	1.	1.	0.

Remaining cards are optional.†

Card 2	1	2	3	4	5	6	7	8
Variable	MSGLVL	MAXITR	ABSTOL	RELTOL	OMEGA			TSF
Type	I	I	F	F	F			F
Default	0	500	10^{-10}	10^{-6}	1.0 or 0.			1.

Card 3	1	2	3	4	5	6	7	8
Variable	MXDMP	DTVF	VARDEN					
Type	I	F	I					
Default	0	0.	0					.

Optional card if SOLVER = 17

Card 2	1	2	3	4	5	6	7	8
Variable	MSGLVL	NINNER	ABSTOL	RELTOL	NOUTER			
Type	I	I	F	F	I			
Default	0	100	10^{-10}	10^{-6}	100			

VARIABLE	DESCRIPTION

ATYPE

Thermal analysis type:

EQ.0: Steady state analysis,

EQ.1: transient analysis.

PTYPE

Thermal problem type: (see *CONTROL_THERMAL_NONLINEAR if nonzero)

EQ.0: linear problem,

EQ.1: nonlinear problem with material properties evaluated at gauss point temperature.

EQ.2: nonlinear problem with material properties evaluated at element average temperature.

VARIABLE **DESCRIPTION**

SOLVER

Thermal analysis solver type:

EQ.1: using solver 11 (enter -1 to use the old ACTCOL solver),

EQ.2: nonsymmetric direct solver,

EQ.3: diagonal scaled conjugate gradient iterative (default),

EQ.4: incomplete choleski conjugate gradient iterative,

EQ.5: nonsymmetric diagonal scaled bi-conjugate gradient

EQ.11: symmetric direct solver

For MPP executables:

EQ.11: symmetric direct solver,

EQ.12: diagonal scaling (default for mpp) conjugate gradient iterative,

EQ.13: symmetric Gauss-Siedel conjugate gradient iterative,

EQ.14: SSOR conjugate gradient iterative,

EQ.15: ILDLT0 (incomplete factorization) conjugate gradient iterative,

EQ.16: modified ILDLT0 (incomplete factorization) conjugate gradient iterative.

For Conjugate Heat transfer problems:

EQ.17: GMRES solver.

CGTOL

Convergence tolerance for SOLVER = 3 and 4.

EQ.0.0: use default value 10^{-4} single or 10^{-6} double precision

GPT

Number of Gauss points to be used in the solid elements:

EQ.0.0: use default value 8,

EQ.1.0: one point quadrature is used.

EQHEAT

Mechanical equivalent of heat (see Remark 10).

EQ.0.0: default value 1.0,

LT.0.0: designates a load curve number for EQHEAT versus time.

FWORK

Fraction of mechanical work converted into heat.

EQ.0.0: use default value 1.0.

VARIABLE **DESCRIPTION**

SBC

Stefan Boltzmann constant. Value is used with enclosure radiation surfaces, see *BOUNDARY_RADIATION_…

LT.0.0: use a smoothing algorithm when calculating view factors to force the row sum = 1.

MSGLVL

Output message level (For SOLVER > 10)

EQ.0: no output (default),

EQ.1: summary information,

EQ.2: detailed information, use only for debugging.

MAXITR

Maximum number of iterations. For SOLVER > 11.

EQ.0: use default value 500,

ABSTOL

Absolute convergence tolerance. For SOLVER > 11.

EQ.0.0: use default value 10^{-10}

RELTOL

Relative convergence tolerance. Replaces CGTOL for SOLVER > 11.

EQ.0.0: use default value 10^{-6}

OMEGA

Relaxation parameter omega for SOLVER 14 and 16.

EQ.0.0: use default value 1.0 for Solver 14, use default value 0.0 for Solver 16.

TSF

Thermal Speedup Factor. This factor multiplies all thermal parameters with units of time in the denominator (e.g., thermal conductivity, convection heat transfer coefficients). It is used to artificially time scale the problem. Its main use is in metal stamping. If the velocity of the stamping punch is artificially increased by 1000, then set TSF = 1000 to scale the thermal parameters.

MXDMP

Matrix Dumping for SOLVER > 11

EQ.0: No Dumping

GT.0: Dump using ASCII format every MXDMP time steps.

LT.0: Dump using binary format every |MXDMP| time steps.

DTVF

Time interval between view factor updates.

VARIABLE **DESCRIPTION**

VARDEN Variable thermal density. This parameter is only applicable for solid elements in a coupled thermal-stress problem. Setting this parameter will adjust the material thermal density in the thermal solver to account for very large volume changes when using an EOS or large coefficient of thermal expansion. For most applications, the default value, VARDEN = 0, should be used.

EQ.0: use constant density (default)

EQ.1: modify thermal density to account for volume change when using an EOS.

EQ.2: modify thermal density to account for volume change when using a large coefficient of expansion.

NINNER Number of inner iterations for GMRES solve

NOUTER Number of outer iterations for GMRES solve

Remarks:

1. **Solver Availability in MPP.** Solvers 1, 2, 3 and 4 are only for SMP environments. Solvers 11, 12, 13, 14, 15 and 16 are for SMP and MPP.

2. **Recommended Direct Solver.** Solver 11 is the preferred direct solver. Solver 11 uses sparse matrix storage and requires much less memory than Solver 1.

3. **Direct vs. Iterative Solve.** Use of a direct solver (e.g., SOLVER = 1, 2 or 11) is usually less efficient than using an iterative solver (SOLVER = 3, 4, 12, 13, 14, 15 or 16). Consider using a direct solver to get the model running and then switch to an iterative solver to decrease execution time (particularly for large models). Direct solvers should be used when experiencing slow or no convergence.

4. **Transient Problems.** For transient problems, diagonal scaling conjugate gradient (SOLVER = 3 or 12) should be adequate.

5. **Steady State Problems.** For steady state problems, convergence may be slow or unacceptable, so consider using direct solver (SOLVER = 1, 2 or 11) or a more powerful preconditioner (SOLVER = 4, 13, 14, 15 or 16).

6. **Solvers 13 & 14.** Solver 13 (symmetric Gauss-Seidel) and solver 14 (SSOR) are related. When OMEGA = 1, solver 14 is equivalent to solver 13. The optimal omega value for SSOR is problem dependent but lies between 1 and 2.

7. **Solvers 15 & 16.** Solver 15 (incomplete LDLT0) and solver 16 (modified incomplete LDLT0) are related. Both are no-fill factorizations that require one extra n-vector of storage. The sparsity pattern of the preconditioner is exactly the same as that of the thermal stiffness matrix. Solver 16 uses the relaxation parameter OMEGA. The optimal OMEGA value is problem dependent, but lies between 0 and 1.

8. **Solver 17.** The GMRES solver has been developed as an alternative to the direct solvers in cases where the structural thermal problem is coupled with the fluid thermal problem in a monolithic approach using the ICFD solver. A significant gain of calculation time can be observed when the problem reaches 1M elements.

9. **Completion Conditions for Solvers 12 – 15.** Solvers 12, 13, 14, 15 and 16 terminate the iterative solution process when (1) the number of iterations exceeds MAXITR or (2) the 2-norm of the residual drops below

   $$\text{ABSTOL} + \text{RELTOL} \times \text{2-norm of the initial residual.}$$

10. **Debug Data.** Solvers 11 and up have the ability to dump the thermal conductance matrix and right-hand-side using the same formats as documented under *CONTROL_IMPLICIT_SOLVER. If this option is used files beginning with “T_”will be generated.

11. **Unit Conversion Factor.** EQHEAT is a unit conversion factor. EQHEAT converts the mechanical unit for work into the thermal unit for energy according to,

    $$\text{EQHEAT} \times [\text{work}] = [\text{thermal energy}]$$

    However, it is recommended that a consistent set of units be used with EQHEAT set to 1.0. For example when using SI,

    $$[\text{work}] = 1\text{Nm} = [\text{thermal energy}] = 1\text{J} \Rightarrow \text{EQHEAT} = 1.$$

*CONTROL_THERMAL_TIMESTEP

Purpose: Set time step controls for the thermal solution in a thermal only or coupled structural/thermal analysis. This card requires that the deck also include *CONTROL_-SOLUTION, and, *CONTROL_THERMAL_SOLVER needed.

Card 1	1	2	3	4	5	6	7	8
Variable	TS	TIP	ITS	TMIN	TMAX	DTEMP	TSCP	LCTS
Type	I	F	F	F	F	F	F	I
Default	0	0.5	none	-	-	1.0	0.5	0

VARIABLE	DESCRIPTION
TS	Time step control: EQ.0: fixed time step, EQ.1: variable time step (may increase or decrease).
TIP	Time integration parameter: EQ.0.0: set to 0.5 - Crank-Nicholson scheme, EQ.1.0: fully implicit.
ITS	Initial thermal time step
TMIN	Minimum thermal time step: EQ.0.0: set to structural explicit time step.
TMAX	Maximum thermal time step: EQ.0.0: set to 100 * structural explicit time step.
DTEMP	Maximum temperature change in each time step above which the thermal time step will be decreased: EQ.0.0: set to a temperature change of 1.0.
TSCP	Time step control parameter. The thermal time step is decreased by this factor if convergence is not obtained. 0. < TSCP < 1.0: EQ.0.0: set to a factor of 0.5.

VARIABLE	DESCRIPTION
LCTS	LCTS designates a load curve number which defines data pairs of (thermal time breakpoint, new time step). The time step will be adjusted to hit the time breakpoints exactly. After the time breakpoint, the time step will be set to the 'new time step' ordinate value in the load curve.

*CONTROL_TIMESTEP

Purpose: Set structural time step size control using different options.

Card 1	1	2	3	4	5	6	7	8
Variable	DTINIT	TSSFAC	ISDO	TSLIMT	DT2MS	LCTM	ERODE	MS1ST
Type	F	F	I	F	F	I	I	I
Default	-	0.9 or 0.67	0	0.0	0.0	0	0	0

This card is optional.

Card 2	1	2	3	4	5	6	7	8
Variable	DT2MSF	DT2MSLC	IMSCL			RMSCL		
Type	F	I	I			F		
Default	not used	not used	0			0.0		

VARIABLE	DESCRIPTION
DTINIT	Initial time step size: EQ.0.0: LS-DYNA determines initial step size.
TSSFAC	Scale factor for computed time step (old name SCFT). See Remark 1 below. (Default = 0.90; if high explosives are used, the default is lowered to 0.67).

VARIABLE **DESCRIPTION**

ISDO

Basis of time size calculation for 4-node shell elements. 3-node shells use the shortest altitude for options 0,1 and the shortest side for option 2. This option has no relevance to solid elements, which use a length based on the element volume divided by the largest surface area.

EQ.0: characteristic length is given by

$$\frac{\text{area}}{\min(\text{longest side, longest diagonal})}.$$

EQ.1: characteristic length is given by

$$\frac{\text{area}}{\text{longest diagonal}}.$$

EQ.2: based on bar wave speed and,

$$\max\left[\text{shortest side}, \frac{\text{area}}{\min(\text{longest side, longest diagonal})}\right].$$

> WARNING: Option 2 can give a much larger time step size that can lead to instabilities in some applications, especially when triangular elements are used.

EQ.3: This feature is currently unavailable. Time step size is based on the maximum eigenvalue. This option is okay for structural applications where the material sound speed changes slowly. The cost related to determining the maximum eigenvalue is significant, but the increase in the time step size often allows for significantly shorter run times without using mass scaling.

VARIABLE **DESCRIPTION**

TSLIMT

Shell element minimum time step assignment, TSLIMT. When a shell controls the time step, element material properties (moduli not masses) will be modified such that the time step does not fall below the assigned step size. This option is applicable only to shell elements using material models:

*MAT_PLASTIC_KINEMATIC,

*MAT_POWER_LAW_PLASTICITY,

*MAT_STRAIN_RATE_DEPENDENT_PLASTICITY,

*MAT_PIECE-WISE_LINEAR_PLASTICITY.

> WARNING: This so-called stiffness scaling option is NOT recommended. The DT2MS option below applies to all materials and element classes and is preferred.

If both TSLIMT and DT2MS below are active and if TSLIMT is input as a positive number, then TSLIMT defaults to 10^{-18}, thereby disabling it.

If TSLIMT is negative and less than $|\text{DT2MS}|$, then $|\text{TSLIMT}|$ is applied prior to the mass being scaled. If $|\text{DT2MS}|$ exceeds the magnitude of TSLIMT, then TSLIMT is set to 10^{-18}.

VARIABLE	DESCRIPTION
DT2MS	Time step size for mass scaled solutions. (Default = 0.0) GT.0.0: Positive values are for quasi-static analyses or time history analyses where the inertial effects are insignificant. LT.0.0: TSSFAC × |DT2MS| is the minimum time step size permitted and mass scaling is done if and only if it is necessary to meet the Courant time step size criterion. This option can be used in transient analyses if the mass increases remain insignificant. See also the variable MS1ST below and the *CONTROL_TERMINATION variable ENDMAS. WARNING: Superelements from, *ELEMENT_DIRECT_MATRIX_INPUT, are not mass scaled; consequently, DT2MS does not affect their time step size. In this case an error termination will occur, and DT2MS will need to be reset to a smaller value.
LCTM	Load curve ID that limits the maximum time step size (optional). This load curve defines the maximum time step size permitted versus time. If the solution time exceeds the final time value defined by the curve the computed step size is used. If the time step size from the load curve is exactly zero, the computed time step size is also used.

VARIABLE DESCRIPTION

ERODE

Erosion flag for solids and thick shells.

EQ.0: Calculation will terminate if time step drops to tsmin (see *CONTROL_TERMINATION).

EQ.1: Solids and thick shells whose time step drops to tsmin (see *CONTROL_TERMINATION) will erode, and SPH particles whose time step drops to tsmin will be deactivated.

ERODE = 1 and *tsmin* > 0 also invokes erosion of any solid element whose volume becomes negative, thereby preventing termination of the analysis due to negative volume. The effect of ERODE = 1 on erosion due to negative volue is superceded by a nonzero PSFAIL in *CONTROL_SOLID. PSFAIL serves to limit solid erosion based on negative volume to solids in part set PSFAIL.

MS1ST

Option for mass scaling that applies when DT2MS < 0.

EQ.0: (Default) Mass scaling is considered throughout the analysis to ensure that the minimum time step cannot drop below TSSFAC × |DT2MS|. Added mass may increase with time, but it will never decrease.

EQ.1: Added mass is calculated at the first time step and remains unchanged thereafter. The initial time step will not be less than TSSFAC × |DT2MS|, but the time step may subsequently decrease, depending on how the mesh deforms and the element characteristic lengths change.

DT2MSF

Reduction (or scale) factor for initial time step size to determine the minimum time step size permitted. Mass scaling is done if it is necessary to meet the Courant time step size criterion. If this option is used, DT2MS effectively becomes –DT2MSF multiplied by the initial time step size, Δt, before Δt is scaled by TSSFAC. This option is active if and only if DT2MS = 0 above.

VARIABLE **DESCRIPTION**

DT2MSLC

Load curve for determining the magnitude of DT2MS as a function of time, $f_{\mathrm{DT2MS}}(t)$, during the explicit solutions phase. Time zero must be in the abscissa range of this curve and the ordinate values should all be positive. At a given simulation time t, $f_{\mathrm{DT2MS}}(t) \times \mathrm{sign}(\mathrm{DT2MS})$ plays the role of DT2MS according to the description for DT2MS above. It *is* allowed to use all negative ordinate values in the curve, then $f_{\mathrm{DT2MS}}(t)$ itself (sign and magnitude) determines how mass scaling is performed and DT2MS is neglected. It is however *not* allowed for the ordinate values to change sign during the simulation.

IMSCL

Flag for selective mass scaling if and only if mass scaling active. Selective mass scaling does not scale the rigid body mass and is therefore more accurate. Since it is memory and CPU intensive, it should be applied only to small finely meshed parts.

EQ.0: no selective mass scaling.

EQ.1: all parts undergo selective mass scaling.

LT.0: recommended: |IMSCL| is the part set ID of the parts that undergo selective mass scaling; all other parts are mass scaled the usual way.

RMSCL

Flag for using rotational option in selective mass scaling

EQ.0.: Only translational inertia are selectively mass scaled

NE.0.: Both translational and rotational inertia are selectively mass scaled

Remarks:

During the solution we loop through the elements and determine a new time step size by taking the minimum value over all elements.

$$\Delta t^{n+1} = \mathrm{TSSFAC} \times \min\{\Delta t_1, \Delta t_2, \ldots, \Delta t_N\}$$

where N is the number of elements. The time step size roughly corresponds to the transient time of an acoustic wave through an element using the shortest characteristic distance. For stability reasons the scale factor TSSFAC is typically set to a value of 0.90 (default) or some smaller value. To decrease solution time we desire to use the largest possible stable time step size. Values larger than .90 will often lead to instabilities. Some comments follow:

1. **Sound Speed and Element Size.** The sound speed in steel and aluminum is approximately 5mm per microsecond; therefore, if a steel structure is modeled with element sizes of 5mm, the computed time step size would be 1 microsecond. Elements made from materials with lower sound speeds, such as foams, will give larger time step sizes. Avoid excessively small elements and be aware of the effect of rotational inertia on the time step size in the Belytschko beam element. Sound speeds differ for each material, for example, consider:

Air	331 m/s
Water	1478
Steel	5240
Titanium	5220
Plexiglass	2598

2. **Use Rigid Bodies when Possible.** It is recommended that stiff components be modeled by using rigid bodies. Do not scale the Young's modulus, as that can substantially reduce the time step size.

3. **Triangular Elements.** The altitude of the triangular element should be used to compute the time step size. Using the shortest side is okay only if the calculation is closely examined for possible instabilities. This is controlled by parameter ISDO.

4. **Selective Mass Scaling.** In the explicit time integration context and in contrast to conventional mass scaling, selective mass scaling (SMS) is a well thought out scheme that not only reduces the number of simulation cycles but that also does not significantly affect the dynamic response of the system under consideration. The drawback is that a linear system of equations must be solved in each time step for the accelerations. In this implementation a preconditioned conjugate gradient method (PCG) is used.

   An unfortunate consequence of this choice of solver is that the efficiency will worsen when attempting large time steps since the condition number of the assembled mass matrix increases with the added mass. Therefore caution should be taken when choosing the desired time step size. For large models it is also recommended to only use SMS on critical parts since it is otherwise likely to slow down execution; the bottleneck being the solution step for the system of linear system of equations.

   While some constraints and boundary conditions available in LS-DYNA are not supported for SMS they can be implemented upon request from a user.

   A partial list of constraints and boundary conditions supported with SMS:

   Pointwise nodal constraints in global and local directions

   Prescribed motion in global and local directions

Adaptivity

Rigid walls

Deformable elements merged with rigid bodies

Constraint contacts and spotwelds

Beam release constraints

By default, only the translational dynamic properties are treated. This means that only rigid body translation will be unaffected by the mass scaling imposed. There is an option to also properly treat rigid body rotation in this way, this is invoked by flagging the parameter RMSCL. A penalty in computational expense is incurred but the results could be improved if rotations are dominating the simulation.

*CONTROL_UNITS

Purpose: Specify the user units for the current keyword input deck. This does not provide any mechanism for automatic conversion of units of any entry in the keyword input deck. It is intended to be used for several purposes, but currently only for the situation where an external database in another set of units will be loaded and used in the simulation. In this case, ***CONTROL_UNITS** provides the information necessary to convert the external data into internal units (see ***CHEMISTRY_CONTROL** for such external databases).

If the needed unit is not one of the predefined ones listed for use on the first card, then the second optional card is used to define that unit. Any non-zero scales that are entered on optional card 2 override what is specified on the first card. These scales are given in terms of the default units on card 1. For instance, if 3600.0 is given in the second 20 character field on the optional second card (TIME_SCALE), then 'hour' is the time unit (3600 seconds).

Card 1	1	2	3	4	5	6	7	8
Variable	LENGTH	TIME	MASS	TEMP				
Type	A	A	A	A				
Default	m	sec	kg	K				

Optional Card only used when a new unit needs to be defined:

Card 2	1	2	3	4	5	6	7	8
Variable	LENGTH_SCALE		TIME_SCALE		MASS_SCALE			
Type	F		F		F			
Default	1.0		1.0		1.0			

VARIABLE	DESCRIPTION
LENGTH	Length units: EQ.m: meter (default) EQ.mm: millimeter EQ.cm: centimeter EQ.in: inch EQ.ft: foot
TIME	Time units: EQ.sec: second (default) EQ.ms: msec, millisec EQ.micro_s: microsec
MASS	Mass units: EQ.kg: kilogram (default) EQ.g: gram EQ.mg: milligram EQ.lb: pound EQ.slug: pound × sec2/foot EQ.slinch: pound × sec2/inch EQ.mtrc_ton: metric_ton
TEMP	Temperature units: EQ.K: Kelvin (default) EQ.C: Celsius EQ.F: Fahrenheit EQ.R: Rankine
LENGTH_ SCALE	Number of meters in the length unit for the input deck
TIME_ SCALE	Number of seconds in the time unit for the input deck
MASS_ SCALE	Number of kilograms in the mass unit for the input deck

*DAMPING

The Keyword options in this section in alphabetical order are:

DAMPING_FREQUENCY_RANGE_*{OPTION}

*DAMPING_GLOBAL

*DAMPING_PART_MASS

*DAMPING_PART_STIFFNESS

*DAMPING_RELATIVE

DAMPING_FREQUENCY_RANGE_*{OPTION}

Purpose: This feature provides approximately constant damping (i.e. frequency-independent) over a range of frequencies.

Available OPTIONS are:

<BLANK>	Applies damping to global motion
DEFORM	Applies damping to element deformation

Card 1	1	2	3	4	5	6	7	8
Variable	CDAMP	FLOW	FHIGH	PSID	(blank)	PIDREL	IFLG	
Type	F	F	F	I		I	I	
Default	0.0	0.0	0.0	0		0	0	

VARIABLE	DESCRIPTION
CDAMP	Damping in fraction of critical.
FLOW	Lowest frequency in range of interest (cycles per unit time, e.g. Hz if time unit is seconds)
FHIGH	Highest frequency in range of interest (cycles per unit time, e.g. Hz if time unit is seconds)
PSID	Part set ID. The requested damping is applied only to the parts in the set. If PSID = 0, the damping is applied to all parts except those referred to by other *DAMPING_FREQUENCY_RANGE cards.
PIDREL	Optional part ID of rigid body. Damping is then applied to the motion relative to the rigid body motion. This input does not apply to the DEFORM option.
IFLG	Method used for internal calculation of damping constants: EQ.0: Iterative (more accurate), EQ.1: Approximate (same as R9 and previous versions).

Remarks:

This feature provides approximately constant damping (i.e. frequency-independent) over a range of frequencies. $F_{\mathrm{low}} < F < F_{\mathrm{high}}$It is intended for small damping ratios (e.g. < 0.05) and frequency ranges such that $F_{\mathrm{high}}/F_{\mathrm{low}}$ is in the range 10 to 300. The drawback to this method of damping is that it reduces the dynamic stiffness of the model, especially at low frequencies.

Where the model contains, for example, a rigid foundation or base, the effects of this stiffness reduction can be ameliorated by using PIDREL. In this case, the damping forces resist motion relative to the base, and are reacted onto the rigid part PIDREL. "Relative motion" here means the difference between the velocity of the node being damped, and the velocity of a point rigidly connected to PIDREL at the same coordinates as the node being damped.

This effect is predictable: the natural frequencies of modes close to F_{low} are reduced by 3% for a damping ratio of 0.01 and $F_{\mathrm{high}}/F_{\mathrm{low}}$ in the range 10-30. Near F_{high} the error is between zero and one third of the error at F_{low}. Estimated frequency errors are shown in the next table.

Damping Ratio	% error for $F_{\mathrm{high}}/F_{\mathrm{low}}$ =		
	3 to 30	30 to 300	300 to 3000
0.01	3%	4.5%	6%
0.02	6%	9%	12%
0.04	12%	18%	24%

It is recommended that the elastic stiffnesses in the model be increased slightly to account for this, e.g. for 0.01 damping across a frequency range of 30 to 600Hz, the average error across the frequency range is about 2%. Increase the stiffness by $(1.02)^2$, i.e. by 4%.

Starting from R10, an iterative method is used for the internal calculation of the damping constants (see IFLG above). The new method results in the actual damping matching the user-input damping ratio CDAMP more closely across the frequency range FLOW to FHIGH. As an example, for CDAMP = 0.01, FLOW = 1 Hz and FHIGH = 30 Hz, the actual damping achieved by the previous approximate method varied between 0.008 and 0.012 (different values at different frequencies), i.e. there were errors of up to 20% of the target CDAMP. With the iterative algoritm, the errors are reduced to 1% of the target CDAMP.

Keyword option DEFORM:

The DEFORM option applies damping to the element responses (unlike the standard *DAMPING_FREQUENCY_RANGE which damps the global motion of the nodes). Therefore, rigid body motion is not damped when the DEFORM keyword option is used. For this reason, DEFORM is recommended over the standard option. The damping is adjusted based on current tangent stiffness; this is believed to be more appropriate for a nonlinear analysis, which could be over-damped if a strain-rate-proportional or viscous damping scheme were used.

It works with the following element formulations:

- Solids – types -1, -2, 1, 2, 3, 4, 9, 10, 13, 15, 16, 17, 99
- Beams – types 1, 2, 3, 4, 5, 9 (note: not type 6)
- Shells – types 1-5, 7-17, 20, 21, 23-27, 99
- Discrete elements

The DEFORM option differs from the standard option in several ways:

Standard Damping vs. Deformation Damping

Characteristic Property	Keyword Option	
	<BLANK>	DEFORM
Damping on	Node velocities	Element responses
Rigid body motion	Can be damped	Never damped
Natural frequencies	Reduced (by percentages shown in the above table)	Increased (percentages shown in the above table)
Recommended compensation	Increase elastic stiffness	Reduce elastic stiffness
Effect on timestep	None	Small reduction applied automatically, same percentage as in the frequency change
Element types damped	All	See list above
Damping energy output	Included in “system damping energy”	Included in Internal Energy only if RYLEN = 2 on *CONTROL_ENERGY

*DAMPING_GLOBAL

Purpose: Define mass weighted nodal damping that applies globally to the nodes of deformable bodies and to the mass center of the rigid bodies. For specification of mass damping by part ID or part set ID, use *DAMPING_PART_MASS.

Card 1	1	2	3	4	5	6	7	8
Variable	LCID	VALDMP	STX	STY	STZ	SRX	SRY	SRZ
Type	I	F	F	F	F	F	F	F
Default	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Remarks	1		2	2	2	2	2	2

VARIABLE	DESCRIPTION
LCID	Load curve ID (see *DEFINE_CURVE) which specifies the system damping constant vs. time: EQ.0: a constant damping factor as defined by VALDMP is used, GT.0: system damping is given by load curve LCID (which must be an integer). The damping force applied to each node is $f = -d(t)mv$, where $d(t)$ is defined by load curve LCID.
VALDMP	System damping constant, D_s (this option is bypassed if the load curve number defined above is non zero).
STX	Scale factor on global x translational damping forces.
STY	Scale factor on global y translational damping forces.
STZ	Scale factor on global z translational damping forces.
SRX	Scale factor on global x rotational damping moments.
SRY	Scale factor on global y rotational damping moments.
SRZ	Scale factor on global z rotational damping moments.

Remarks:

1. **Restart.** This keyword is also used for the restart, see *RESTART.

2. **Defaults for Scale Factors.** If STX = STY = STZ = SRX = SRY = SRZ = 0.0 in the input above, all six values are defaulted to unity.

3. **Damping Exceptions.** Mass damping will not be applied to deformable nodes with prescribed motion or to nodes tied with CONSTRAINED_NODE_SET.

4. **Formulation.** With mass proportional system damping the acceleration is computed as:

$$\mathbf{a}^n = \mathbf{M}^{-1}\left(\mathbf{P}^n - \mathbf{F}^n - \mathbf{F}^n_{\text{damp}}\right)$$

where, $\mathbf{M}$ is the diagonal mass matrix, $\mathbf{P^n}$ is the external load vector, $\mathbf{F}^n$ is the internal load vector, and $\mathbf{F}^n_{\text{damp}}$ is the force vector due to system damping. This latter vector is defined as:

$$\mathbf{F}^n_{\text{damp}} = D_s m\mathbf{v}$$

The best damping constant for the system is usually some value approaching the critical damping factor for the lowest frequency mode of interest.

$$(D_s)_{\text{critical}} = 2\omega_{\text{min}}$$

The natural frequency ω_{min} (given in radians per unit time) is generally taken as the fundamental frequency of the structure. This frequency can be determined from an eigenvalue analysis or from an undamped transient analysis. Note that this damping applies to both translational and rotational degrees of freedom. Also note that mass proportional damping will damp rigid body motion as well as vibration.

Energy dissipated by through mass weighted damping is reported as system damping energy in the ASCII file glstat. This energy is computed whenever system damping is active.

DAMPING_PART_MASS_*{OPTION}

OPTION specifies that a part set ID is given with the single option:

<BLANK>

SET

If not used a part ID is assumed.

Purpose: Define mass weighted damping by part ID. Parts may be either rigid or deformable. In rigid bodies the damping forces and moments act at the center of mass. This command may appear multiple times in an input deck but cannot be combined with *DAMPING_GLOBAL.

Card 1	1	2	3	4	5	6	7	8
Variable	PID/PSID	LCID	SF	FLAG				
Type	I	I	F	I				
Default	0	0	1.0	0				

Scale Factor Card. Additional Card for FLAG = 1.

Card 2	1	2	3	4	5	6	7	8
Variable	STX	STY	STZ	SRX	SRY	SRZ		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	0.0	0.0	0.0		

VARIABLE	DESCRIPTION
PID/PSID	Part ID, see *PART or part set ID, see *SET_PART.
LCID	Load curve ID (see *DEFINE_CURVE) which specifies the damping constant vs. time, applied to the part(s) specified in PID/PSID.

VARIABLE	DESCRIPTION
SF	Scale factor for load curve. This allows a simple modification of the load curve values.
FLAG	Set this flag to unity if the global components of the damping forces require separate scale factors.
STX	Scale factor on global x translational damping forces.
STY	Scale factor on global y translational damping forces.
STZ	Scale factor on global z translational damping forces.
SRX	Scale factor on global x rotational damping moments.
SRY	Scale factor on global y rotational damping moments.
SRZ	Scale factor on global z rotational damping moments.

Remarks:

Mass weighted damping damps all motions including rigid body motions. For high frequency oscillatory motion stiffness weighted damping may be preferred. With mass proportional system damping the acceleration is computed as:

$$\boldsymbol{\alpha}^n = \mathbf{M}^{-1}(\mathbf{P}^n - \mathbf{F}^n - \mathbf{F}^n_{\text{damp}})$$

where, $\mathbf{M}$ is the diagonal mass matrix, $\mathbf{P}^n$ is the external load vector, $\mathbf{F}^n$ is the internal load vector, and $\mathbf{F}^n_{\text{damp}}$ is the force vector due to system damping. This latter vector is defined as:

$$\mathbf{F}^n_{\text{damp}} = D_s m \boldsymbol{v}$$

The critical damping constant for the lowest frequency mode of interest is

$$D_s = 2\omega_{\text{min}}$$

where ω_{min} is that lowest frequency in units of radians per unit time. The damping constant specified as the ordinate of curve LCID is typically less than the critical damping constant. The damping is applied to both translational and rotational degrees of freedom. The component scale factors can be used to limit which global components see damping forces.

Energy dissipated by through mass weighted damping is reported as system damping energy in the ASCII file glstat. This energy is computed whenever system damping is active.

Mass damping will not be applied to deformable nodes with prescribed motion or to nodes tied with CONSTRAINED_NODE_SET.

*DAMPING_PART_STIFFNESS_{OPTION}

OPTION specifies that a part set ID is given with the single option:

<BLANK>

SET

If the SET option is not used, a part ID goes in the first field of Card 1.

Purpose: Assign Rayleigh stiffness damping coefficient by part ID or part set ID. This damping command does not apply to parts comprised of discrete elements (*ELEMENT_DISCRETE) or discrete beams (*ELEMENT_BEAM with ELFORM = 6).

Card 1	1	2	3	4	5	6	7	8
Variable	PID/PSID	COEF						
Type	I	F						
Default	none	0.0						

VARIABLE	DESCRIPTION
PID/PSID	Part ID (see *PART) or part set ID (see *SET_PART).
COEF	Rayleigh damping coefficient. Two methods are now available: LT.0.0: Rayleigh damping coefficient in units of time, set based on a given frequency and applied uniformly to each element in the specified part or part set. See remarks below. EQ.0.0: Inactive. GT.0.0: Rayleigh damping coefficient for stiffness weighted damping. Values between 0.01 and 0.25 are recommended. Higher values are strongly discouraged, and values less than 0.01 may have little effect. The damping coefficient is uniquely defined for each element of the part ID.

Remarks:

The damping matrix in Rayleigh damping is defined as:

$$\mathbf{C} = \alpha\mathbf{M} + \beta\mathbf{K}$$

where **C**, **M**, and **K** are the damping, mass, and stiffness matrices, respectively. The constants α. and β are the mass and stiffness proportional damping constants. The mass proportional damping can be treated by system damping, see keywords: *DAMPING_GLOBAL and DAMPING_PART_MASS. Transforming **C** with the i$^{\text{th}}$ eigenvector $\boldsymbol{\phi}_i$ gives:

$$\boldsymbol{\phi}_i^{\mathrm{T}}\mathbf{C}\boldsymbol{\phi}_i = \boldsymbol{\phi}_i^{\mathrm{T}}(\alpha\mathbf{M} + \beta\mathbf{K})\boldsymbol{\phi}_i = \alpha + \beta\omega_i^2 = 2\omega_i\xi_i\delta_{ij}$$

where ω_i is the ith frequency (radians/unit time) and ξ_i is the corresponding modal damping parameter.

Generally, the stiffness proportional damping is effective for high frequencies and is orthogonal to rigid body motion. Mass proportional damping is more effective for low frequencies and will damp rigid body motion. <u>If a large value of the stiffness based damping coefficient is used, it may be necessary to lower the time step size significantly</u>. This must be done manually by reducing the time step scale factor on the *CONTROL_TIMESTEP control card. Since a good value of β is not easily identified, the coefficient, COEF, is defined such that a value of .10 roughly corresponds to 10% damping in the high frequency domain.

In LS-DYNA versions prior to 960 or if COEF is input as less than 0, the critical damping coefficient is equal to 2 divided by ω_i. For example, 10% of critical damping in the ith mode corresponds to

$$\beta = \frac{0.20}{\omega_i}$$

and COEF would be input as -β. Typically, this method of applying stiffness damping is stable only if β is significantly smaller than the explicit time step size.

Energy dissipated by Rayleigh damping is computed if and only if the flag, RYLEN, on the control card, *CONTROL_ENERGY is set to 2. This energy is accumulated as element internal energy and is included in the energy balance. In the glstat file this energy will be lumped in with the internal energy.

NOTE: Type 2 beam elements are a special case in which COEF is internally scaled by 0.1. Thus there is a factor of 10 less damping than stated above. This applies to both negative and positive values of COEF.

*DAMPING_RELATIVE

Purpose: Apply damping relative to the motion of a rigid body. For example, it could damp the deformation of a rotating tire relative to the wheel without damping the rotating motion.

Card 1	1	2	3	4	5	6	7	8
Variable	CDAMP	FREQ	PIDRB	PSID	DV2	LCID		
Type	F	F	F	I	F	I		
Default	0	0	0	0	0.0	0		

VARIABLE	DESCRIPTION
CDAMP	Fraction of critical damping.
FREQ	Frequency at which CDAMP is to apply (cycles per unit time, e.g. Hz if time unit is seconds).
PIDRB	Part ID of rigid body, see *PART. Motion relative to this rigid body will be damped.
PSID	Part set ID. The requested damping is applied only to the parts in the set.
DV2	Optional constant for velocity-squared term. See remarks.
LCID	ID of curve that defines fraction of critical damping vs. time. CDAMP will be ignored if LCID is non-zero.

Remarks:

1. This feature provides damping of vibrations for objects that are moving through space. The vibrations are damped, but not the rigid body motion. This is achieved by calculating the velocity of each node relative to that of a rigid body, and applying a damping force proportional to that velocity. The forces are reacted onto the rigid body such that overall momentum is conserved. It is intended that the rigid body is embedded within the moving object.

2. Vibrations at frequencies below FREQ are damped by more than CDAMP, while those at frequencies above FREQ are damped by less than CDAMP. It is

recommended that FREQ be set to the frequency of the lowest mode of vibration.

3. The damping force of each node is calculated as follows:

$$F = -(D\ .\ m\ .\ v) - (DV2\ .\ m\ .v^{2)}.$$

Where:

$D = 4\pi\ .\ \text{CDAMP}\ .\ \text{FREQ}$

m = mass of the node

v = velocity of node relative to the velocity of a point on the rigid body at the same coordinates as the node.

*DATABASE

The database definitions are optional, but are necessary to obtain output files containing results information. In this section the database keywords are defined in alphabetical order:

DATABASE_*OPTION

*DATABASE_ALE

*DATABASE_ALE_MAT

DATABASE_BINARY_*OPTION

DATABASE_BINARY_*D3PROP

*DATABASE_CPM_SENSOR

DATABASE_CROSS_SECTION_*OPTION1_{OPTION2}

DATABASE_EXTENT_*OPTION

*DATABASE_FATXML

*DATABASE_FORMAT

DATABASE_FREQUENCY_ASCII_*OPTION

DATABASE_FREQUENCY_BINARY_*OPTION

*DATABASE_FSI

*DATABASE_FSI_SENSOR

DATABASE_HISTORY_*OPTION

*DATABASE_MASSOUT

*DATABASE_NODAL_FORCE_GROUP

*DATABASE_PROFILE

*DATABASE_PAP_OUTPUT

*DATABASE_PWP_FLOW

*DATABASE_PWP_OUTPUT

*DATABASE_RCFORC_MOMENT

*DATABASE_RECOVER_NODE

*DATABASE_SPRING_FORWARD

*DATABASE_SUPERPLASTIC_FORMING

*DATABASE_TRACER

*DATABASE_TRACER_GENERATE

The ordering of the database definition cards in the input file is completely arbitrary.

DATABASE_*OPTION1_{OPTION2}

OPTION1 specifies the type of database. LS-DYNA will *not* create an ASCII database unless the corresponding *DATABASE_*OPTION1* card is included in the input deck. OPTION1 may be any of the items in the following list:

ABSTAT Airbag statistics.

ATDOUT Automatic tiebreak damage statistics for *CONTACT_AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE_TIEBREAK, OPTIONs 7, 9, 10, and 11 (only SMP at the moment).

AVSFLT AVS database. See *DATABASE_EXTENT_*OPTION*.

BEARING *ELEMENT_BEARING force output.

BNDOUT Boundary condition forces and energy

CURVOUT Output from *DEFINE_CURVE_FUNCTION.

DEFGEO Deformed geometry file. (Note that to output this file in Chrysler format insert the following line in your .cshrc file: "setenv LSTC_DEFGEO chrysler") The nasbdf file (NASTRAN Bulk Data) is created whenever the DEFGEO file is requested.

DCFAIL Failure function data for *MAT_SPOTWELD_DAIMLERCHRYSLER

DEFORC Discrete spring and damper element (*ELEMENT_DISCRETE) data. If the user wishes to be selective about which discrete elements are output in deforc, use *DATABASE_HISTORY_DISCRETE_*OPTION* to select elements for output (but only if BEAM = 0 in *DATABASE_BINARY_D3PLOT) or set PF = 1 in *ELEMENT_DISCRETE to turn off output for particular elements; otherwise all discrete elements are output.

DEMASSFLOW Measure mass flow rate across defined plane and use together with *DEFINE_DE_MASSFLOW_PLANE.

DISBOUT Discrete beam element, type 6, relative displacements, rotations, and force resultants, all in the local coordinate system, which is also output. Use with *DATABASE_HISTORY_BEAM.

ELOUT Element data. See *DATABASE_HISTORY_*OPTION*. Also, see Card 3 of the *DATABASE_EXTENT_BINARY parameters INTOUT and NODOUT. This latter option will output all integration point data or extrapolated data to the connectivity nodes in a file call eloutdet.

GCEOUT Geometric contact entities.

GLSTAT Global data. Always obtained if ssstat file is activated.

H3OUT Hybrid III rigid body dummies.

JNTFORC Joint force file

MATSUM Material energies. See Remarks 1 and 2 below.

MOVIE See MOVIE option of *DATABASE_EXTENT_*OPTION*.

MPGS See MPGS option of *DATABASE_EXTENT_*OPTION*.

NCFORC Nodal interface forces. See *CONTACT - Card 1 (SPR and MPR)

NODFOR Nodal force groups. See *DATABASE_NODAL_FORCE_GROUP.

NODOUT Nodal point data. See *DATABASE_HISTORY_NODE_*OPTION*.

PBSTAT Particle blast data. See *PARTICLE_BLAST

PLLYOUT Pulley element data for *ELEMENT_BEAM_PULLEY.

PRTUBE Pressure tube data for *DEFINE_PRESSURE_TUBE.

RBDOUT Rigid body data. See Remark 2 below.

RCFORC Resultant interface forces. Output in a local coordinate system is available, see *CONTACT, Optional Card C.

RWFORC Wall forces.

SBTOUT Seat belt output file

SECFORC Cross section forces. See *DATABASE_CROSS_SECTION_*OPTION*.

SLEOUT Sliding interface energy. See *CONTROL_ENERGY

SPCFORC SPC reaction forces.

SPHOUT SPH data. See *DATABASE_HISTORY_*OPTION*.

SSSTAT Subsystem data. See *DATABASE_EXTENT_SSSTAT.

SWFORC Nodal constraint reaction forces (spot welds and rivets).

TPRINT Thermal output from a coupled structural/thermal or thermal only analysis. Includes all nodes unless *DATABASE_HISTORY_-NODE_OPTION is also provided in the keyword input.

TRHIST Tracer particle history information. See *DATABASE_TRACER.

OPTION2, if it set, must be set to FILTER, and this can only be used when OPTION1 is set to NCFORC. When set to FILTER the keyword requires an additional data card, see Card 2 below.

To include global and subsystem mass and inertial properties in the glstat and ssstat files add the keyword option MASS_PROPERTIES as show below. If this option is active the current mass and inertia properties are output including the principle inertias and their axes. Mass of deleted nodes and rigid bodies are not included in the calculated properties.

GLSTAT_MASS_PROPERTIES This is an option for the glstat file to include mass and inertial properties.

SSSTAT_MASS_PROPERTIES This is an option for the ssstat file to include mass and inertial properties for the subsystems.

Card 1	1	2	3	4	5	6	7	8
Variable	DT	BINARY	LCUR	IOOPT	OPTION1	OPTION2	OPTION3	OPTION4
Type	F	I	I	I	F/I	I	I	I
Default	0.	1 or 2	none	0.	0	0	0	0

VARIABLE **DESCRIPTION**

DT Time interval between outputs. If DT is zero, no output is printed.

BINARY Flag for binary output. See remarks under "Output Files and Post-Processing" in Appendix O, "LS-DYNA MPP User Guide."

EQ.1: ASCII file is written: This is the default for shared memory parallel (SMP) LS-DYNA executables.

EQ.2: Data written to a binary database "binout", which contains data that would otherwise be output to the ASCII file. The ASCII file in this case is not created. This is the default for MPP LS-DYNA executables.

EQ.3: ASCII file is written and the data is also written to the binary database (NOTE: MPP LS-DYNA executables will only produce the binary database).

LCUR Optional curve ID specifying time interval between dumps. Use *DEFINE_CURVE to define the curve; abscissa is time and ordinate is time interval between dumps.

IOOPT Flag to govern behavior of the plot frequency load curve defined by LCUR:

EQ.1: At the time each plot is generated, the load curve value is added to the current time to determine the next plot time.(this is the default behavior)

EQ.2: At the time each plot is generated, the next plot time, t, is

VARIABLE DESCRIPTION

computed so that

$$t = \text{the current time} + \text{LCUR}(t)\ .$$

EQ.3: A plot is generated for each abscissa point in the load curve definition. The actual value of the load curve is ignored.

OPTION1

OPTION1 applies to either the bndout, nodout or elout files. For the nodout file OPTION1 is a *real* variable that defines the time interval between outputs for the high frequency file, nodouthf. If OPTION1 is zero, no output is printed. Nodal points that are to be output at a higher frequency are flagged using HFO in the DATABASE_HISTORY_NODE_LOCAL input.

For the elout file OPTION1 is an *integer* variable that gives the number of additional history variables written into the elout file for each integration point in the solid elements. See Remark 7 below for the elout file and Remark 9 for the bndout file.

OPTION2

OPTION2 applies to either the bndout, nodouthf or elout files. For the nodouthf OPTION2 defines the binary file flag for the high frequency nodouthf file. See BINARY above.

For the elout file OPTION2 is an *integer* variable that gives the number of additional history variables written into the elout file for each integration point in the shell elements. See Remark 7 below for the elout file and Remark 9 for the bndout file.

OPTION3

OPTION3 applies to the bndout and elout files only. For the elout file OPTION3 is an *integer* variable that gives the number of additional history variables written into the elout file for each integration point in the thick shell elements. See Remark 7 below for the elout file and Remark 9 for the bndout file.

OPTION4

OPTION4 applies to the bndout and elout files only. For the elout file OPTION4 is an *integer* variable that gives the number of additional history variables written into the elout file for each integration point in the beam elements. See Remark 7 below for the elout file and Remark 9 for the bndout file.

The following Card 2 applies only to *DATABASE_NCFORC_FILTER

Card 2	1	2	3	4	5	6	7	8
Variable	RATE	CUTOFF	WINDOW	TYPE				
Type	F	F	F	I				
Default	none	none	none	0	0	0	0	0

VARIABLE	DESCRIPTION

RATE Time interval T between filter sampling.

CUTOFF Frequency cut-off C in Hz.

WINDOW The width of the window W in units of time for storing the single, forward filtering required for the TYPE = 2 filter option. Increasing the width of the window will increase the memory required for the analysis. A window that is too narrow will reduce the amplitude of the filtered result significantly, and values below 15 are not recommended for that reason. In general, the results for the TYPE = 2 option are sensitive to the width of the window and experimentation is required.

TYPE Flag for filtering options.

EQ.0: No filtering (default).

EQ.1: Single pass, forward Butterworth filtering.

EQ.2: Two pass filtering over the specified time window. Backward Butterworth filtering is applied to the forward Butterworth results that have been stored. This option improves the phase accuracy significantly at the expense of memory.

The file names and corresponding unit numbers are:

Description	I/O Unit #	File Name
Airbag statistics	43	abstat
Automatic tiebreak damage	92	atdout
ASCII database	44	avsflt
Boundary conditions	46	bndout (nodal forces and energies)

Description	I/O Unit #	File Name
Smug animator database	40	defgeo
Discrete elements	36	deforc
Discrete elements mass flow	219	demflow
Discrete beam elements	215	disbout
Element data	34	elout
Contact entities	48	gceout
Global data	35	glstat
Joint forces	53	jntforc
Material energies	37	matsum
MOVIE file family	50	movie*nnn*.xxx where.*nnn*=001-999
MPGS file family	50	mpgsnnn.xxx where *nnn* = 001-999
Nastran/BDF file	49	nasbdf (see comment below)
Nodal interface forces	38	ncforc
Nodal force group	45	nodfor
Nodal point data	33	nodout
Pulley element data	216	pllyout
Pressure tube data	421	prtube
Rigid body data	47	rbdout
Resultant interface forces	39	rcforc
Rigidwall forces	32	rwforc
Seat belts	52	sbtout
Cross-section forces	31	secforc
Interface energies	51	sleout
SPC reaction forces	41	spcforc
SPH element data	68	sphout
Subsystems statistics	58	ssstat
Nodal constraint resultants	42	swforc (spot welds/rivets)
Thermal output	73	tprint
Tracer particles	70	trhist

Output Components for ASCII Files.

ABSTAT
volume
pressure
internal energy
input mass flow rate
output mass flow rate
mass
temperature
density

BNDOUT
x, y, z force
energies
moment (rigid bodies)

DCFAIL
failure function
normal term
bending term
shear term
weld area
effective strain rate
axial force
shear force
torsional moment
bending moment

DEFORC
x, y, z force

ELOUT		
Beams	**(t)Shells**	**Solids**
axial force resultant	xx, yy, zz stress	xx, yy, zz stress
s shear resultant	xy, yz, zx stress	xy, yz, zx stress
t shear resultant	plastic strain	effective stress
s moment resultant	xx, yy, zz strain†	yield function
t moment resultant	xy, yz, zx strain†	xx, yy, zz strain†
torsional resultant		xy, yz, zx strain†

† Strains written for solids and for lower and upper integration points of shells and tshells if STRFLG = 1 in *DATABASE_EXTENT_BINARY.

GCEOUT	
x, y, z force	x, y, z moment

GLSTAT	
time step	total energy
kinetic energy	external work
internal energy	total and initial energy
sprint and damper energy	energy ratio without eroded energy
hourglass energy	element & part ID controlling time step
system damping energy	global x, y, z velocity
sliding interface energy	time per zone cycle
eroded kinetic energy	joint internal energy
eroded internal energy	stonewall energy
eroded hourglass energy	rigid body stopper energy
added mass	percentage [mass] increase

JNTFORC	
x, y, z force	x, y, z moment

MATSUM	
kinetic energy	x, y, z rigid body velocity
internal energy	eroded internal energy
hourglass energy	eroded kinetic energy
x, y, z momentum	added mass

NCFORC
X force
Y force
Z force

NODOUT
x, y, z displacement
X, y, z velocity
X, y, z acceleration
X, y, z rotation
X, y, z rotational velocity
X, y, z rotation acceleration

NODFOR
X, y. z force

PRTUBE
cross section area pressure velocity density

PLLYOUT
adjacent beam IDs
slip
slip rate
resultant force
wrap angle

RBDOUT
x, y, z displacement
x, y, z velocity
x, y, z acceleration

RCFORC
x, y, z force
Mass of nodes in contact

RWFORC
normal
x, y, z force

SECFORC
x, y, z force
x, y, z moment
x, y, z center
area
resultant force

SLEOUT
slave energy
master energy
frictional energy

SPCFORC
x, y, z force
x, y, z moment

SWFORC
axial force
shear force
failure function
weld length
resultant moment
torsion

SPHOUT
xx, yy, zz stress
xy, yz, zx stress
density
number of neighbors
xx, yy, zz strain
xy, yz, zx strain
half of smoothing length
plastic strain
particle active state
effective stress
temperature
xx,yy,zz strain rate
xy,yz,zx strain rate
SPH to SPH coupling forces

Remarks:

1. **Discrepancies Between "matsum" and "glstat" Output.** The kinetic energy quantities in the matsum and glstat files may differ slightly in values for several reasons. First, the energy associated with added mass (from mass-scaling) is included in the glstat calculation, but is not included in matsum. Secondly, the energies are computed element by element in matsum for the deformable mate-

rials and, consequently, nodes which are merged with rigid bodies will also have their kinetic energy included in the rigid body total. Furthermore, kinetic energy is computed from nodal velocities in glstat and from element midpoint velocities in matsum.

2. **PRINT Keyword Option on *PART.** The PRINT option in the part definition allows some control over the extent of the data that is written into the matsum and rbdout files. If the print option is used the variable PRBF can be defined such that the following numbers take on the meanings:

   EQ.0: default is taken from the keyword *CONTROL_OUTPUT,

   EQ.1: write data into rbdout file only,

   EQ.2: write data into matsum file only,

   EQ.3: do not write data into rbdout and matsum.

   Also see CONTROL_OUTPUT and PART_PRINT.

3. **The Restart Feature.** This keyword is also used in the restart phase, see *RESTART. Thus, the output interval can be changed when restarting.

4. **LS-PrePost.** All information in the files except in AVSFLT, MOVIE, and MPGS can also be plotted using LS-PrePost. Arbitrary cross plotting of results between ASCII files is easily handled.

5. **The "rcforc" File.** Resultant contact forces reported in rcforc are averaged over the preceding output interval.

6. **Spring and Damper Energy.** "Spring and damper energy" reported in glstat is a subset of "Internal energy". The "Spring and damper energy" includes internal energy of discrete elements, seatbelt elements, and that associated with joint stiffness. (see *CONSTRAINED_JOINT_STIFFNESS_…)

7. **OPTION*n* Field for "elout".** OPTION1, OPTION2, OPTION3, and OPTION4 give the number of additional history variables output for the integrated solids, shells, thick shells, and beams, respectively. Within this special option, each integration point is printed with its corresponding history data. No integration points are averaged. This is different than the default output where the stress data within a shell ply of a fully integrated shell, for example, are averaged and then written as output. The primary purpose of this database extension is to allow the actual integration point stress data and history variable data to be checked. There are no transformations applied to either the output stresses or history data.

8. **The Failure Function.** The failure function reported to the DCFAIL database is set to zero when the weld fails. If damage is active, then it is set to the negative of the damage scale factor which goes from 1 to 0 as damage grows.

9. **OPTION*n* Field for "bndout".** For the bndout file, OPTION1 controls the nodal force group output, OPTION2 controls the concentrated force output, OPTION3 controls the pressure boundary condition output, and OPTION4 controls the velocity/displacement/acceleration nodal boundary conditions. If the value is 0 or left blank, the category is included (the default), and if it is 1, the category is not included in the bndout file.

10. **Contents of "glstat".** The glstat table above includes all items that *may* appear in the glstat data. The items that are actually written depend on the contents of the input deck. For example, hourglass energy appears only if HGEN = 2 in *CONTROL_ENERGY and added mass only appears if DT2MS < 0 in *CONTROL_TIMESTEP.

11. **Element ID Controlling the Time Step.** The element ID controlling the time step is included in the glstat data but is not read by LS-PrePost. If the element ID is of interest to the user, the ASCII version of the glstat file can be opened with a text editor.

12. **The FILTER Option.** The FILTER option uses a Butterworth filter for the forward, single pass filtering and the backward, double pass filtering options. The forward filtered output $Y(n)$ at sampling interval n is obtained from the solution value $X(n)$ using the formula

$$Y(n) = a_0X(n) + a_1X(n-1) + a_2X(n-2) + b_1Y(n-1) + b_2Y(n-2)$$

where the coefficients are

$$\begin{aligned}
\omega_d &= 2\pi\left(\frac{C}{0.6}\right)1.25 \\
\omega_a &= \tan(\omega_a T/2) \\
a_0 &= \omega_a^2/(1+\sqrt{2}\omega_a+\omega_a^2) \\
a_1 &= 2a_0 \\
a_2 &= a_0 \\
b_1 &= 2(1-\omega_a^2)/\left(1+\sqrt{2}\omega_a+\omega_a^2\right) \\
b_2 &= \left(-1+\sqrt{2}\omega_a-\omega_a^2\right)/\left(1+\sqrt{2}\omega_a+\omega_a^2\right)
\end{aligned}$$

The two previous solution values and filtered values at $n-1$ and $n-2$ are stored.

Backward filtering improves the phase response of the filtered output. It is performed according to the formula

$$Z(n) = a_0 Y(n) + a_1 Y(n+1) + a_2 Y(n+2) + b_1 Z(n+1) + b_2 Z(n+2)$$

where $Z(n)$ is the backward filtered value at sample time n. This implies that all the forward filtered values $Y(n)$ are stored during the analysis, and that would require a prohibitive amount of memory. To limit the amount of memory required, the forward filtered values at stored for the time interval W, where the number of stored states is W/T, and the backward filtering is applied starting at the last saved value of the forward filtered values. As the window width increases, the filtered values approach the values that would be obtained from storing all of the forward filtered values.

The results of the backward filtering are sensitive to the window width, and experimentation with the width is necessary to obtain good results with the minimum window width. A window width of at least 10 to 15 times the sample rate T should be used as a starting point. Some applications may require a window width that is much larger. The required window width decreases as the cut-off frequency increases. Or, to put it another way, the window width must be increased to make the filtered output smoother.

As an example, a random series of numbers between 0 and 1 was generated and filtered at intervals of 0.1 milliseconds with cut-off frequencies from 60 Hz to 420 Hz. The reverse filtering was applied with various window widths to determine how many forward filtered states must be saved to achieve fixed levels of accuracy compared to complete reverse filtering from the last state to the first state. The results are shown in the table below. Note that the error is calculated *only* for the first state and the numbers being filtered are random. *This example should only be used as a very rough guide that indicates the overall trends and not as a recommendation for specific problems.*

Cut-off Frequency	No. of States 50% Error	No. of States 25% Error	No. of States 10% Error	No. of States 5% Error	No. of States 1% Error
60 Hz	26	33	55	68	87
120 Hz	13	16	30	37	44
180 Hz	8	10	22	26	30
240 Hz	6	8	17	19	23
300 Hz	5	6	12	15	18
360 Hz	5	6	10	12	16
420 Hz	4	5	9	10	15

*DATABASE_ALE

Purpose: For each ALE group (or material), this card controls the output for element time-history variables (in a tabular format that can be plotted in LS-PrePost by using the XYPlot button).

Card 1	1	2	3	4	5	6	7	8
Variable	DTOUT	SETID						
Type	F	I						
Default	none	none						

Variable Cards. Optional cards that can be used to add more variables with the volume fractions in the database (the volume fractions are always output). Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	VAR	VAR	VAR	VAR	VAR	VAR	VAR	VAR
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

VARIABLE **DESCRIPTION**

DTOUT Time interval between the outputs

SETID ALE element set ID.

If the model is 1D (*SECTION_ALE1D), the set should be *SET_BEAM

If the model is 2D (*SECTION_ALE2D), the set should be *SET_SHELL

If the model is 3D (*SECTION_SOLID), the set should be *SET_SOLID

VARIABLE	DESCRIPTION
VAR	Variable rank in the following list: EQ.1: xx-stress EQ.2: yy-stress EQ.3: zz-stress EQ.4: xy-stress EQ.5: yz-stress EQ.6: zx-stress EQ.7: plastic strain EQ.8: internal energy EQ.9: bulk viscosity EQ.10: previous volume EQ.11: pressure EQ.12: mass EQ.13: volume EQ.14: density EQ.15: kinetic energy If there is a blank column between 2 variable ranks, the list between these 2 ranks is selected. For example, if the card is as follows: `1, ,6` The 6 stresses are added to the database.

Remarks:

1. The .xy files are created when the termination time is reached or if one of the following switches (after pressing the keys Ctrl - C) stops the job: sw1, stop, quit. During the run, they can be created with the switch sw2.

2. The .xy files are created by element. There is a curve by ALE group (or material). A last curve can be added for volume averaged variables.

*DATABASE_ALE_MAT

Purpose: For each ALE group (or material), this card activates extra output for:

1. material volume: alematvol.xy,
2. material mass: alematmas.xy,
3. internal energy: alematEint.xy,
4. kinetic energy: alematEkin.xy,
5. and kinetic energy loss during the advection: alematEkinlos.xy.

These files are written in the ".xy" format, which LS-PrePost can plot with its "XYPlot" button.

Card	1	2	3	4	5	6	7	8
Variable	DTOUT	BOXLOW	BOXUP					
Type	F	I	I					
Default	none	0	0					

VARIABLE	DESCRIPTION
DTOUT	Time interval between the outputs
BOXLOW, BOXUP	Range of *DEFINE_BOX ids. BOXLOW is the lower bound for the range while BOXUP is the upper bound. The series of volumes covered by the specified range of *DEFINE_BOX determines the mesh regions for which ALE material data are to be output.

Remarks:

The ".xy" files are created at termination or if one of the following switches (Ctrl-C) is encountered: sw2, sw1, stop, quit.

*DATABASE_BINARY_*OPTION1*_*OPTION2*

This keyword is used to request binary output. See also *DATABASE_EXTENT_BINARY.

Choices for *OPTION1* are:

BLSTFOR Blast pressure database. See also *LOAD_BLAST_ENHANCED and Remark 3.

CPMFOR Corpuscular Particle Method interface force database. see Remark 2.

D3DRLF Dynamic relaxation database.

D3DUMP Database for restarts. Define output frequency in cycles.

D3PART Database for subset of parts. See also *DATABASE_EXTENT_BINARY and *DATABASE_EXTENT_D3PART.

D3PLOT Database for entire model. See also *DATABASE_EXTENT_BINARY.

D3PROP Database containing property data. See *DATABASE_BINARY_-D3PROP.

D3THDT Database containing time histories for subsets of elements and nodes. See *DATABASE_HISTORY. This database includes no geometry.

DEMFOR DEM interface force database. See Remark 5.

FSIFOR ALE interface force database. See Remark 1.

FSILNK ALE interface linking database. See Remark 4.

RUNRSF Database for restarts. Define output frequency in cycles.

INTFOR Contact interface database. Its file name must either be given using the FILE option or on the execution line using "S=". Also see *CONTACT variables SPR and MPR.

PBMFOR Particle Blast Method interface force database.

D3CRACK Option to control output interval for ASCII “aea_crack” file for the Winfrith concrete model (*MAT_084/085). Oddly, this command does not control the output of the binary crack database for the Winfrith concrete model. The binary crack database is written when “q=” appears on the execution line and its output interval is taken from *DATABASE_BINARY_D3PLOT, It is used by LS-PrePost together with the D3PLOT database to display cracks in the deformed Winfrith concrete materials.

OPTION2 only applies when *OPTION1* is set to INTFOR and the only choice for *OPTION2* is FILE. *DATABASE_BINARY_INTFOR_FILE requires one extra line of input that specifies the name of the intfor database.

The D3DUMP and the RUNRSF options create complete databases which are necessary for restarts, see *RESTART. When RUNRSF is specified, the same file is overwritten after each interval, an option allows a series of files to be overwritten in a cyclic order. When D3DUMP is specified, a new restart file is created after each interva, thus a "family" of files is created numbered sequentially, e.g., d3dump01, d3dump02, etc. The default file names are runrsf and d3dump unless other names are specified on the execution line, see the INTRODUCTION, EXECUTION SYNTAX. Since all data held in memory is written into the restart files, these files can be quite large and care should be taken with the d3dump files not to create too many. If *DATABASE_BINARY_D3PLOT is not specified in the keyword deck then the output interval for d3plot is automatically set to 1/20th the termination time.

The d3plot, d3part, d3drlf, and intfor databases contain histories of geometry and of state variables. Thus using these databases, one can, e.g., animate deformed geometry and plot time histories of element stresses and nodal displacements with LS-PrePost.

The d3thdt database contains no geometry but rather time history data for element subsets as well as global information, see *DATABASE_HISTORY. This data can be plotted with LS-PrePost. The intfor database does not have a default filename and one must be specified by adding s=filename to the execution line.

Similarly, for the fsifor database, a unique filename must be specified on the execution line with h=filename; see the INTRODUCTION, EXECUTION SYNTAX. The file structure is such that each file contains the full geometry at the beginning, followed by the analysis generated output data at the specified time intervals.

For the contents of the d3plot, d3part and d3thdt databases, see also the *DATABASE_-EXTENT_BINARY definition. It is possible to restrict the information that is dumped and consequently reduce the size of the databases. The contents of the d3thdt database are also specified with the *DATABASE_HISTORY definition. It should also be noted in particular that the databases can be considerably reduced for models with rigid bodies containing many elements.

FILE Card: Provide this card only for *DATABASE_BINARY_INTFOR_FILE.

FILE Card	1	2	3	4	5	6	7	8
Variable	FNAME							
Type	A80							
Default	none							

VARIABLE	DESCRIPTION
FNAME	Name of the database for the intfor data. S = filename on the execution line will override FNAME.

Card 1	1	2	3	4	5	6	7	8
Variable	DT/CYCL	LCDT/NR	BEAM	NPLTC	PSETID	CID		
Type	F	I	I	I	I	I		
Default	-	-	-	-	-	-		

D3PLOT Card. Additional Card for D3PLOT option.

Card 2	1	2	3	4	5	6	7	8
Variable	IOOPT	RATE	CUTOFF	WINDOW	TYPE	PSET		
Type	I	F	F	F	I	I		
Default	0	none	none	none	0	0		

VARIABLE **DESCRIPTION**

DT / CYL

This field defines the time interval between output states, DT, for all options except D3DUMP, RUNRSF, and D3DRLF.

For D3DUMP, RUNRSF, and D3DRLF options the first field contains CYCL instead of DT. These databases are updated every CYCL convergence checks during the explicit dynamic relaxation phase.

NR

Number of RUNning ReStart Files, runrsf, written in a cyclical fashion. The default is 1, i.e., only one runrsf file is created and the data therein is overwritten each time data is output.

LCDT

Optional load curve ID specifying time interval between dumps. This variable is only available for options D3PLOT, D3PART, D3THDT, INTFOR and BLSTFOR.

VARIABLE **DESCRIPTION**

BEAM

Discrete element option flag (*DATABASE_BINARY_D3PLOT only).

EQ.0: Discrete spring and damper elements are added to the d3plot database where they are displayed as beam elements. The discrete elements' global x, global y, global z and resultant forces (moments), and change in length (rotation) are written to the database where LS-PrePost (incorrectly) labels them as though they were beam quantities, i.e., axial force, S-shear resultant, T-shear resultant, etc.

EQ.1: No discrete spring, damper and seatbelt elements are added to the d3plot database. This option is useful when translating old LS-DYNA input decks to KEYWORD input. In older input decks there is no requirement that beam and spring elements have unique ID's, and beam elements may be created for the spring and dampers with identical ID's to existing beam elements causing a fatal error. However, this option comes with some limitations and, therefore, should be used with caution.

1. Contact interfaces which are based on part IDs of seatbelt elements will not be properly generated if this option is used.

2. DEFORMABLE_TO_RIGID will not work if PID refers to discrete, damper, or seatbelt elements.

EQ.2: Discrete spring and damper elements are added to the d3plot database where they are displayed as beam elements (similar to option 0). In this option the element resultant force is written to its first database position allowing beam axial forces and spring resultant forces to be plotted at the same time. This can be useful during some post-processing applications.

This flag, set in *DATABASE_BINARY_D3PLOT, also affects the display of discrete elements in several other databases such as d3drlf, d3part.

NPLTC

DT = ENDTIME/NPLTC. Applies to D3PLOT and D3PART options only. This overrides the DT specified in the first field.

CID

Coordinate system ID for FSIFOR and FSILNK, see *DEFINE_COORDINATE_SYSTEM.

VARIABLE **DESCRIPTION**

PSETID

Part set ID for D3PART and D3PLOT options only. See *SET_-PART. Parts in PSETID will excluded in the d3plot database. Onlyparts in PSETID are included in the d3part database.

IOOPT

This variable applies to the D3PLOT option only. Flag to govern behavior of the plot frequency load curve defined by LCDT:

EQ.1: At the time each plot is generated, the load curve value is added to the current time to determine the next plot time (this is the default behavior).

EQ.2: At the time each plot is generated, the next plot time T is computed so that T = the current time plus the load curve value at time T.

EQ.3: A plot is generated for each abscissa point in the load curve definition. The actual value of the load curve is ignored.

RATE

Time interval *T* between filter sampling.

CUTOFF

Frequency cut-off *C* in Hz.

WINDOW

The width of the window *W* in units of time for storing the single, forward filtering required for the TYPE = 2 filter option. Increasing the width of the window will increase the memory required for the analysis. A window that is too narrow will reduce the amplitude of the filtered result significantly, and values below 15 are not recommended for that reason. In general, the results for the TYPE = 2 option are sensitive to the width of the window and experimentation is required.

TYPE

Flag for filtering options.

EQ.0: No filtering (default).

EQ.1: Single pass, forward Butterworth filtering.

EQ.2: Two pass filtering over the specified time window. Backward Butterworth filtering is applied to the forward Butterworth results that have been stored. This option improves the phase accuracy significantly at the expense of memory.

VARIABLE	DESCRIPTION
PSET	Part set ID for filtering. If no set is specified, all parts are included. For each element integration point in the d3plot file, 24 words of memory are required in LS-DYNA for the single pass filtering, and more for the two pass filtering. Specifying PSET is recommended to minimize the memory requirements.

Remarks:

1. **FSIFOR.** *DATABASE_BINARY_FSIFOR only applies to models having penalty-based coupling between Lagrangian and ALE materials (CTYPE=4 or 5 in the coupling card, *CONSTRAINED_LAGRANGE_IN_SOLID). When *DATABASE_FSI is defined, a few pieces of coupling information of some Lagrangian surface entities interacting with the ALE materials may be output as history parameters into a file called "dbfsi". Coupling pressure is one of the output variables. However, this coupling pressure is averaged over the whole surface entity being monitored. To obtain coupling pressure contour plot as a function of time over the coupled surface, a user can define the *DATABASE_-BINARY_FSIFOR keyword. To use it, three things must be done:

   a) The INTFORC parameter (*CONSTRAINED_LAGRANGE_IN_SOLID, 4th row, 3rd column) must be turned ON (INTFORC = 1).

   b) A *DATABASE_BINARY_FSIFOR card is defined controlling the output interval. The time interval between output is defined by the parameter DT in this card.

   c) This interface force file is activated by executing LS-DYNA as follows:

      ```
      lsdyna i=inputfilename.k ... h=interfaceforcefilename
      ```

      LS-DYNA will then write out the segment coupling pressure and forces to a binary interface force file for contour plotting over the whole simulation interval.

      To plot the binary data in this file, type: lsprepost interfaceforcefilename.

      For example, when all 3 of the above actions are taken, and assuming "h" is set to "fsifor", then a series of "fsifor##" binary files are output for contour plotting. To plot this, type "lsprepost fsifor" (without the double quotes).

2. **CPMFOR.** *DATABASE_BINARY_CPMFOR applies to models using *AIRBAG_PARTICLE feature which controls the output interval of CPM interface force file. This interface force file is activated by executing LS-DYNA with command line option (cpm=).

   lsdyna i=**inputfilename.k** … cpm=**interfaceforce_filename**

   CPM interface force file stores segment's coupling pressure and forces. The coupling pressure is averaged over each segment without considering the effect of ambient pressure, P_{atm}.

3. **BLSTFOR.** The BLSTFOR database is not available for two dimensional axisymmetric analysis.

4. **FSILNK.** The *DATABASE_BINARY_FSILNK variant writes the selected *CONSTRAINED_LAGRANGE_IN_SOLID interface's segment pressure to the fsilink file for the next analysis without ALE meshes.

   lsdyna i=**inputfilename.k** … fsilink=**filename**

5. **DEMFOR.** *DATABASE_BINARY_DEMFOR applies to models using DEM coupling option *DEFINE_DE_TO_SURFACE_COUPLING. This card will control the output interval of DEM interface force file. This interface force file is activated by LS-DYNA command line option (dem=).

   lsdyna i=**inputfilename.k** … dem=**interfaceforce_filename**

   DEM interface force file stores segment's coupling pressure and forces.

6. **PBMFOR.** *DATABASE_BINARY_PBMFOR applies to models using *PARTICLE_BLAST feature which controls the output interval of PBM interface force file. This interface force file is activated by executing LS-DYNA with command line option (pbm=).

   lsdyna i=**inputfilename.k** … pbm=**interfaceforce_filename**

   PBM interface force file stores segment's coupling pressure and forces.

*DATABASE_BINARY_D3PROP

Purpose: This card causes LS-DYNA to add the part, material, equation of state, section, and hourglass data to the first d3plot file or else write the data to a separate database d3prop. Rigidwall data can also be included. LS-PrePost does not read the additional data so use of this command is of dubious benefit.

Card 1	1	2	3	4	5	6	7	8
Variable	IFILE	IMATL	IWALL					
Type	I	I	I					
Default	1	0	0					

VARIABLE	DESCRIPTION
IFILE	Specify file for d3prop output. (This can also be defined on the command line by adding d3prop = 1 or d3prop = 2 which also sets IMATL = IWALL = 1) EQ.1: Output data at the end of the first d3plot file. EQ.2: Output data to the file d3prop.
IMATL	Output *EOS, *HOURGLASS, *MAT, *PART and *SECTION data. EQ.0: No EQ.1: Yes
IWALL	Output *RIGIDWALL data. EQ.0: No EQ.1: Yes

.

*DATABASE_CPM_SENSOR

Purpose: This card activates an ASCII file "cpm_sensor". Its input defines sensors' locations based on the positions of some Lagrangian segments. The output gives the history of the velocity, temperature, density and pressure averaged on the number of particles contained in the sensors. This card is activated only when the *AIRBAG_PARTICLE card is used.

Card 1	1	2	3	4	5	6	7	8
Variable	DT	BINARY						
Type	F	I						

Sensor Definition Cards. Each card defines one sensor. This card may be repeated to define multiple sensors. Input stops when the next "*" Keyword is found.

Card 2	1	2	3	4	5	6	7	8
Variable	SEGID	OFFSET	R/LX	LEN/LY	LZ			
Type	I	F	F	F	F			

VARIABLE	DESCRIPTION
DT	Output interval
BINARY	Flag for the binary file EQ.1: ASCII file is written, EQ.2: Data written to the binary file "binout", EQ.3: ASCII file is written and the data written to the binary file "binout".
SEGID	Segment set ID
OFFSET	Offset distance between the center of the sphere sensor and the segment center. If it is positive, Or, the distance between the base of the cylinder and the segment center while LENGTH is not zero. it is on the side pointed to by the segment normal vector. See remarks1 and 3.

VARIABLE	DESCRIPTION
R/LX	Radius(sphere)/length in local X direction(rectangular) of the sensor. See remarks 2 and 3.
LEN/LY	Length(cylinder)/length in local Y direction(rectangular) of the sensor.
LZ	Length in local Z direction(rectangular) of the sensor see remark 4

Remarks:

1. Each segment has a sensor. The distance between the segment center and the sensor center is defined by OFFSET (2nd parameter on the 2nd line) in the normal direction defined by the segment. This distance is constant: the sensor moves along with the segment.

2. The sensor is a sphere with a radius given by RADIUS (3rd parameter on the 2nd line).

3. OFFSET should be larger than RADIUS to prevent the segment from cutting the sphere. For cylindrical sensor, OFFSET is the distance from segment to the base of the cylinder.

4. For rectangular sensor, OFFSET is the distance from reference segment to the sensor. The sensor is defined using the segment's coordinates system. The base point is n1 and local X direction is along the vector n2 - n1. The local Z direction is the segment normal direction and local Y direction is constructed by local X and Z directions.

5. The output parameters in the "cpm_sensor" file are:

velx	=	x-velocity
vely	=	y-velocity
velz	=	z-velocity
velr	=	velocity
temp	=	temperature
dens	=	density
pres	=	pressure

These values are averaged on the number of particles in the sensor. RADIUS should be large enough to contain a reasonable number of particles for the averages.

```
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|.
$ INPUT:
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|.
*DATABASE_CPM_SENSOR
```

```
      0.01
$   SEGSID    OFFSET    RADIUS    LENGTH
       123       5.0       5.0
       124      -0.2       0.1
       125       0.7       0.6       1.0
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|..
$ The segment set id: 123 has 1 segment.
$ The segment set id: 123 has 1 segment.
$ The segment set id: 123 has 11 segments.
$ Each segment has an ID defined in D3HSP
$ The D3HSP file looks like the following:
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|..
Segments for sensor          1
  Sensor id        n1        n2        n3        n4
         1      3842      3843      3848      3847
 Segments for sensor          2
  Sensor id        n1        n2        n3        n4
         2      3947      3948      3953      3952

 Segments for sensor          3
  Sensor id        n1        n2        n3        n4
         3      3867      3868      2146      2145
         4      3862      3863      3868      3867
         5      3857      3858      3863      3862
         6      3852      3853      3858      3857
         7      3847      3848      3853      3852
         8      3837      3838      3843      3842
         9      3842      3843      3848      3847
        10      3832      3833      3838      3837
        11      3827      3828      3833      3832
        12      3822      3823      3828      3827
        13      1125      1126      3823      3822
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|..
```

*DATABASE_CROSS_SECTION_*OPTION1*_*{OPTION2}*

Option 1 includes:

PLANE

SET

To define an ID and heading for the database cross section use the option:

ID

Purpose: Define a cross section for resultant forces written to ASCII file secforc.

1. For the PLANE option, a set of two cards is required for each cross section. Then a cutting plane has to be defined, see Figure 14-1.
2. If the SET option is used, just one card is needed which identifies a node set and at least one element set. In this case the node set(s) defines the cross section, and the forces from the elements belonging to the element set(s) are summed up to calculate the section forces. Thus the element set(s) should include elements on only one side (not both sides) of the cross section.

The cross-section should cut through deformable elements only, not rigid bodies. Cutting through master segments for deformable solid element spot welds can lead to incorrect section forces since the constraint forces are not accounted for in the force and moment summations. Beam element modeling of welds do *not* require any special precautions.

ID Card. Additional card for ID keyword option.

Optional	1	2	3	4	5	6	7	8
Variable	CSID	HEADING						
Type	I	A70						

The heading is picked up by some of the peripheral LS-DYNA codes to aid in post-processing.

VARIABLE	DESCRIPTION
CSID	Cross section ID. This must be a unique number.
HEADING	Cross section descriptor. It is suggested that unique descriptions be used.

Plane Card 1. First additional card for PLANE keyword option.

Card 1	1	2	3	4	5	6	7	8
Variable	PSID	XCT	YCT	ZCT	XCH	YCH	ZCH	RADIUS
Type	I	F	F	F	F	F	F	F
Default	0	0.	0.	0.	0.	0.	0.	0.

Plane Card 2. Second additional card for PLANE keyword option.

Card 2	1	2	3	4	5	6	7	8
Variable	XHEV	YHEV	ZHEV	LENL	LENM	ID	ITYPE	
Type	F	F	F	F	F	I	I	
Default	0.	0.	0.	infinity	infinity	global	0	

The set option requires that the equivalent of the automatically generated input by the cutting plane capability be identified manually and defined in sets. All nodes in the cross-section and their related elements that contribute to the cross-sectional force resultants must be defined.

Set Card. Additional Card for the SET keyword option.

Card 1	1	2	3	4	5	6	7	8
Variable	NSID	HSID	BSID	SSID	TSID	DSID	ID	ITYPE
Type	I	I	I	I	I	I	I	I
Default	required	0	0	0	0	0	global	0

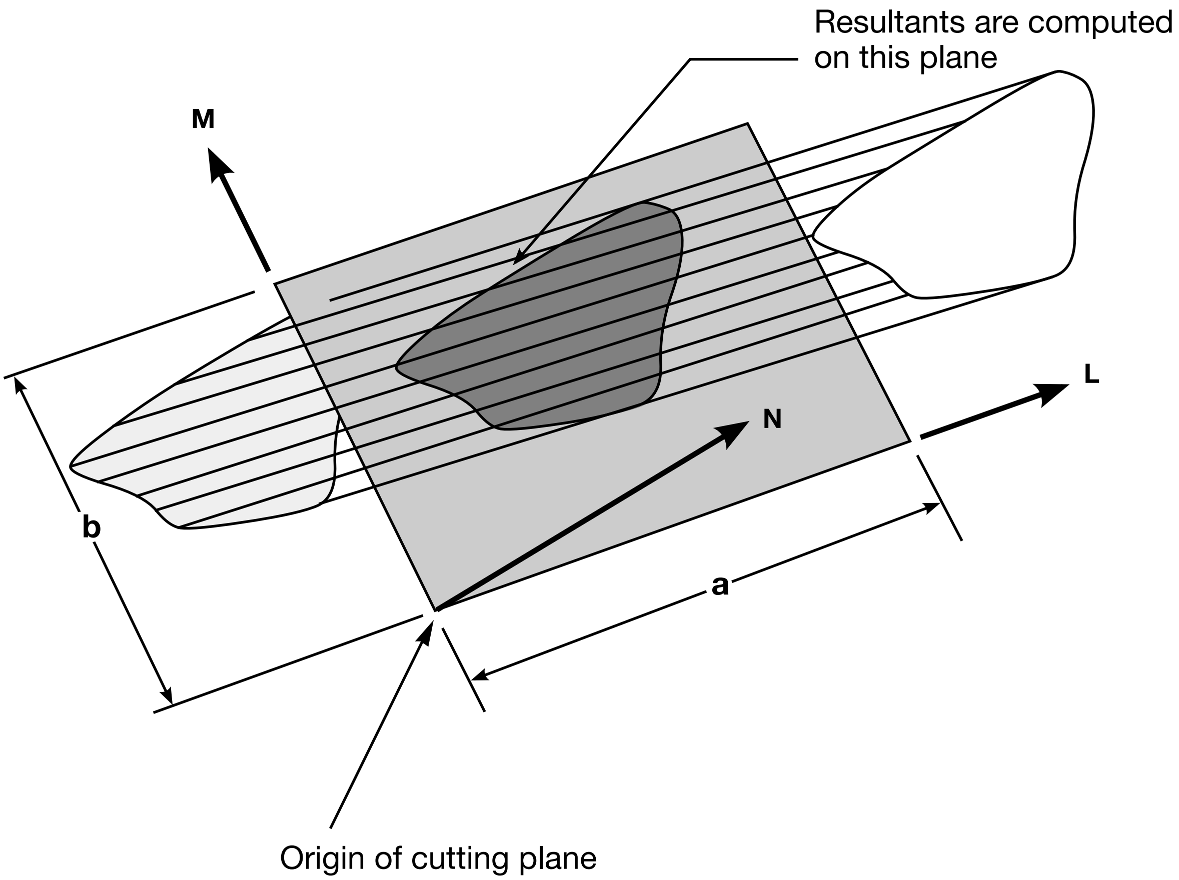


Figure 14-1. Definition of cutting plane for automatic definition of interface for cross-sectional forces. The automatic definition does not check for springs and dampers in the section. For best results the cutting plane should cleanly pass through the middle of the elements, distributing them equally on either side. Elements that intersect the edges of the cutting plane are deleted from the the cross-section.

VARIABLE	DESCRIPTION
CSID	Optional ID for cross section. If not specified cross section ID is taken to be the cross section order in the input deck.
PSID	Part set ID. If zero all parts are included.
XCT	*x*-coordinate of tail of any outward drawn normal vector, **N**, originating on wall (tail) and terminating in space (head), see Figure 14-1.
YCT	*y*-coordinate of tail of normal vector, **N**.
ZCT	*z*-coordinate of tail of normal vector, **N**.
XCH	*x*-coordinate of head of normal vector, **N**.

VARIABLE	DESCRIPTION
YCH	*y*-coordinate of head of normal vector, **N**.
ZCH	*z*-coordinate of head of normal vector, **N**.
RADIUS	Optional radius. If a radius is set (radius ≠ 0), then circular cut plane centered at (XCT, YCT ,ZCT) of radius = RADIUS, with the normal vector originating at (XCT, YCT, ZCT) and pointing towards (XCH, YCH, ZCH) will be created. In this case the variables XHEV, YHEV, ZHEV, LENL, and LENM, which are defined on the 2^{nd} card will be ignored.
XHEV	*x*-coordinate of head of edge vector, **L**.
YHEV	*y*-coordinate of head of edge vector, **L**.
ZHEV	*z*-coordinate of head of edge vector, **L**.
LENL	Length of edge *a*, in **L** direction.
LENM	Length of edge *b*, in **M** direction.
NSID	Nodal set ID, see *SET_NODE_*OPTION*.
HSID	Solid element set ID, see *SET_SOLID.
BSID	Beam element set ID, see *SET_BEAM.
SSID	Shell element set ID, see *SET_SHELL_*OPTION*.
TSID	Thick shell element set ID, see *SET_TSHELL.
DSID	Discrete element set ID, see *SET_DISCRETE.
ID	Rigid body (see *MAT_RIGID, type 20), accelerometer ID (see *ELEMENT_SEATBELT_ACCELEROMETER), or coordinate ID, see *DEFINE_COORDINATE_NODES. The force resultants are output in the updated local system of the rigid body or accelerometer. For ITYPE = 2, the force resultants are output in the updated local coordinate system if FLAG = 1 in *DEFINE_COORDINATE_NODES or if NID is nonzero in *DEFINE_COORDINATE_VECTOR.

VARIABLE	DESCRIPTION
ITYPE	Flag that specifies whether ID above pertains to a rigid body, an accelerometer, or a coordinate system. EQ.0: rigid body, EQ.1: accelerometer, EQ.2: coordinate system.

DATABASE_EXTENT_*OPTION

Available options include:

AVS

BINARY

D3PART

INTFOR

MOVIE

MPGS

SSSTAT

Purpose: Control to some extent the content of specific output databases.

The BINARY option of *DATABASE_EXTENT applies to the binary databases d3plot, d3thdt, and d3part. In the case of the d3part database, variables set using the D3PART option will override the corresponding variables of the BINARY option. See also *DATABASE_BINARY_*OPTION*.

The AVS, MOVIE, and MPGS databases will be familiar to users that have a use for those databases.

*DATABASE_EXTENT_AVS

This command controls content written to the avsflt database. See AVSFLT option to *DATABASE card.

Varriable Cards. Define as many cards as needed. Input ends at next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	VTYPE	COMP						
Type	I	I						

VARIABLE **DESCRIPTION**

VTYPE Variable type:

EQ.0: node,

EQ.1: brick,

EQ.2: beam,

EQ.3: shell,

EQ.4: thick shell.

COMP Component ID. For the corresponding VTYPE, integer components from the following tables can be chosen:

VTYPE.EQ.0: Table 10.1,

VTYPE.EQ.1: Table 10.2,

VTYPE.EQ.2: not supported,

VTYPE.EQ.3: Table 10.3,

VTYPE.EQ.4: not supported.

Remarks:

The AVS database consists of a title card, then a control card defining the number of nodes, brick-like elements, beam elements, shell elements, and the number of nodal vectors, NV, written for each output interval. The next NV lines consist of character strings that describe the nodal vectors. Nodal coordinates and element connectivity follow. For each state the solution time is written, followed by the data requested below. The last word in the file is the number of states. We recommend creating this file and examining its contents, since the organization is relatively transparent.

Table 14-2. Nodal Quantities

Component ID	Quantity
1	x, y, z-displacements
2	x, y, z-velocities
3	x, y, z-accelerations

Table 14-3. Brick Element Quantities

Component ID	Quantity
1	x-stress
2	y-stress
3	z-stress
4	xy-stress
5	yz-stress
6	zx-stress
7	effective plastic strain

Table 14-4. Shell and Thick Shell Element Quantities

Component ID	Quantity
1	midsurface x-stress
2	midsurface y-stress
3	midsurface z-stress
4	midsurface xy-stress
5	midsurface yz-stress
6	midsurface xz-stress
7	midsurface effective plastic strain
8	inner surface x-stress
9	inner surface y-stress
10	inner surface z-stress
11	inner surface xy-stress
12	inner surface yz-stress
13	inner surface zx-stress
14	inner surface effective plastic strain
15	outer surface x-stress
16	outer surface y-stress
17	outer surface z-stress
18	outer surface xy-stress
19	outer surface yz-stress
20	outer surface zx-stress
21	outer surface effective plastic strain
22	bending moment-mxx (4-node shell)
23	bending moment-myy (4-node shell)
24	bending moment-mxy (4-node shell)
25	shear resultant-qxx (4-node shell)
26	shear resultant-qyy (4-node shell)
27	normal resultant-nxx (4-node shell)
28	normal resultant-nxx (4-node shell)
29	normal resultant-nxx (4-node shell)

Component ID	Quantity
30	thickness (4-node shell)
31	element dependent variable
32	element dependent variable
33	inner surface x-strain
34	inner surface y-strain
35	inner surface z-strain
36	inner surface xy-strain
37	inner surface yz-strain
38	inner surface zx-strain
39	outer surface x-strain
40	outer surface y-strain
41	outer surface z-strain
42	outer surface xy-strain
43	outer surface yz-strain
44	outer surface zx-strain
45	internal energy
46	midsurface effective stress
47	inner surface effective stress
48	outer surface effective stress
49	midsurface max. principal strain
50	through thickness strain
51	midsurface min. principal strain
52	lower surface effective strain
53	lower surface max. principal strain
54	through thickness strain
55	lower surface min. principal strain
56	lower surface effective strain
57	upper surface max. principal strain
58	through thickness strain
59	upper surface min. principal strain

Component ID	Quantity
60	upper surface effective strain

Table 14-5. Beam Element Quantities

Component ID	Quantity
1	x-force resultant
2	y-force resultant
3	z-force resultant
4	x-moment resultant
5	y-moment resultant
6	z-moment resultant

DATABASE_EXTENT_BINARY_*{OPTION}

Purpose: Control to some extent the content of binary output databases d3plot, d3thdt, and d3part. See also *DATABASE_BINARY_*OPTION* and *DATBASE_EXTENT_D3-PART. The content of the binary output database intfor may be modified using *DATA-BASE_EXTENT_INTFOR. The option COMP controls to the content of binary output databases d3plot and d3eigv. When the option COMP is used, it will suppress most of settings in *DATABASE_EXTENT_BINARY.

Available options include:

<BLANK>

COMP

If no option is specified, use the following cards:

Card 1	1	2	3	4	5	6	7	8
Variable	NEIPH	NEIPS	MAXINT	STRFLG	SIGFLG	EPSFLG	RLTFLG	ENGFLG
Type	I	I	I	I	I	I	I	I
Default	0	0	3	0	1	1	1	1
Remarks			1	10				

Card 2	1	2	3	4	5	6	7	8
Variable	CMPFLG	IEVERP	BEAMIP	DCOMP	SHGE	STSSZ	N3THDT	IALEMAT
Type	I	I	I	I	I	I	I	I
Default	0	0	0	1	1	1	2	1
Remarks			2					

Remaining cards are optional.[†]

Card 3	1	2	3	4	5	6	7	8
Variable	NINTSLD	PKP_SEN	SCLP	HYDRO	MSSCL	THERM	INTOUT	NODOUT
Type	I	I	F	I	I	I	A	A
Default	1	0	1.0	0	0	0	none	none
Remarks							4	4

Card 4	1	2	3	4	5	6	7	8
Variable	DTDT	RESPLT	NEIPB					
Type	I	I	I					
Default	1	0	0					

For COMP option, use Card 1 below (no Cards 2-4)

Card 1	1	2	3	4	5	6	7	8
Variable	IGLB	IXYZ	IVEL	IACC	ISTRS	ISTRA	ISED	
Type	I	I	I	I	I	I	I	
Default	0	0	0	0	0	0	0	
Remarks								

VARIABLE	DESCRIPTION
NEIPH	Number of additional integration point history variables written to the binary databases (d3plot, d3part, d3drlf) for solid elements and SPH particles. The integration point data is written in the same order that it is stored in memory-each material model has its own history variables that are stored. For user defined materials it is important to store the history data that is needed for plotting before the data which is not of interest. See also *DEFINE_MATERIAL_HISTORIES. For output of additional integration point history variables for solid elements to the elout database, see the variable OPTION1 in *DATABASE_ELOUT.
NEIPS	Number of additional integration point history variables written to the binary databases (d3plot, d3part, d3drlf) for both shell and thick shell elements for each integration point, see NEIPH above and *DEFINE_MATERIAL_HISTORIES. For output of additional integration point history variables for shell and thick shell elements to the elout database, see the variables OPTION2 and OPTION3, respectively, in *DATABASE_ELOUT.

VARIABLE	DESCRIPTION
MAXINT	Number of shell and thick shell through-thickness integration points for which output is written to d3plot. This does not apply to strain tensor output flagged by STRFLG.

MAXINT (def = 3)	number of Integration Points	Description
3	> 3 (even & odd)	results are output for the outermost (top) and innermost (bottom) integration points together with results for the neutral axis.
3	1	All three results are identical.
> 3	≤ MAXINT	Results for the first MAXINT integration points in the element will be output.
≠ 3	Even	See above. This will *exclude* mid-surface results, whereas when MAXINT = 3 mid-surface results are calculated and reported.
< 0	Any	MAXINT integration points are output for each in plane integration point location and no averaging is used. This can greatly increase the size of the binary databases d3plot, d3thdt, and d3part.

See Remark 1 for more information.

VARIABLE	DESCRIPTION
STRFLG	STRFLG is interpreted digit-wise STRFLG = [NML], $$\mathrm{STRFLG} = L + M \times 10 + N \times 100$$ L.EQ.1: Write strain tensor data to d3plot and elout. For shell and thick shell elements two tensors are written, one at the innermost and one at the outermost integration point. For solid elements a single strain tensor is written. M.EQ.1: Write plastic strain data to d3plot. N.EQ.1: Write thermal strain data to d3plot. *Examples.* For STRFLG = 11 (011) LS-DYNA will write both strain and plastic strain tensors, but no thermal strain tensors. Whereas for STRFLG = 110, LS-DYNA will write plastic and thermal strain tensors but no strain tensors. For more information and supported elements and materials, see Remark 10.
SIGFLG	Flag for including the stress tensor for shells and solids. EQ.1: include (default), EQ.2: exclude for shells, include for solids. EQ.3: exclude for shells and solids.
EPSFLG	Flag for including the effective plastic strains for shells and solids. EQ.1: include (default), EQ.2: exclude for shells, include for solids. EQ.3: exclude for shells and solids.
RLTFLG	Flag for including stress resultants in the shell LS-DYNA database: EQ.1: include (default), EQ.2: exclude.
ENGFLG	Flag for including shell internal energy density and shell thickness. EQ.1: include (default), EQ.2: exclude.

VARIABLE DESCRIPTION

CMPFLG

Flag to indicate the coordinate system for output of stress and strain in solids, shells and thick shells comprised of orthotropic or anisotropic materials. See Remark 4.

EQ.-1: Same as 1, but for *MAT_FABRIC (forms 14 and -14) and *MAT_FABRIC_MAP the stress and strain is in engineering quantities instead of Green-Lagrange strain and 2nd Piola-Kirchhoff stress.

EQ.0: global coordinate system with exception of elout for shells (see EOCS in *CONTROL_OUTPUT).

EQ.1: local material coordinate system (as defined by AOPT and associated parameters in the *MAT input, and if applicable, by angles B1, B2, etc. in *SECTION_SHELL, *SECTION_TSHELL, or *PART_COMPOSITE, and by optional input in the *ELEMENT data). CMPFLG = 1 affects both d3plot and elout databases.

IEVERP

Every output state for the d3plot database is written to a separate file.

EQ.0: more than one state can be on each plot file,

EQ.1: one state only on each plot file.

BEAMIP

Number of beam integration points for output. This option does not apply to beams that use a resultant formulation. See Remark 2.

VARIABLE **DESCRIPTION**

DCOMP

Data compression to eliminate rigid body data:

EQ.1: off (default), no rigid body data compression,

EQ.2: on, rigid body data compression active,

EQ.3: off, no rigid body data compression, but all nodal velocities and accelerations are eliminated from the database.

EQ.4: on, rigid body data compression active and all nodal velocities and accelerations are eliminated from the database.

EQ.5: on, rigid body data compression active and rigid nodal data are eliminated from the database. Only 6 DOF rigid body motion is written.

EQ.6: on, rigid body data compression active, rigid nodal data, and all nodal velocities and accelerations are eliminated from the database. Only 6 DOF rigid body motion is written.

SHGE

Flag for including shell hourglass energy density.

EQ.1: off (default), no hourglass energy written,

EQ.2: on.

STSSZ

Flag for including shell element time step, mass, or added mass.

EQ.1: off (default),

EQ.2: output time step size,

EQ.3: output mass, added mass, or time step size.

See Remark 3 below.

N3THDT

Flag for including material energy in d3thdt database.

EQ.1: off, energy is NOT written to d3thdt database,

EQ.2: on (default), energy is written to d3thdt database.

IALEMAT

Output solid part ID list containing ALE materials.

EQ.1: on (default)

VARIABLE **DESCRIPTION**

NINTSLD

Number of solid element integration points written to the LS-DYNA database. When NINTSLD is set to 1 (default) or to any value other than 8, integration point values are averaged and only those averages are written output. To obtain values for individual integration points, set NINTSLD to 8, even if the multi-integration point solid has fewer than 8 integration points.

PKP_SEN

Flag to output the peak pressure and surface energy computed by each contact interface into the interface force database. To obtain the surface energy, FRCENG, must be sent to 1 on the control contact card. When PKP_SEN = 1, it is possible to identify the energies generated on the upper and lower shell surfaces, which is important in metal forming applications. This data is mapped after each H-adaptive remeshing.

EQ.0: No data is written

EQ.1: Output the peak pressures and surface energy by contact interface

SCLP

A scaling parameter used in the computation of the peak pressure. This parameter is generally set to unity (the default), but it must be greater than 0.

HYDRO

Either 3, 5 or 7 additional history variables useful to shock physics are output as the last history variables to d3plot (does not apply to elout). For HYDRO = 1, the internal energy per reference volume, the reference volume, and the pressure from bulk viscosity are added to the database, and for HYDRO = 2, the relative volume and current density are also added. For HYDRO = 4, two further variables are added: volumetric strain (defined as relative volume – 1.0), and Hourglass energy per unit initial volume.

MSSCL

Output nodal information related to mass scaling into the d3plot database. This option can be activated if and only if DT2MS < 0.0, see control card *CONTROL_TIMESTEP.

EQ.0: No data is written

EQ.1: Output incremental nodal mass

EQ.2: Output percentage increase in nodal mass

See Remark 3.

VARIABLE	DESCRIPTION
THERM	Output of thermal data to d3plot. The use of this option (THERM > 0) may make the database incompatible with other 3rd party software. EQ.0: (default) output temperature EQ.1: output temperature EQ.2: output temperature and flux EQ.3: output temperature, flux, and shell bottom and top surface temperature
INTOUT	Output stress/strain at all integration points for detailed element output in the ASCII file eloutdet. DT and BINARY of *DATABASE_ELOUT apply to eloutdet. See Remark 4. EQ.STRESS: when stress output is required EQ.STRAIN: when strain output is required EQ.ALL: when both stress and strain output are required
NODOUT	Output extrapolated stress/strain at connectivity nodes for detailed element output in the ASCII file eloutdet. DT and BINARY of *DATABASE_ELOUT apply to eloutdet. See Remark 4. EQ.STRESS: when stress output is required EQ.STRAIN: when strain output is required EQ.ALL: when both stress and strain output are required EQ.STRESS_GL: when nodal averaged stress output along the global coordinate system is required EQ.STRAIN_GL: when nodal averaged strain output along the global coordinate system is required EQ.ALL_GL: for global nodal averaged stress and strain output
DTDT	Output of node point Δtemperature/Δtime data to d3plot. EQ.0: (default) no output EQ.1: output $\Delta T/\Delta t$

VARIABLE **DESCRIPTION**

RESPLT

Output of translational and rotational residual forces to d3plot and d3iter.

EQ.0: No output

EQ.1: Output residual

NEIPB

Number of additional element or integration point history variables written to the binary databases (d3plot, d3part, d3drlf) for beam elements, see NEIPH above, BEAMIP and *DEFINE_MATERIAL_HISTORIES. See also Remark 12. For output of additional integration point history variables for beam elements to the elout database, see the variable OPTION4 in *DATABASE_-ELOUT.

IGLB

Output flag for global data

EQ.0: no
EQ.1: yes

IXYZ

Output flag for geometry data

EQ.0: no
EQ.1: yes

IVEL

Output flag for velocity data

EQ.0: no
EQ.1: yes

IACC

Output flag for acceleration data

EQ.0: no
EQ.1: yes

ISTRS

Output flag for stress data

EQ.0: no
EQ.1: yes

ISTRA

Output flag for strain data

EQ.0: no
EQ.1: yes

ISED

Output flag for strain energy density data

EQ.0: no
EQ.1: yes

Remarks:

1. **MAXINT Field.** If MAXINT is set to 3 then mid-surface, inner-surface and outer-surface stresses are output at the center of the element. For an even number of integration points, the points closest to the center are averaged to obtain the midsurface values. If multiple integration points are used in the shell plane, the stresses at the center of the element are found by computing the average of these points. For MAXINT equal to 3, LS-DYNA assumes that the data for the user defined integration rules are ordered from bottom to top even if this is not the case. If MAXINT is not equal to 3, then the stresses at the center of the element are output in the order that they are stored for the selected integration rule. If multiple points are used in plane the stresses are first averaged.

2. **BEAMIP Field.** Beam stresses are output if and only if BEAMIP is greater than zero. In this latter case the data that is output is written in the same order that the integration points are defined. The data at each integration point consists of the following five values for elastic-plastic Hughes-Liu beams: the normal stress, σ_{rr}; the transverse shear stresses, σ_{rs} and σ_{tr}; the effective plastic strain, and the axial strain which is logarithmic. For beams that are not elastic-plastic, the first history variable, if any, is output instead of the plastic strain. For the beam elements of Belytschko and his co-workers, the transverse shear stress components are not used in the formulation. No data is output for the Belytschko-Schwer resultant beam.

3. **Mass Scaling.** If mass scaling is active, the output of the time step size reveals little information about the calculation. If global mass scaling is used for a constant time step, the total element mass is output; however, if the mass is increased so that a minimum time step size is maintained (DT2MS is negative), the added mass is output. Also, see the control card *CONTROL_TIMESTEP.

4. **Output Coordinate System.** Output coordinate system used. When the parameters: INTOUT or NODOUT is set to STRESS, STRAIN, or ALL, the output coordinate system of the data, similar to the ASCII file elout, is determined by CMPFLG in *DATABASE_EXTENT_BINARY.

   a) When NODOUT is set to STRESS, STRAIN , or ALL. Each node of the element nodal connectivity will be output. See Example 1.

   b) Nodal output when NODOUT is set to STRESS_GL, STRAIN_GL, or ALL_GL. Averaged nodal results are calculated by summing up all contributions from elements sharing the common node, and then dividing the total by the number of contributing elements. Averaged nodal values are always output in the global coordinate system. See Example 2.

5. **Contents of eloutdet.** Available stress/strain components in eloutdet stress components includes 6 stress components (sig-*xx*, sig-*yy*, sig-*zz*, sig-*xy*, sig-*yz*,

sig-*zx*), yielding status, and effective plastic strain. Strain components includes 6 strain components

6. **Shell Element Output at Integration Points.** stresses at all integration points can be output. The strain at the top and bottom integration layer can be output. At a connective node the extrapolated stress and strain at the top and bottom layer can be output

7. **Thick Shells.** Thick shell element output includes the six stress components at each integration point. Strain at the top and bottom layer can be output. At the element node, values at the bottom layer are extrapolated to yield the values of nodes 1-4, and values at the top layer are extrapolated to yield values of nodes 5-8.

8. **Integration Point Locations.** Stresses and strain at all integration points can be output. The integration point order is as follows:

   a) point #1 is the point closest to node #1 in the connectivity array

   b) point #2 is the closest point to node #2, etc

   c) For tetrahedrons type 4, 16 and 17 with 5 integration points, point #5 is the midpoint.

   d) For the nodal points, values at the integration points are extrapolated.

9. **Reporting Residual Forces and Moments.** The output of residual forces and moments is supported for implicit and double precision only. With this option the forces and moments appear under the *Ndv* button in the fringe menu in LS-PrePost. The residual for rigid bodies is distributed to the slave nodes for the body without scaling for the purpose of capturing the complete residual vector.

10. **Calculation of Strains (STRFLG).** The strain tensor ε that are output to the d3plot database are calculated using proper time integration of the rate-of-deformation tensor **D**. More specifically, to assert objectivity of the resulting strain, it is for solids using a Jaumann rate of strain whereas for shells it uses the co-rotational strain rate. In mathematical terms the integration is using the following strain rates

$$\dot{\varepsilon} = \mathbf{D} - \varepsilon\mathbf{W} + \mathbf{W}\varepsilon \quad \text{(solids)}$$

$$\dot{\varepsilon} = \mathbf{D} - \varepsilon\boldsymbol{\Omega} + \boldsymbol{\Omega}\varepsilon \quad \text{(shells)}$$

where **W** is the spin tensor and $\boldsymbol{\Omega} = \dot{\mathbf{Q}}\mathbf{Q}^{\mathrm{T}}$ is the rotational velocity of the co-rotational system **Q** used for the shell element in question, taking into account invariant node numbering and such. This is to say that the resulting strains would be equal to the Cauchy stress for a hypo-elastic material (MAT_ELASTIC) with a Young's modulus of 1 and a Poisson's ratio of 0. This should be kept in mind when interpreting the results since they are not invariant to changes in element formulations and possibly nodal connectivities.

11. **Plastic and Thermal Strain (STRFLG).** The algorithm for writing plastic and thermal strains, which is also activated using STRFLG, is a modification of the algorithm used for mechanical strains (see Remark 10).

    a) For solids the element average strain in the global system having 6 components is written (local system if CMPFLG is set).

    b) For shells both plastic and thermal strains have 6 components. The thermal strain is written as a single tensor as in the solid case. The plastic strain output consists of 3 plane-averaged tensors: one for the bottom, one for the middle, and one for the top. For an even number of through thickness integration points, the middle is taken to be the average of the two integration points closest to the mid surface. Currently, only the following element/materials combinations are supported but other *will* be added upon request.

Thermal strain tensors			*Plastic strain tensors*		
Shells	Solids	Materials	Shells	Solids	Materials
2, 16, 23	1, 2	Add thermal expansion, 255	2, 16, 23	1, 2	24, 255

12. **History Variables for Beams (NEIPB).** In general, NEIPB follows the same conventions as NEIPH and NEIPS do for solid and shell elements and is supported in LS-PrePost v4.3 or later. Average, min and max values for each element are output, including data for resultant elements. If BEAMIP is nonzero, then element data is complemented with BEAMIP integration point values that can be examined individually. Beam history data is post-processed similarly to that of solid and shell element history data.

Example 1:

Excerpt from eloutdet file for a shell element with two through-thickness integration points and four in-plane integration points, with INTOUT = STRESS and NODOUT = STRESS:

```
element  materl
     ipt  stress  sig-xx   sig-yy   sig-zz   sig-xy   sig0yz   sig-zx   yield         location
       1-      1
   1- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  int.  point  1
   1- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  int.  point  2
   1- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  int.  point  3
   1- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  int.  point  4
   1- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  node        21
   1- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  node       22
   1- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  node       20
   1- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  node       19
   2- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  int.  point  1
   2- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  int.  point  2
   2- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  int.  point  3
   2- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  int.  point  4
   2- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  node       21
   2- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  node       22
   2- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  node       20
   2- 10 elastic  4.41E-2  2.51E-1  0.00E+0  7.76E-8  0.00E+0  0.00E+0  0.00E+0  node       19
```

Example 2:

Excerpt from eloutdet file for averaged nodal strain:

```
nodal strain calculations for time step   24  (at time 9.89479E+01 )
 node (global)
          strain   eps-xx     eps-yy         eps-zz   eps-xy       eps-yz      eps-zx
          1-
 lower surface     2.0262E-01 -2.6058E-02 -7.5669E-02 -5.1945E-03  0.0000E+00  0.0000E+00
 upper surface     2.0262E-01 -2.6058E-02 -7.5669E-02 -5.1945E-03  0.0000E+00  0.0000E+00
          2-
 lower surface     1.9347E-01  2.3728E-04 -8.3019E-02 -1.4484E-02  0.0000E+00  0.0000E+00
 upper surface     1.9347E-01  2.3728E-04 -8.3019E-02 -1.4484E-02  0.0000E+00  0.0000E+00
          3-
 lower surface     2.0541E-01 -5.7521E-02 -6.3383E-02 -1.7668E-03  0.0000E+00  0.0000E+00
 upper surface     2.0541E-01 -5.7521E-02 -6.3383E-02 -1.7668E-03  0.0000E+00  0.0000E+00
          ⋮            ⋮          ⋮            ⋮          ⋮            ⋮            ⋮
```

*DATABASE_EXTENT_D3PART

The following cards control content to the d3part binary database (Card 3 is optional). The parameters given here will supercede the corresponding parameters in *DATABASE_EXTENT_BINARY when writing the d3part binary database. See also *DATABASE_BINARY_D3PART which defines the output interval for d3part and the set of part included in d3part.

Card 1	1	2	3	4	5	6	7	8
Variable	NEIPH	NEIPS	MAXINT	STRFLG	SIGFLG	EPSFLG	RLTFLG	ENGFLG
Type	I	I	I	I	I	I	I	I
Default	0	0	3	0	1	1	1	1
Remarks			1					

Card 2	1	2	3	4	5	6	7	8
Variable		IEVERP			SHGE	STSSZ		
Type		I			I	I		
Default		0			0	0		

Card 3	1	2	3	4	5	6	7	8
Variable	NINTSLD							
Type	I							
Default	1							

VARIABLE **DESCRIPTION**

NEIPH

Number of additional integration point history variables written to the binary database for solid elements. The integration point data is written in the same order that it is stored in memory-each material model has its own history variables that are stored. For user defined materials it is important to store the history data that is needed for plotting before the data which is not of interest.

NEIPS

Number of additional integration point history variables written to the binary database for both shell and thick shell elements for each integration point, see NEIPH above.

MAXINT

Number of shell integration points written to the binary database, see also *INTEGRATION_SHELL. If the default value of 3 is used then results are output for the outermost (top) and innermost (bottom) integration points together with results for the neutral axis. If MAXINT is set to 3 and the element has 1 integration point then all three results will be the same. If a value other than 3 is used then results for the first MAXINT integration points in the element will be output. Note: If the element has an even number of integration points and MAXINT is not set to 3 then you will not get mid-surface results. See Remarks below. If MAXINT is set to a negative number, MAXINT integration points are output for each in plane integration point location and no averaging is used. This can greatly increase the size of the binary d3part database.

STRFLG

Set to 1 to dump strain tensors for solid, shell and thick shell elements for plotting by LS-PrePost and ASCII file elout. For shell and thick shell elements two tensors are written, one at the innermost and one at the outermost integration point. For solid elements a single strain tensor is written.

SIGFLG

Flag for including the stress tensor for shells.

EQ.1: include (default),

EQ.2: exclude.

EPSFLG

Flag for including the effective plastic strains for shells.

EQ.1: include (default),

EQ.2: exclude.

VARIABLE DESCRIPTION

RLTFLG

Flag for including stress resultants for shells.

EQ.1: include (default),

EQ.2: exclude.

ENGFLG

Flag for including shell internal energy density and shell thickness.

EQ.1: include (default),

EQ.2: exclude.

IEVERP

Every plot state for d3part database is written to a separate file. This option will limit the database to 1000 states:

EQ.0: more than one state can be on each plot file,

EQ.1: one state only on each plot file.

SHGE

Flag for including shell hourglass energy density.

EQ.1: off (default), no hourglass energy written,

EQ.2: on.

STSSZ

Flag for including shell element time step, mass, or added mass.

EQ.1: off (default),

EQ.2: output time step size,

EQ.3: output mass, added mass, or time step size.

See remark 3 below.

NINTSLD

Number of solid element integration points written. The default value is 1. For solids with multiple integration points NINTSLD may be set to 8. Currently, no other values for NINTSLD are allowed. For solids with multiple integration points, an average value is output if NINTSLD is set to 1.

*DATABASE_EXTENT_INTFOR

The following card controls to some extent the content of the optional intfor binary database. See also *DATABASE_BINARY_INTFOR. The intfor database contains geometry and time history data pertaining to those contact surfaces which are flagged in *CONTACT with the variables SPR and/or MPR. The name of the intfor database must be given on the execution line via "s=**filename**".

Card 1	1	2	3	4	5	6	7	8
Variable	NGLBV	NVELO	NPRESU	NSHEAR	NFORC	NGAPC	NFAIL	IEVERF
Type	I	I	I	I	I	I	I	I
Default	1	1	1	1	1	1	0	0

Optional Card.

Card 2	1	2	3	4	5	6	7	8
Variable	NWEAR	NWUSR	NHUF					
Type	I	I	I					
Default	0	0	0					

VARIABLE	DESCRIPTION
NGLBV	Output global variables: EQ.-1: no, EQ.1: yes (default).
NVELO	Output nodal velocity: EQ.-1: no, EQ.1: yes (default).

VARIABLE	DESCRIPTION
NPRESU	Output pressures: EQ.-1: no, EQ.1: normal interface pressure (default), EQ.2: normal interface pressure and peak pressure, EQ.3: normal interface pressure, peak pressure and time to peak.
NSHEAR	Output shear stresses: EQ.-1: no, EQ.1: shear stress in r-direction and s-direction (default).
NFORC	Output forces: EQ.-1: no, EQ.1: *x*-, *y*-, *z*-force at all nodes (default).
NGAPC	Output contact gaps at all nodes and surface energy density EQ.-1: no, EQ.1: yes (default).
NFAIL	Flag for display of deleted contact segments EQ.0: all segments are displayed, EQ.1: remove deleted contact segments from display.
IEVERF	Every interface force state for the "intfor" database is written to a separate file: EQ.0: more than one interface force state can be on each intfor file, EQ.1: one interface force output state only on each intfor file.
NWEAR	Output contact wear data, see *CONTACT_ADD_WEAR EQ.0: No output. GE.1: Output wear depth. GE.2: Output sliding distance.
NWUSR	Number of user wear history variables to output from user defined wear routines, see *CONTACT_ADD_WEAR.

VARIABLE	DESCRIPTION
NHUF	Number of user friction history variables to output from user defined friction routines, see *USER_INTERFACE_FRICTION (MPP only).

Remarks:

For gaps in Mortar contact, see NGAPC, these are measured with respect to the *nominal* contact surfaces of the two interacting segments. For instance, if IGNORE = 2 on *CONTACT_...MORTAR then an initial penetration d will dislocate the slave contact surface in the negative direction of the slave surface normal $\boldsymbol{n}$. The gap g reported to the intfor file is still measured between the master and slave surface neglecting this dislocation, thus only *physical* gaps are reported.

Wear outputs are governed by NWEAR and NWUSR, and requires the usage of a wear model associated with the contact interface. For NWEAR the "wear depth" (NWEAR.GE.1) and "sliding distance" (NWEAR.GE.2) are listed under the Nodal fringe menu in LS-PrePost. Following this, NWUSR user defined history variables are listed, corresponding to user wear history variables in a user wear routine. These are listed in the order that they are stored in the wear routine, see WTYPE.LT.0 on *CONTACT_ADD_WEAR.

*DATABASE_EXTENT_MOVIE

This keyword controls the content written to the BYU MOVIE databases. See movie option on *DATABASE manual entry.

Varriable Cards. Define as many cards as needed. Input ends at next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	VTYPE	COMP						
Type	I	I						

VARIABLE	DESCRIPTION
VTYPE	Variable type: EQ.0: node, EQ.1: brick, EQ.2: beam, EQ.3: shell, EQ.4: thick shell.
COMP	Component ID. For the corresponding VTYPE, integer components from the following tables can be chosen: VTYPE.EQ.0: Table 10.1 (see DATABASE_EXTENT_AVS), VTYPE.EQ.1: Table 10.2 (see DATABASE_EXTENT_AVS), VTYPE.EQ.2: not supported, VTYPE.EQ.3: Table 10.3 (see DATABASE_EXTENT_AVS), VTYPE.EQ.4: not supported.

.

*DATABASE_EXTENT_MPGS

Define as many cards as necessary. The created MPGS databases consist of a geometry file and one file for each output database. See MPGS option to *DATABASE keyword.

Card 1	1	2	3	4	5	6	7	8
Variable	VTYPE	COMP						
Type	I	I						

VARIABLE	DESCRIPTION
VTYPE	Variable type: EQ.0: node, EQ.1: brick, EQ.2: beam, EQ.3: shell, EQ.4: thick shell.
COMP	Component ID. For the corresponding VTYPE, integer components from the following tables can be chosen: VTYPE.EQ.0: Table 14-2 (see DATABASE_EXTENT_AVS), VTYPE.EQ.1: Table 14-3 (see DATABASE_EXTENT_AVS), VTYPE.EQ.2: not supported, VTYPE.EQ.3: Table 14-4 (see DATABASE_EXTENT_AVS), VTYPE.EQ.4: not supported.

DATABASE_EXTENT_SSSTAT_*OPTION

The only *OPTION* is:

ID

The ID option allows the definition of a heading which will be written at the beginning of the ASCII file ssstat.

Purpose: This command defines one or more subsystems. A subsystem is simply a set of parts, grouped for convenience. The ASCII output file ssstat provides histories of energy (kinetic, internal, hourglass) and momentum (x, y, and z) for each subsystem. The ssstat file is thus similar to glstat and matsum, but whereas glstat provides data for the whole model and matsum provides data for each individual part, ssstat provides data for each subsystem. The output interval for the ssstat file is given using *DATABASE_SSSTAT. To also include histories of subsystem mass properties in the ssstat file, use *DATABASE_SSSTAT_MASS_PROPERTIES.

For *DATABASE_EXTENT_BINARY without the ID option, the following card(s) apply. Define as many cards as necessary. Define one part set ID per subsystem, up to 8 subsystems per card.

Card 1	1	2	3	4	5	6	7	8
Variable	PSID1	PSID2	PSID3	PSID4	PSID5	PSID6	PSID7	PSID8
Type	I	I	I	I	I	I	I	I

For *DATABASE_EXTENT_BINARY_ID option, the following card(s) apply. Define as many cards as necessary. Define one part set ID per subsystem, 1 subsystem per card.

Card 1	1	2	3	4	5	6	7	8
Variable	PSID1	HEADING1						
Type	I	A70						

VARIABLE	DESCRIPTION
PSID*n*	Part set ID for subsystem *n*; see *SET_PART.

VARIABLE	DESCRIPTION
HEADING*n*	Heading for subsystem *n*.

*DATABASE_FATXML

Purpose: Process FATXML data. FATXML is an open, standardized data format based on the Extensible Markup Language (XML) which is developed by the German Research Association of Automotive Technology (Forschungsvereinigung Automobiltechnik - FAT). It is designed for consistent data management in the overall CAE process chain. A comprehensive explanation of the FATXML data format specification is given by Schulte-Frankenfeld and Deiters [2016].

LS-DYNA reads all lines between this keyword and the next keyword recognized by the star (*) sign, processes the data with respect to the include file structure and writes everything together in one output file called ‘d3plot.xml’.

Remarks:

It is intended that a corresponding FE model consists of one master file with several associated include files each containing a description by *DATABASE_FATXML data, usually at the end of the file. The master file loads the include files via *INCLUDE_TRANSFORM with potential offset values for nodes, elements, parts, etc. (IDNOFF, IDEOFF, IDPOFF, …). Finally, all data from different include files with different offsets are collected and then summarized in ‘d3plot.xml’. Since the resulting data format is public domain, Post-Processors are able to read that data and correlate it with the associated CAE model.

Example:

```
...
*DATABASE_FATXML
<?xml version=”1.0”?>
<CAE_META_DATA>
  < PART_ID NAME=”TestCase”>
    <PDM_DATA>
      ...
      <PDD_THICKNESS>
        < THICKNESS ID=”123”>1.0</THICKNESS >
        < THICKNESS ID=”124”>1.1</THICKNESS >
        ...
      </PDD_THICKNESS >
      ...
    </PDM_DATA >
  </PART_ID >
</CAE_META_DATA >
*END
```

*DATABASE_FORMAT

Purpose: Define the output format for binary files.

Card 1	1	2	3	4	5	6	7	8
Variable	IFORM	IBINARY						
Type	I	I						
Default	0	0						
Remarks	1	2						

VARIABLE **DESCRIPTION**

IFORM

Output format for d3plot and d3thdt files

EQ.0: LS-DYNA database format (default),

EQ.1: ANSYS database format,

EQ.2: Both LS-DYNA and ANSYS database formats.

IBINARY

Word size of the binary output files (d3plot, d3thdt, d3drlf and interface files for 64 bit computer such as CRAY and NEC.

EQ.0: default 64 bit format,

EQ.1: 32 bit IEEE format

Remarks:

1. The ANSYS output option is not available in MPP and is not universally available in SMP. The LS-DYNA banner in d3hsp will include “ANSYS database format” under the list of “Features enabled” if the option is available.

2. By using this option one can reduce the size of the binary output files which are created by 64 bits computer such as CRAY and NEC.

DATABASE_FREQUENCY_ASCII_*OPTION

Options for frequency domain ASCII databases with the default names given include:

NODOUT_SSD ASCII database for nodal results for SSD (displacement, velocity and acceleration). See also *FREQUENCY_DOMAIN_SSD.

ELOUT_SSD ASCII database for element results for SSD (stress and strain components). See also *FREQUENCY_DOMAIN_SSD.

Card 1	1	2	3	4	5	6	7	8
Variable	FMIN	FMAX	NFREQ	FSPACE	LCFREQ			
Type	F	F	I	I	I			
Default	0.0	0.0	0	0	0			

VARIABLE	DESCRIPTION
FMIN	Minimum frequency for output (cycles/time)
FMAX	Maximum frequency for output (cycles/time).
NFREQ	Number of frequencies for output.
FSPACE	Frequency spacing option for output: EQ.0: linear EQ.1: logarithmic EQ.2: biased
LCFREQ	Load Curve ID defining the frequencies for output.

Remarks:

1. The keyword defines output frequencies for NODOUT_SSD and ELOUT_SSD, and they can be different from output frequencies for D3SSD (which is defined by keyword *DATABASE_FREQUENCY_BINARY_D3SSD).

2. The ASCII databases NODOUT_SSD and ELOUT_SSD are saved in binout files. LS-PREPOST is able to read the binout files directly. Users can also convert these files to ASCII format simply feed them to the l2a program like this:

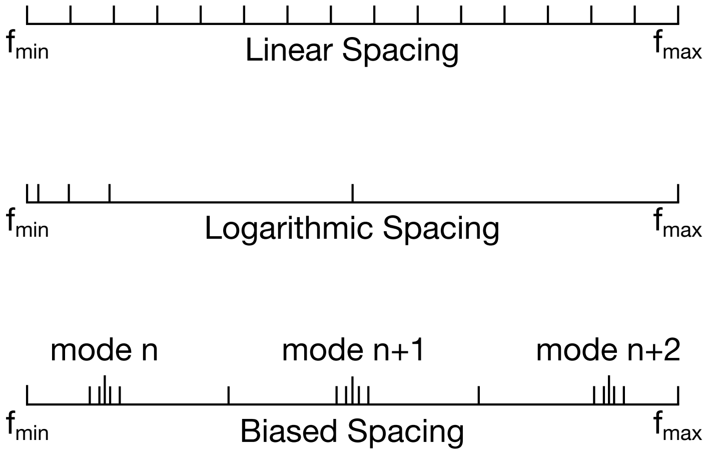


Figure 14-6. Spacing options of the frequency points.

l2a binout*

3. The nodes to be output to NODOUT_SSD are specified by card *DATABASE_HISTORY_NODE.

4. The solid, beam, shell and thick shell elements to be output to ELOUT_SSD are specified by the following cards:

   DATABASE_HISTORY_SOLID_*{OPTION}

   DATABASE_HISTORY_BEAM_*{OPTION}

   DATABASE_HISTORY_SHELL_*{OPTION}

   DATABASE_HISTORY_TSHELL_*{OPTION}

5. There are two methods to define the output frequencies.

   a) The first method is to define FMIN, FMAX, NFREQ and FSPACE. FMIN and FMAX specify the frequency range of interest and NFREQ specifies the number of frequencies at which results are required. FSPACE specifies the type of frequency spacing (linear, logarithmic or biased) to be used. These frequency points for which results are required can be spaced equally along the frequency axis (on a linear or logarithmic scale). Or they can be biased toward the eigenfrequencies (the frequency points are placed closer together at eigenfrequencies in the frequency range) so that the detailed definition of the response close to resonance frequencies can be obtained.

   b) The second method is to use a load curve (LCFREQ) to define the frequencies of interest.

DATABASE_FREQUENCY_BINARY_*OPTION

Options for frequency domain binary output files with the default names given include:

D3ACC Binary output file for BEM acoustics (element acoustic pressure contribution and contribution percentage). See also *FREQUENCY_DOMAIN_ACOUSTIC_BEM.

D3ACS Binary output file for FEM acoustics (acoustic pressure and sound pressure level). See also *FREQUENCY_DOMAIN_ACOUSTIC_FEM.

D3ATV Binary output file for acoustic transfer vectors given by BEM acoustic analysis. See also *FREQUENCY_DOMAIN_ACOUSTIC_BEM_ATV.

D3FTG Binary output file for random vibration fatigue analysis. See also *FREQUENCY_DOMAIN_RANDOM_VIBRATION_FATIGUE.

D3PSD Binary Power Spectral Density output file for random vibration analysis. See also *FREQUENCY_DOMAIN_RANDOM_VIBRATION.

D3RMS Binary Root Mean Square output file for random vibration analysis. See also *FREQUENCY_DOMAIN_RANDOM_VIBRATION.

D3SPCM Binary output file for response spectrum analysis. See also *FREQUENCY_DOMAIN_RESPONSE_SPECTRUM.

D3SSD Binary output file for steady state dynamics. See also *FREQUENCY_DOMAIN_SSD.

The D3ACC, D3ACS, D3ATV, D3FTG, D3PSD, D3RMS, D3SPCM and D3SSD files contain plotting information to plot data over the three dimensional geometry of the model. These databases can be plotted with LS-PrePost.

- The D3PSD file contains PSD state data for a range of frequencies. The D3SSD file contains state data for a range of frequencies.
- For D3SSD, the data can be real or complex, depending on the variable BINARY defined below.
- The D3ACC file contains acoustic pressure contribution (and contribution percentage) from each of the boundary elements for a range of frequencies, which are defined in the keyword *FREQUENCY_DOMAIN_ACOUSTIC_BEM.
- The D3ACS file contains acoustic results including acoustic pressure and sound pressure level for a range of frequencies, which are defined in the keyword *FREQUENCY_DOMAIN_ACOUSTIC_FEM.

- The D3FTG, D3RMS and D3SPCM files contain only one state each as they are the data for cumulative fatigue damage ratio, root mean square for random vibration and peak response for response spectrum analysis separately.
- The D3ATV file contains NFIELD × NFREQ states, where NFIELD is the number of acoustic field points and NFREQ is the number of output frequencies.

Card 1	1	2	3	4	5	6	7	8
Variable	BINARY							
Type	I							
Default	-							
Remarks	1							

Additional cards for D3ACC keyword options.

Card 2	1	2	3	4	5	6	7	8
Variable	NID1	NID2	NID3	NID4	NID5	NID6	NID7	NID8
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

Additional card for D3PSD and D3SSD keyword options.

Card 2	1	2	3	4	5	6	7	8
Variable	FMIN	FMAX	NFREQ	FSPACE	LCFREQ			
Type	F	F	I	I	I			
Default	0.0	0.0	0	0	0			

VARIABLE	DESCRIPTION
BINARY	Flag for writing the binary plot file. EQ.0: Off EQ.1: write the binary plot file EQ.2: write the complex variable binary plot file (D3SSD only) EQ.90: write only real part of frequency response (D3SSD only) EQ.91: write only imaginary part of frequency response (D3SSD only)
NID1,…	Field point node ID for writing D3ACC file (up to 10 NID are allowed)
FMIN	Minimum frequency for output (cycles/time)
FMAX	Maximum frequency for output (cycles/time).
NFREQ	Number of frequencies for output.
FSPACE	Frequency spacing option for output: EQ.0: linear EQ.1: logarithmic EQ.2: biased
LCFREQ	Load Curve ID defining the frequencies for output.

Remarks:

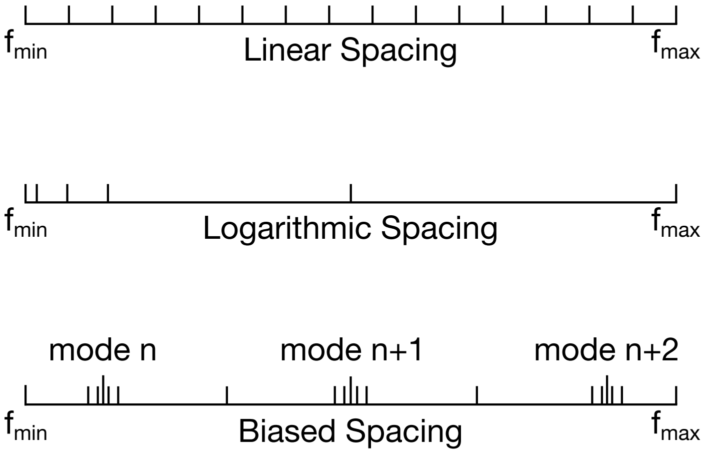


Figure 14-7. Spacing options of the frequency points.

1. For OPTION = D3SSD, If BINARY = 1, only the magnitude of the displacement, velocity, acceleration and stress response is written into the binary database "d3ssd" which can be accessed by LS-PrePost 3.0 or older versions. For customers using LS-PrePost 3.0 or older versions, it is suggested to set BINARY = 1. If BINARY = 2, both the magnitude and the phase angle of the response are written into "d3ssd" so that LS-PrePost (3.1 or higher versions) can run modal expansion (to show the cyclic time history fringe plot) on each output frequency. If BINARY = 90 or 91, only real or imaginary part of the response is written into "d3ssd".

2. There are two methods to define the output frequencies.

   a) The first method is to define FMIN, FMAX, NFREQ and FSPACE. FMIN and FMAX specify the frequency range of interest and NFREQ specifies the number of frequencies at which results are required. FSPACE specifies the type of frequency spacing (linear, logarithmic or biased) to be used. These frequency points for which results are required can be spaced equally along the frequency axis (on a linear or logarithmic scale). Or they can be biased toward the eigenfrequencies (the frequency points are placed closer together at eigenfrequencies in the frequency range) so that the detailed definition of the response close to resonance frequencies can be obtained.

   b) The second method is to use a load curve (LCFREQ) to define the frequencies of interest.

*DATABASE_FSI

Purpose: When a Lagrangian mesh overlaps with an Eulerian or ALE mesh, the fluid-structure (or ALE-Lagrangian) interaction is often modeled using the *CONSTRAINED_LAGRANGE_IN_SOLID card. This keyword (*DATABASE_FSI) causes certain coupling information related to the flux through and load on selected Lagrangian surfaces defined in corresponding *CONSTRAINED_LAGRANGE_IN_SOLID card to be written to the ASCII-based dbfsi file or in the case of MPP-DYNA the binout file.

> NOTE: This card *must* be associated with a *CONSTRAINED_LAGRANGE_IN_SOLID penalty method coupling. This card is *not* compatible with constrained-based coupling.

Card 1	1	2	3	4	5	6	7	8
Variable	DTOUT							
Type	F							

Surface Card. Add one card per surface. This input terminates at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	DBFSI_ID	SID	SIDTYPE	SWID	CONVID	NDSETID	CID	
Type	I	I	I	I	I	I	I	

VARIABLE	DESCRIPTION
DTOUT	Output interval time step
DBFSI_ID	Surface ID (for reference purposes only) or a DATABASE_FSI entity ID. It consists of a geometric entity defined by the SID below.

VARIABLE **DESCRIPTION**

SID

Set ID defining the geometrical surface(s) through which or upon which some data is to be tracked and output to a file called "dbfsi". This set ID can be a (1) PID or (2) PSID or (3) SGSID. This Lagrangian SID must be contained in a Lagrangian slave SID defined in a corresponding coupling card, *CONSTRAINED_LAGRANGE_IN_SOLID.

SIDTYPE

Set type:

EQ.0: Part set

EQ.1: Part

EQ.2: Segment set

SWID

This is an ID from a corresponding *ALE_FSI_SWITCH_MMG_ID card. This card allows for the AMMG ID of an ALE material to be switched as it passes across a monitoring surface. If defined, the accumulative mass of the "switched" ALE multi-material group (AMMG) is written out under the "mout" parameter in the "dbfsi" file.

CONVID

This is used mostly for airbag application only: CONVID is an ID from a corresponding *LOAD_ALE_CONVECTION_ID card which computes the heat transfer between inflator gas (ALE material) and the inflator canister (Lagrangian part). If defined, the temperature of the Lagrangian part having heat transfer with the gas, and its change in temperature as function of time are output in the "dbfsi" file.

NDSETID

Set ID consisting of the nodes on which the moments of the forces applied on SID are computed. See Remark 3.

CID

Coordinate system ID, see *DEFINE_COORDINATE_SYSTEM.

Remarks:

1. **Overview of dbfsi File.** The dbfsi parameters output are enumerated below.

pres = Averaged estimated coupling pressure over each surface entity being monitored. For example, if using SI base units for mass-length-time-temperature, this pressure would then be in Pascal.

fx, fy, fz = Averaged total estimated coupling force components (N in metric units) along the global coordinate directions, over each surface entity defined, and acting at the centroid of each surface.

mout = Accumulated mass (Kg in metric units) passing through each DBFS_ID surface entity. See Remark 2 below. (This parameter used to be called "pleak").

obsolete = (This parameter used to be called "mflux").

gx, gy, gz = Average estimated leakage-control force component over the surface entity. This data is useful for debugging. Leakage control forces are too large (relative to the main coupling forces, fx, fy and fz) may indicate that alternate coupling approach should be considered since the main coupling force is putting out too little resistance to leakage. (These parameters used to be called fx-lc, fy-lc and fz-lc).

Ptmp = Lagrangian part Temperature (Activated only when the *LOAD_ALE_CONVECTION card is used).

PDt = Lagrangian part Temperature change (Activated only when the *LOAD_ALE_CONVECTION card is used).

2. **MOUT.** "mout" parameter in the "dbfsi" output from this keyword contains the accumulated mass passing through each DBFS_ID surface entity. For 4 different cases:

   a) When LCIDPOR is defined in the coupling card (CLIS), porous accumulated mass transport across a Lagrangian shell surface may be monitored and output in "mout".

   b) Porous flow across Lagrangian shell may also be defined via a load curve in the *MAT_FABRIC card, and similar result will be tracked and output. This is an alternate form of (a).

   c) When NVENT in the CLIS card is defined (isentropic venting), the venting mass transport across the isentropic vent hole surface may be output in "mout".

   d) When an *ALE_FSI_SWITCH_MMG_ID card is defined, and the SWID parameter specifies this ID to be tracked, then the amount of accumulated mass that has been switched when passing across a monitoring surface is output.

3. **Calculation of Moments for NDSETID.** A geometrical surface SID has a centroid where the coupling forces are averaged. The distances between this centroid and the nodes defined by the set NDSETID are the lever arms. The moments are the cross-products of these distances with the averaged coupling forces. For each node in the set NDSETID, a new line in the “dbfsi” file is inserted after each output for the corresponding coupling forces (see Remark 1). These additional lines have the format following the template established by the example in Remark 1 where the forces are replaced by the moments and the node ID replaces the DBFSI_ID values.

Example:

Consider a model with a Lagrangian mesh overlaps with an Eulerian or ALE mesh. On the Lagrangian mesh, there are 3 Lagrangian surface sets over which some data is to be written out.

```
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
$ INPUT:
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
*DATABASE_FSI
$       dt
  2.97E-06
$ DBFSI_ID       SID     STYPE      swid    convid [STYPE: 0=PSID;1=PID;2=SGSID]
        11         1         2
        12         2         2
        13         3         1
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
$ This reads:
$ DBFSI_ID 11 is defined by a SID=1: a SGSID = as specified by STYPE=2
$ DBFSI_ID 12 is defined by a SID=2: a SGSID = as specified by STYPE=2
$ DBFSI_ID 13 is defined by a SID=3: a PID   = as specified by STYPE=1
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
$ An OUTPUT file called “dbfsi” looks like the following:
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
     Fluid-structure interaction output
     Number of surfaces:       3

        id          pres            fx            fy            fz          mout
                obsolete            gx            gy            gz          Ptmp
PDt
         time=  0.00000E+00
        11    0.0000E+00    0.0000E+00    0.0000E+00    0.0000E+00    0.0000E+00
              0.0000E+00    0.0000E+00    0.0000E+00    0.0000E+00    0.0000E+00
0.0000E+00

        12    0.0000E+00    0.0000E+00    0.0000E+00    0.0000E+00    0.0000E+00
              0.0000E+00    0.0000E+00    0.0000E+00    0.0000E+00    0.0000E+00
0.0000E+00

        13    0.0000E+00    0.0000E+00    0.0000E+00    0.0000E+00    0.0000E+00
              0.0000E+00    0.0000E+00    0.0000E+00    0.0000E+00    0.0000E+00
0.0000E+00

         time=  0.29709E-05
        11    0.0000E+00    0.0000E+00    0.0000E+00    0.0000E+00    0.0000E+00
              0.0000E+00    0.0000E+00    0.0000E+00    0.0000E+00    0.0000E+00
0.0000E+00
```

```
       12    0.0000E+00    0.0000E+00    0.0000E+00    0.0000E+00    0.0000E+00
             0.0000E+00    0.0000E+00    0.0000E+00    0.0000E+00    0.0000E+00
0.0000E+00

       13    0.0000E+00    0.0000E+00    0.0000E+00    0.0000E+00    0.0000E+00
             0.1832E-06    0.0000E+00    0.0000E+00    0.0000E+00    0.0000E+00
0.0000E+00
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
```

*DATABASE_FSI_SENSOR

Purpose: This card activates the output of an ASCII file called "dbsensor". Its input defines the pressure sensors' locations which follow the positions of some Lagrangian segments during the simulation. Its ASCII output file, dbsensor, contains the spatial position of the sensor and its recorded pressure from the ALE elements containing the sensors. This card is activated when a *CONSTRAINED_LAGRANGE_IN_SOLID card is used and the Lagrangian shell elements defining the locations of the sensors must be included in the slave or structure coupling set.

Card 1	1	2	3	4	5	6	7	8
Variable	DT							
Type	F							

Surface Card. Add one card per surface. This input terminates at the next keyword ("*") card.

Card	1	2	3	4	5	6	7	8
Variable	DBFSI_ID	NID	SEGMID	OFFSET	ND1	ND2	ND3	
Type	I	I	I	F	I	I	I	

VARIABLE	DESCRIPTION
DT	Output interval
DBFSI_ID	Pressure-Sensor ID.
NID	An optional Lagrangian node ID defining an approximate pressure sensor location with respect to a Lagrangian shell element. This is not a required input.
SEGMID	A required Lagrangian element ID for locating the pressure sensor. If NID = 0 or blank, the sensor will be automatically placed in the center of this SEGMID, accounting for the offset distance. If the model is 3D, the Lagrangian element can be a shell or solid (for this latter, ND1 and ND2 are required to define the face). If the model is 2D, the Lagrangian element can be a beam or shell (for this latter, ND1 and ND2 are required to define the side).

VARIABLE	DESCRIPTION
OFFSET	Offset distance between the pressure sensor and the Lagrangian segment surface. If it is positive, it is on the side pointed to by the segment normal vector and vice versa.
ND1, ND2, ND3	Nodes defining the solid face in 3D or shell side in 2D, from which the sensor is located. In 3D, if the solid face has 4 nodes, only the diagonal opposites ND1 and ND2 are required. If the solid face is triangular, a third node ND3 should be provided. In 2D, only ND1 and ND2 are required to define the shell side.

Remarks:

1. The output parameters in the "dbsensor" ASCII file are:

   ID = Sensor ID.

   x, y, z = Sensor spatial location.

   P = Sensor recorded pressure (Pa) from the ALE fluid element containing the sensor.

   For example, to plot the sensor pressure in LS-Prepost, select:

   ASCII → dbsensor → LOAD → (select sensor ID) → Pressure → PLOT

Example 1:

```
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
$ INPUT:
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
*DATABASE_FSI_SENSOR
      0.01
$ DBFSI_ID       NID SEGMENTID    OFFSET
        10       360       355      -0.5
        20       396       388      -0.5
        30       324       332      -0.5
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
$ The 1st line reads:
$ SENSOR_ID 10 is located by segment-ID=355.  Node-ID=360 precisely locate this
$ sensor (if NID=0, then the sensor is located at the segment center).  This
$ sensor is located 0.5 length unit away from the segment surface.  Negative
$ sign indicates a direction opposite to the segment normal vector.
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
$ An OUTPUT file called "dbsensor" looks like the following:
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
     ALE sensors output
     Number of sensors:  3

        id         x             y             z             p
         time=  0.17861E-02
        10    0.0000E+00    0.0000E+00   -0.3900E+00    0.1085E-03
        20   -0.2250E+02    0.2250E+02   -0.3900E+00    0.1085E-03
        30    0.2250E+02   -0.2250E+02   -0.3900E+00    0.1085E-03
         time=  0.20081E-02
        10    0.0000E+00    0.0000E+00   -0.3900E+00    0.1066E-03
        20   -0.2250E+02    0.2250E+02   -0.3900E+00    0.1066E-03
        30    0.2250E+02   -0.2250E+02   -0.3900E+00    0.1066E-03
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
$ ID    = DBFSI_ID
$ x,y,z = Sensor location (defined based on a Lagrangian segment)
$ p     = Sensor pressure as taken from the fluid element containing the sensor.
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
```

DATABASE_HISTORY_*OPTION

Available options include:

BEAM

BEAM_SET

BEAM_ID

DISCRETE

DISCRETE_ID

DISCRETE_SET

NODE

NODE_ID

NODE_LOCAL

NODE_LOCAL_ID

NODE_SET

NODE_SET_LOCAL

SEATBELT

SEATBELT_ID

SHELL

SHELL_ID

SHELL_SET

SOLID

SOLID_ID

SOLID_SET

SPH

SPH_SET

TSHELL

TSHELL_ID

TSHELL_SET

Purpose: Control which nodes or elements are output into the binary history file, d3thdt, the ASCII file nodout, the ASCII file elout and the ASCII file sphout. Define as many cards as necessary. The next "*" card terminates the input. See also *DATABASE_BINARY_*OPTION* and *DATABASE_*OPTION*.

Node/Element Cards for Case I (no "ID", and no "LOCAL"). Cards for keyword options BEAM, BEAM_SET, DISCRETE, DISCRETE_SET, NODE, NODE_SET, SEATBELT, SHELL, SHELL_SET, SOLID, SOLID_SET, SPH, SPH_SET, TSHELL, and TSHELL_SET. Include as many as needed. Input terminates at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	ID1	ID2	ID3	ID4	ID5	ID6	ID7	ID8
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
ID*n*	NODE/NODE_SET or element/element set ID *n*. Elements may be BEAM/BEAM_SET, DISCRETE/DISCRETE_SET, SEATBELT, SHELL/SHELL_SET, SOLID/SOLID_SET, or TSHELL/TSHELL_SET. The contents of the files are given in Table 14-2 for nodes, Table 14-3 for solid elements, Table 14-4 for shells and thick shells, and Table 14-5 for beam elements. In the binary file, D3THDT, the contents may be extended or reduced with the *DATABASE_EXTENT_BINARY definition.

Node/Element Cards for Case II ("ID" option, but no "LOCAL"). Cards for keyword options BEAM_ID, NODE_ID, SEATBELT_ID, SHELL_ID, SOLID_ID, and TSHELL_ID. Include as many as needed. Input terminates at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	HEADING						
Type	I	A70						

VARIABLE	DESCRIPTION
ID	Node or element ID

VARIABLE	DESCRIPTION
HEADING	A description of the node or element. It is suggested that unique descriptions be used. This description is written into the D3HSP file and into the ASCII databases nodout and elout.

Node Cards for Case III ("LOCAL" option). Card 1 for keyword options NODE_LOCAL, NODE_LOCAL_ID, and NODE_SET_LOCAL. Include as many cards as needed to specify all the nodes. This input terminates at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	CID	REF	HFO				
Type	I	I	I	I				

ID Card for Case III. Additional card for ID option. This card is only used for the NODE_LOCAL_ID keyword option. When activated, each node is specified by a pair of cards consisting of "Card 1," and, secondly, this card. Include as many pairs as needed to specify all the nodes. This input terminates at the next keyword ("*") card.

<table>
<tr><td>Card 2</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr>
<tr><td>Variable</td><td colspan="7">HEADING</td><td></td></tr>
<tr><td>Type</td><td colspan="7">A70</td><td></td></tr>
</table>

VARIABLE	DESCRIPTION
ID	NODE/NODE_SET set ID. The contents of the files are given in Table 14-2 for nodes. See the remark below concerning accelerometer nodes.
CID	Coordinate system ID for nodal output. See DEFINE_COORDINATE options.

VARIABLE **DESCRIPTION**

REF

Output coordinate system for displacements, velocities, and accelerations. (Nodal coordinates are always in the global coordinate system.)

EQ.0: Output is in the local system fixed for all time from the beginning of the calculation. If CID is nonzero, FLAG in the corresponding *DEFINE_COORDINATE_NODES command must be set to 0. FLAG has no bearing on results when REF is set to 1 or 2.

EQ.1: Translational output is the projection of the node's absolute translational motion onto the local system. The local system is defined by the *DEFINE_COORDINATE_NODES command and can change orientation according to the movement of the three defining nodes. The defining nodes can belong to either deformable or rigid parts.

EQ.2: Translational output is the projection of the node's relative translational motion onto the local system. Here, "relative" means relative to node N1 of that local system. In other words, the displacement of the origin (node N1) of the local coordinate system is first subtracted from the displacement of the node of interest before projecting it onto the translating and rotating local coordinate system. The local system is defined as described in REF = 1 above. If dynamic relaxation is used, the reference location is reset when convergence is achieved. Rotational output is truly relative to the updated location coordinate system only if REF = 2.

HFO

Flag for high frequency output into nodouthf

EQ.0: Nodal data written to nodout file only

EQ.1: Nodal data also written nodouthf at the higher frequency

HEADING

A description of the nodal point. It is suggested that unique description be used. This description is written into the d3hsp file and into the ASCII database nodout.

Remarks:

1. If a node belongs to an accelerometer, see *ELEMENT_SEATBELT_ACCELEROMETER, and if it also appears as an active node in the NODE_LOCAL

or NODE_SET_LOCAL keyword, the coordinate system, CID, transformations will be skipped and the LOCAL option will have no effect.

*DATABASE_MASSOUT

Purpose: Output nodal masses into ASCII file MASSOUT.

Card 1	1	2	3	4	5	6	7	8
Variable	SETID	NDFLG	RBFLG					
Type	I	I	I					
Default	0	1	0					

VARIABLE	DESCRIPTION
SETID	Optional set ID. EQ.0: mass output for all nodes, LT.0: no output, GT.0: set ID identifying nodes whose mass will be output.
NDFLG	Database extent: EQ.1: output translational mass for deformable nodes identified by SETID (default), EQ.2: output translational mass and rotary inertias for the deformable nodes identified by the SETID. EQ.3: output translational mass for deformable and rigid nodes identified by SETID (default), EQ.4: output translational mass and rotary inertias for the deformable and rigid nodes identified by the SETID.
RBFLG	Rigid body data: EQ.0: no output for rigid bodies, EQ.1: output rigid body mass and inertia.

Remarks:

1. Nodes and rigid bodies with no mass are not output. By inference, when the set ID is zero and no output shows up for a node, then the mass of that node is zero.

*DATABASE_NODAL_FORCE_GROUP

Purpose: Define a nodal force group for output into the ASCII file nodfor. The output interval must be specified using *DATABASE_NODFOR (see *DATABASE_*OPTION*).

Card 1	1	2	3	4	5	6	7	8
Variable	NSID	CID						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
NSID	Nodal set ID, see *SET_NODE_*OPTION*.
CID	Coordinate system ID for output of data in local system,

Remarks:

1. The reaction forces in the global x, y, and z directions (and local x, y, and z directions if CID is defined above) for the nodal force group are written to the nodfor file (see *DATABASE_NODFOR) along with the external work done by these reaction forces. The reaction forces in the global x, y, and z directions for each node in the nodal force group are also written to nodfor. These forces can be a result of applied boundary forces such as nodal point forces and pressure boundary conditions, body forces, and contact interface forces. In the absence of body forces, interior nodes would always yield a null force resultant vector. In general this option would be used for surface nodes.

*DATABASE_PAP_OUTPUT

Purpose: Set contents of output files for pore air pressure calculations.

Card 1	1	2	3	4	5	6	7	8
Variable	IVEL	IACCX	IACCY	IACCZ	NCYOUT			
Type	I	I	I	I	I			
Default	0	0	0	0	100			

VARIABLE	DESCRIPTION

IVEL

Meaning of "Velocity" in d3plot and d3thdt output files

EQ.0: Nodal velocity vector

EQ.1: Seepage velocity vector

IACCX, Y, Z

Meaning of "X/Y/Z-Acceleration" in d3plot and d3thdt output files

EQ.0: Not written

EQ.21: Nodal air density

EQ.22: Nodal pore air pressure

EQ.24: Nodal air mass

EQ.25: Nodal air mass flow rate

NCYOUT

Number of cycles between outputs of calculation status to d3hsp and log files

*DATABASE_PROFILE

Purpose: Plot the distribution or profile of a data along x, y, or z-direction.

Card 1	1	2	3	4	5	6	7	8
Variable	DT	ID	TYPE	DATA	DIR	UPDLOC	MMG	
Type	I	I	I	I	I	I	I	
Default	none	none	none	none	none	0	0	

VARIABLE **DESCRIPTION**

DT Interval time.

ID Set ID.

TYPE Set type:

EQ.1: Node Set,

EQ.2: Solid Set,

EQ.3: Shell Set,

EQ.4: Segment Set,

EQ.5: Beam Set.

DATA Data type:

EQ.1: x-velocity,

EQ.2: y-velocity,

EQ.3: z-velocity,

EQ.4: velocity magnitude,

EQ.5: x-acceleration,

EQ.6: y-acceleration,

EQ.7: z-acceleration,

EQ.8: acceleration magnitude,

EQ.9: pressure,

EQ.10: xx-stress,

VARIABLE	DESCRIPTION
	EQ.11: *yy*-stress, EQ.12: *zz*-stress, EQ.13: *xy*-stress, EQ.14: *yz*-stress, EQ.15: *zx*-stress, EQ.16: temperature, EQ.17: volume fraction, EQ.18: kinetic energy, EQ.19: internal energy, EQ.20: density.
DIR	Direction: EQ.1: *x*-direction, EQ.2: *y*-direction, EQ.3: *z*-direction, EQ.4: Curvilinear (relative distances between elements of set ID are added up in the order defined by the set)
UPDLOC	Flag to update the set location: EQ.0: Only the initial position of set ID is considered EQ.1: The positions of the elements composing the set are updated each DT
MMG	Multi-Material ALE group id. See Remark 2. GT.0: Multi-Material ALE group id LT.0: |MMG| is the id of a *SET_MULTI-MATERIAL_GROUP_LIST that can list several Multi-Material ALE group ids.

Remarks:

1. At a given time *T* the profile is written in a file named profile_DATA_DIR_time*T*.xy (DATA and DIR are replaced by the data and direction names respectively). The file has a xyplot format that LS-PrePost can read and plot. For example, DATA = 9, DIR = 2 and DT = 0.1 sec will save a pressure profile at

$t = 0.0$ sec in profile_pressure_y_time0.0.xy, at $t = 0.1$ sec in profile_pressure_y_time0.1.xy, at $t = 0.2$ sec in profile_pressure_y_time0.2.xy.

2. In the case of a multi-material ALE model (elform = 11 in *SECTION_SOLID or *SECTION_ALE2D or *SECTION_ALE1D), an element can contain several materials with each material being associated with its own pressures and stresses. It is the default behavior for volume averaging to be applied to element data before being written out; however, when the multi-material group field, MMG, is set, then element data are output only for the specified materials.

*DATABASE_PWP_FLOW

Purpose: Request output containing net inflow of fluid at a set of nodes.

Card 1	1	2	3	4	5	6	7	8
Variable	NSET							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
NSET	Node set ID

Remarks:

Any number of these cards can be used. Nett inflow or outflow arises when maintaining an applied PWP boundary condition implies addition or removal of water.

Output is written to a file named database_pwp_flow.csv, a comma-separated ascii file. Each line consists of (time, flow1, flow2, …) where flow1 is the total inflow at the node set for the first DATABASE_PWP_FLOW request, flow2 is for the second, etc.

*DATABASE_PWP_OUTPUT

Purpose: Set contents of output files for pore pressure calculations.

Card 1	1	2	3	4	5	6	7	8
Variable	IVEL	IACCX	IACCY	IACCZ	NCYOUT			
Type	I	I	I	I	I			
Default	0	0	0	0	100			

VARIABLE	DESCRIPTION
IVEL	Meaning of "Velocity" in d3plot and d3thdt output files EQ.0: Nodal velocity vector EQ.1: Seepage velocity vector
IACCX, Y, Z	Meaning of "X/Y/Z-Acceleration" in d3plot and d3thdt output files EQ.0: Not written EQ.1: Total pwp head EQ.2: Excess pwp head (this is also written as temperature) EQ.3: Target rate of volume change EQ.4: Actual rate of volume change EQ.7: Hydraulic pwp head EQ.8: Error in rate of volume change (calculated from seepage minus actual) EQ.9: Volume at node EQ.10: Rate of volume change calculated from seepage EQ.14: Void volume (generated at suction limit) EQ.17: NFIXCON (e.g: +4/-4 for nodes on suction limit)
NCYOUT	Number of cycles between outputs of calculation status to d3hsp, log, and tdc_control_output.csv files (time-dependent and steady-state analysis types).

*DATABASE_RCFORC_MOMENT

Purpose: Define contact ID and nodes for moment calculations. Moments are written to rcforc according to output interval given in *DATABASE_RCFORC. If *DATABASE_RCFORC_MOMENT is not used, the moments reported to rcforc are about the origin (0, 0, 0).

Card 1	1	2	3	4	5	6	7	8
Variable	CID	NODES	NODEM					
Type	I	I	I					

VARIABLE	DESCRIPTION
CID	Contact ID
NODES	Node about which moments are calculated due to contact forces on slave surface.
NODEM	Node about which moments are calculated due to contact forces on master surface.

*DATABASE_RECOVER_NODE

Purpose: Recovers the stresses at nodal points of solid or thin shell elements by using Zienkiewicz-Zhu's Superconvergent Patch Recovery method.

Card 1	1	2	3	4	5	6	7	8
Variable	PSID	IAX	IAY	IAZ				
Type	I	A	A	A				
Default	0	0	0	0				

VARIABLE **DESCRIPTION**

PSID

Part set ID of solid or thin shell elements whose nodal stress will be recovered

IAX, IAY, IAZ

Meaning of "$x/y/z$-Acceleration" in d3plot and d3thdt output files

EQ.SMNPD: the minimum principal deviator stress

EQ.SMNPR: the minimum principal stress

EQ.SMXPD: the maximum principal deviator stress

EQ.SMXPR: the maximum principal stress

EQ.SMXSH: the maximum shear stress

EQ.SPR: nodal pressure

EQ.SVM: nodal von Mises stress

EQ.SXX: nodal normal stress along x direction

EQ.SYY: nodal normal stress along y direction

EQ.SZZ: nodal normal stress along z direction

EQ.SXY: nodal shear stress along x-y direction

EQ.SYZ: nodal shear stress along y-z direction

EQ.SZX: nodal shear stress along z-x direction

For shell elements append either "B" or "T" to the input string to recover nodal stresses at the bottom or top layer of shell elements. For example, SPRT recovers the nodal pressure at the top layer.

Remarks:

1. Recovered stresses are in global coordinate system.

*DATABASE_SPRING_FORWARD

Purpose: Create spring forward nodal force file. This option is to output resultant nodal force components of sheet metal at the end of the forming simulation into an ASCII file, "SPRING-FORWARD", for spring forward and die corrective simulations.

Card 1	1	2	3	4	5	6	7	8
Variable	IFLAG							
Type	I							

VARIABLE	DESCRIPTION
IFLAG	Output type: EQ.0: off, EQ.1: output element nodal force vector for deformable nodes.

*DATABASE_SUPERPLASTIC_FORMING

Purpose: Specify the output intervals to the superplastic forming output files. The option *LOAD_SUPERPLASTIC_FORMING must be active.

Card 1	1	2	3	4	5	6	7	8
Variable	DTOUT							
Type	F							

VARIABLE	DESCRIPTION
DTOUT	Output time interval for output to “pressure”, “curve1” and “curve2” files. The “pressure” file contains general information from the analysis and the files “curve1” and “curve2” contain pressure versus time from phases 1 and 2 of the analysis. The data in the pressure and curve files may be plotted using ASCII → superpl in LS-PrePost.

DATABASE_TRACER_*{OPTION}

Purpose: Tracer particles will save a history of either a material point or a spatial point into an ASCII file: trhist. This history includes positions, velocities, and stress components. The option *DATABASE_TRHIST must be active. This option applies to ALE, SPH and DEM (Discrete Element Method) problems.

Available options are:

<BLANK>

DE

The DE option defines a tracer corresponding to discrete elements (*ELEMENT_DISCRETE_SPHERE) . See Remarks 2 and 4.

Card	1	2	3	4	5	6	7	8
Variable	TIME	TRACK	X	Y	Z	AMMGID	NID	RADIUS
Type	F	I	F	F	F	I	I	F
Default	0.0	0	0	0	0	0	0	0.0

VARIABLE	DESCRIPTION
TIME	Start time for tracer particle
TRACK	Tracking option: EQ.0: particle follows material, EQ.1: particle is fixed in space.
X	Initial *x*-coordinate
Y	Initial *y*-coordinate
Z	Initial *z*-coordinate
AMMGID	The AMMG ID (ALE multi-material group) of the material being tracked in a multi-material ALE element. See Remark 1.

VARIABLE **DESCRIPTION**

NID

An optional node ID defining the initial position of a tracer particle. If defined, its coordinates will overwrite the x, y, z coordinates above. This feature is for TRACK = 0 only and can be applied to ALE tracers and DE tracers. See Remark 2.

RADIUS

Radius is used only for the DE option to indicate whether the tracer follows and monitors a single discrete element or multiple discrete elements.

GT.0: The tracer takes the average results of all discrete elements located inside a sphere with radius = RADIUS. That sphere stays centered on the DE tracer.

LT.0: The discrete element closest to the tracer is used. The magnitude of RADIUS in this case is unimportant.

Remarks:

1. **Multi-Material Groups.** ALE elements can contain multi-materials. Each material is referred to as an ALE multi-material group or AMMG. Each AMMG has its list of history variables that can be output. For example, if a tracer is in a mixed element consisting of 2 AMMGs, and the history variables of AMMG 1 are to be output or tracked, the AMMGID should be defined as AMMGID=1. If AMMGID=0, a volume-fraction-weighted-averaged pressure will be reported instead.

2. **NID Description.** For ALE, NID is a massless dummy node. Its location will be updated according to the motion of the ALE material.

   For the DE option, NID is a discrete element node that defines the initial location of the tracer. The DE tracer continues to follow that node if RADIUS < 0. On the other hand, the DE tracer's location is updated according to the average motion of the group of DE nodes inside the sphere defined by RADIUS when RADIUS > 0.

3. **Tracer particles in ambient ALE elements.** Since the auxiliary variables (6 stresses, plastic strain, internal energy, …) for ambient elements are reset to their initial values before and after advection and tracer data are stored in trhist during the advection cycle, tracers in ambient elements show the initial stresses, not the current ones.

4. **Discrete Elements.** If _DE is used, tracer particles will save a history of either a material point or a spatial point into an ASCII file: demtrh. This history in-

cludes positions, velocities components, stress components, porosity, void ratio, and coordination number. The option *DATABASE_TRHIST must be active.

*DATABASE_TRACER_GENERATE

Purpose: Generate tracer particles along an isosurface for a variable defined in the VALTYPE list. The tracer particles follow the motion of this surface and save data histories into a binary file called trcrgen_binout (See Remarks 4 and 5). These histories are identical to the ones output by *DATABASE_TRACER into the trhist file. They include positions, velocities, and stress components. Except for the positions and element id specifying where the tracer is, the output can be controlled with the VARLOC and VALTYPE2 fields. This option applies to ALE problems.

Card 1	1	2	3	4	5	6	7	8
Variable	DT	VALOW	VALUP	VALTYPE1	SET	SETYPE	MMGSET	UPDT
Type	F	F	F	I	I	I	I	F
Default	none	0.0	0.0	none	0	0	none	0.0

Optional Variable Cards. Cards defining new variables to be output to t trcrgen_binout instead of the default ones. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	VARLOC	VALTYPE2	MMGSET					
Type	I	I	I					
Default	0	0	0					

VARIABLE	DESCRIPTION
DT	Interval time between each tracer generation and position update (See Remark 1).
VALOW, VALUP	Range of values between which the isosurface is defined. VALOW is the lower bound while VALUP is the upper bound. See Remark 2. The value at the isosurface is 0.5(VALOW + VALUP). The variable with this value is defined by VALTYPE.

VARIABLE **DESCRIPTION**

VALTYPE1 The variable that will be used to generate the isosurfaces. See VALTYPE2 for enumeration of values.

VALTYPE2 Data to be output to the trcrgen_binout file. The interpretation of VALTYPE1 and VALTYPE2 is enumerated in the following list:

EQ.1:	xx-stress
EQ.2:	yy-stress
EQ.3:	zz-stress
EQ.4:	xy-stress
EQ.5:	yz-stress
EQ.6:	zx-stress
EQ.7:	plastic strain
EQ.8:	internal energy
EQ.9:	bulk viscosity
EQ.10:	relative volume
GE.11 and LE.19:	other auxiliary variables
EQ.20:	pressure
EQ.21:	density
EQ.22:	material volume
EQ.23:	compression ratio
EQ.24:	element volume fraction
EQ.25:	nodal volume fraction
EQ.26:	x-position
EQ.27:	y-position
EQ.28:	z-position
EQ.29:	x-velocity
EQ.30:	y-velocity
EQ.31:	z-velocity
EQ.31:	velocity
EQ.33:	x-acceleration
EQ.34:	y- acceleration

VARIABLE	DESCRIPTION
	EQ.35: *z*- acceleration EQ.36: acceleration EQ.37: nodal mass EQ.38: nodal temperature
SET	Set ID (See Remark 2)
SETYPE	Type of set (See Remark 2): EQ.0: solid set EQ.1: segment set EQ.2: node set
MMGSET	Multi-material group set (See Remark 3).
UPDT	Time interval between tracer position update (See Remark 1).
VARLOC	Variable location in trcrgen_binout to be replaced with the variable specified in the VALTYPE2 field: EQ.4: *x*-velocity EQ.5: *y*-velocity EQ.6: *z*-velocity EQ.7: *xx*-stress EQ.8: *yy*-stress EQ.9: *zz*-stress EQ.10: *xy*-stress EQ.11: *yz*-stress EQ.12: *zx*-stress EQ.13: plastic strain EQ.14: density EQ.15: relative volume

Remarks:

1. **DT.** The frequency to create tracers is defined by DT. The default value of UPDT, which is the time interval between updates to the tracer position, is also

set to DT. The default behavior, then, is to update tracer positions when a new tracer is created, however, by setting UPDT to a value less than DT tracer positions can be updated more frequently without creating new tracers.

2. **Tracing Algorithm.** When LS-DYNA adds new tracer particles (see DT) tracers are created at element centers, segment centers, or nodes depending on the set type (SETYPE). A new tracer particle is created when the value at the element center, segment center, or node center is in the bounding interval [VALOW, VALUP], provided that there is *not* already a nearby tracer particle. The tracer particles follow the iso-surface defined by the midpoint of the bounding interval (VALOW + VALUP)/2.

3. **Multi-Material Groups.** ALE elements can contain several materials. Each material is referred to as an ALE multi-material group. The volume fractions define how much of the element volume is occupied by the groups. Each group has their own variables for 0<VALTYPE<26. IF VALTYPE<21 or VALTYPE=23, the variable is volume averaged over the groups defined by MMGSET.

4. **Post-Procesing.** The output of *DATABASE_TRACER_GENERATE is written to a file named trcrgen_binout. To access the output in LS-PrePost: [TAB 2] → [LOAD] → [trcrgen_binout] → [trhist] → "Trhist Data" window contains a list of variables output for each tracer.

5. **Binary to ASCII File Conversion.** The variables in trhist and trcrgen_binout are arranged in an identical order. Therefore, the trhist can be obtained from the trcgen_binout file by using the l2a program located at http://ftp.lstc.com/-user/lsda.

.

*DEFINE

The keyword *DEFINE provides a way of defining boxes, coordinate systems, load curves, tables, and orientation vectors for various uses. The keyword cards in this section are defined in alphabetical order:

*DEFINE_ADAPTIVE_SOLID_TO_DES

*DEFINE_ADAPTIVE_SOLID_TO_SPH

*DEFINE_BOX

*DEFINE_BOX_ADAPTIVE

*DEFINE_BOX_COARSEN

*DEFINE_BOX_DRAWBEAD

*DEFINE_BOX_SPH

*DEFINE_CONNECTION_PROPERTIES_{OPTION}

*DEFINE_CONSTRUCTION_STAGES

*DEFINE_CONTACT_EXCLUSION

*DEFINE_CONTACT_VOLUME

*DEFINE_COORDINATE_NODES

*DEFINE_COORDINATE_SYSTEM

*DEFINE_COORDINATE_VECTOR

*DEFINE_CPM_BAG_INTERACTION

*DEFINE_CPM_CHAMBER

*DEFINE_CPM_GAS_PROPERTIES

*DEFINE_CPM_VENT

*DEFINE_CRASHFRONT

*DEFINE_CURVE_{OPTION}

*DEFINE_CURVE_BOX_ADAPTIVITY

*DEFINE_CURVE_COMPENSATION_CONSTRAINED

*DEFINE_CURVE_DRAWBEAD

*DEFINE_CURVE_DUPLICATE

*DEFINE_CURVE_ENTITY

*DEFINE_CURVE_FEEDBACK

*DEFINE_CURVE_FLC

*DEFINE_CURVE_FUNCTION

*DEFINE_CURVE_SMOOTH

*DEFINE_CURVE_TRIM_{OPTION}

*DEFINE_DEATH_TIMES_{OPTION}

*DEFINE_DE_ACTIVE_REGION

*DEFINE_DE_BOND

*DEFINE_DE_BY_PART

*DEFINE_DE_HBOND

*DEFINE_DE_INJECTION

*DEFINE_DE_MASSFLOW_PLANE

*DEFINE_DE_TO_BEAM_COUPLING

*DEFINE_DE_TO_SURFACE_COUPLING

*DEFINE_DE_TO_SURFACE_TIED

*DEFINE_ELEMENT_DEATH_{OPTION}

*DEFINE_ELEMENT_GENERALIZED_SHELL

*DEFINE_ELEMENT_GENERALIZED_SOLID

*DEFINE_FABRIC_ASSEMBLIES

*DEFINE_FILTER

*DEFINE_FORMING_BLANKMESH

*DEFINE_FORMING_CLAMP

*DEFINE_FORMING_CONTACT

*DEFINE_FRICTION

*DEFINE_FRICTION_ORIENTATION

*DEFINE_FUNCTION

*DEFINE_FUNCTION_TABULATED

*DEFINE_GROUND_MOTION

*DEFINE_HAZ_PROPERTIES

*DEFINE_HAZ_TAILOR_WELDED_BLANK

DEFINE_HEX_SPOTWELD_ASSEMBLY_*{OPTION}

*DEFINE_LANCE_SEED_POINT_COORDINATES

*DEFINE_MATERIAL_HISTORIES

*DEFINE_MULTI_DRAWBEADS_IGES

*DEFINE_PBLAST_AIRGEO

*DEFINE_PBLAST_GEOMETRY

*DEFINE_PLANE

*DEFINE_POROUS_{OPTION}

*DEFINE_PRESSURE_TUBE

*DEFINE_REGION

*DEFINE_SD_ORIENTATION

*DEFINE_SET_ADAPTIVE

*DEFINE_SPH_ACTIVE_REGION

*DEFINE_SPH_DE_COUPLING

*DEFINE_SPH_INJECTION

*DEFINE_SPH_TO_SPH_COUPLING

DEFINE_SPOTWELD_FAILURE_*{OPTION}

*DEFINE_SPOTWELD_FAILURE_RESULTANTS

*DEFINE_SPOTWELD_MULTISCALE

*DEFINE_SPOTWELD_RUPTURE_PARAMETER

*DEFINE_SPOTWELD_RUPTURE_STRESS

*DEFINE_STAGED_CONSTRUCTION_PART

*DEFINE_TABLE

*DEFINE_TABLE_2D

*DEFINE_TABLE_3D

*DEFINE_TABLE_MATRIX

*DEFINE_TARGET_BOUNDARY

*DEFINE_TRACER_PARTICLES_2D

*DEFINE_TRANSFORMATION

*DEFINE_TRIM_SEED_POINT_COORDINATES

*DEFINE_VECTOR

*DEFINE_VECTOR_NODES

Unless noted otherwise, an additional option "**TITLE**" may be appended to the ***DEFINE** keywords. If this option is used then an addition line is read for each section in 80a format which can be used to describe the defined curve, table, etc. At present LS-DYNA does make use of the title. Inclusion of titles gives greater clarity to input decks.

Examples for the *DEFINE keyword can be found at the end of this section.

DEFINE_ADAPTIVE_SOLID_TO_DES_*{OPTION}

Purpose: Adaptively transform a Lagrangian solid part or part set to DES (Discrete Element Sphere) particles (elements) when the Lagrangian solid elements comprising those parts fail. One or more DES particles will be generated for each failed element as debris. The DES particles replacing the failed element inherit the properties of the failed solid element including mass, and kinematical state.

The available options include:

<BLANK>

ID

ID Card. Additional card for the ID keyword option.

Optional	1	2	3	4	5	6	7	8
Variable	DID	HEADING						
Type	I	A70						
Default	none	None						

Card 1	1	2	3	4	5	6	7	8
Variable	IPID	ITYPE	NQ	IPDES	ISDES	RSF	OUTDES	
Type	I	I	I	I	I	F	I	
Default	none	none	None	none	None	1.0	0	

VARIABLE	DESCRIPTION
DID	Definition ID. This must be a unique number.
HEADING	Definition descriptor. It is suggested that unique descriptions be used.
IPID	ID of the solid part or part set to transform.

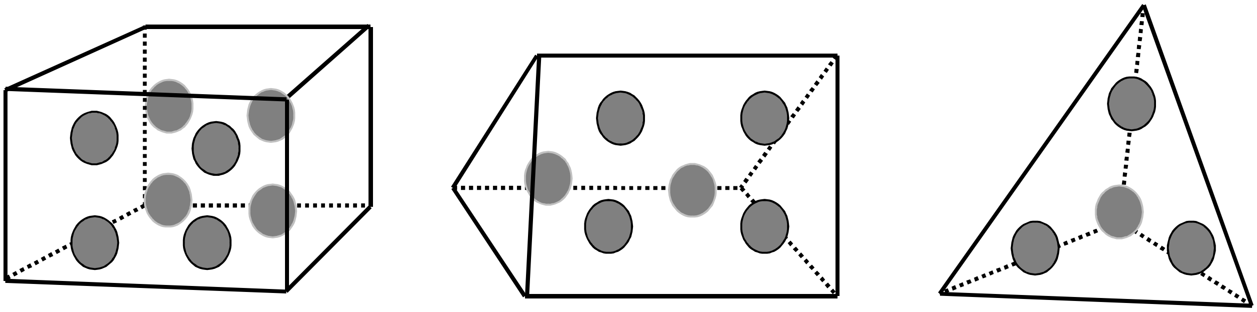


Figure 15-1. Left to right, illustration of conversion from solid to DES for NQ = 2 of hexahedron, pentahedron, and tetrahedron elements.

VARIABLE	DESCRIPTION
ITYPE	IPID type: EQ.0: Part ID, NE.1: Part set ID.
NQ	Adaptive option for hexahedral elements. For tetrahedral and pentahedral elements, see Remark 1: EQ.1: Adapt one solid element to one discrete element, EQ.2: Adapt one solid element to 8 discrete elements, EQ.3: Adapt one solid element to 27 discrete elements.
IPDES	Part ID for newly generated discrete elements, See Remark 2.
ISDES	Section ID for discrete elements, See Remark 2.
RSF	DES radius scale down factor.
OUTDES	Allow user output generated discrete element nodes and DES properties toa keyword file. EQ.0: `No output.  (Default)` EQ.1: Write data under filename, desvfill.inc

Remarks:

1. **DES Element to Sold Element Ratio.** The DES particles are evenly distributed within the solid element. For hexahedral elements the number of the generated DES particles is NQ × NQ × NQ. For pentahedral elements, the number of generated DES particles is 1, 6, and 18 for NQ = 1, 2, and 3 respectively. For tetrahedral elements, the number generated DES particles is 1, 4, and 10 for NQ = 1, 2, and 3 respectively. See Figure 15-1.

2. **Part ID.** The Part ID for newly generated DES particles can be either a new Part ID or the ID of an existing DES Part.

DEFINE_ADAPTIVE_SOLID_TO_SPH_*{OPTION}

Purpose: Adaptively transform a Lagrangian solid Part or Part Set to SPH particles, when the Lagrangian solid elements comprising those parts fail. One or more SPH particles (elements) will be generated for each failed element. The SPH particles replacing the failed solid Lagrangian elements inherit all the Lagrange nodal quantities and all the Lagrange integration point quantities of these failed solid elements. Those properties are assigned to the newly activated SPH particles. The constitutive properties assigned to the new SPH part will correspond to the MID and EOSID referenced by the SPH *PART definition.

The available options include:

<BLANK>

ID

ID Card. Additional card for the ID keyword option.

Optional	1	2	3	4	5	6	7	8
Variable	DID	HEADING						
Type	I	A70						
Default	none	none						

Card 1	1	2	3	4	5	6	7	8
Variable	IPID	ITYPE	NQ	IPSPH	ISSPH	ICPL	IOPT	CPCD
Type	I	I	I	I	I	I	I	F
Default	none	none	none	none	none	none	none	average

VARIABLE	DESCRIPTION
DID	Definition ID. This must be a unique number.
HEADING	Definition descriptor. It is suggested that unique descriptions be used.

VARIABLE	DESCRIPTION
IPID	ID of the solid part or part set to transform.
ITYPE	IPID type: EQ.0: Part ID, NE.0: Part set ID.
NQ	Adaptive option for hexahedral elements. For tetrahedral and pentahedral elements, see remark 1: EQ.n: Adapt one 8-node solid element to $(n \times n \times n)$ SPH elements. The range of n is from 1 to 8.
IPSPH	Part ID for newly generated SPH elements, See Remark 2.
ISSPH	Section ID for SPH elements, See Remark 2.
ICPL	Coupling of newly generated SPH elements to the adjacent solid elements: EQ.0: Failure without coupling (debris simulation), EQ.1: Coupled to Solid element. EQ.3: Pure thermal coupling between SPH part and Solid part (combined with IOPT = 0 option, See Remark 4).
IOPT	Coupling method (for ICPL = 1 only See Remark 3): EQ.0: Coupling from beginning (used as constraint between SPH elements and Solid elements), EQ.1: Coupling begins when Lagrange element fails.
CPCD	Thermal coupling conductivity between SPH part and Solid part for ICPL = 3 option. The default value is set as the average value of the conductivity from SPH part and the conductivity from Solid part.

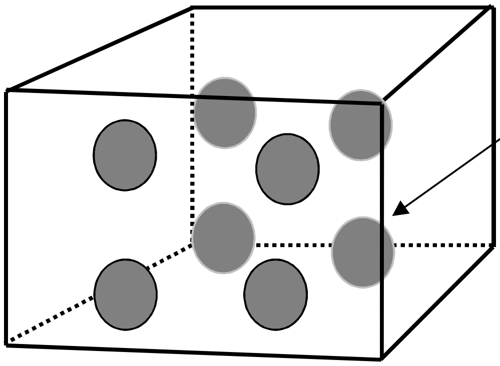
Remarks:

1. The SPH particles are evenly distributed within the solid element. For hexahedral elements the number of the generated SPH particles is NQ*NQ*NQ. For pentahedral elements, the number of generated SPH particles is 1, 6, and 18 for NQ = 1, 2, and 3 respectively. For tetrahedral elements, the number generated SPH particles is 1, 4, and 10 for NQ = 1, 2, and 3 respectively.

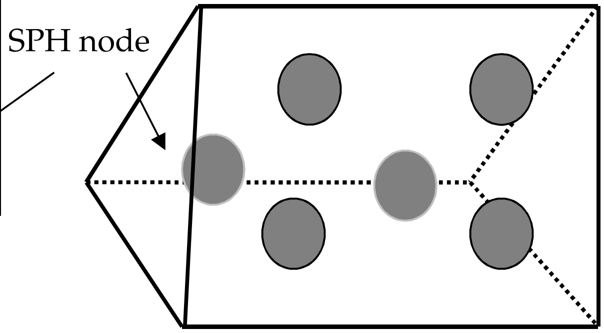
2. The Part ID for newly generated SPH particles can be either a new Part ID or the ID of an existing SPH Part. For constraint coupling (i.e. ICPL = 1 and IOPT = 0), the newly generated SPH part ID should be different from the existing one.

3. ICPL = 0 is used for debris simulation, no coupling happens between newly generated SPH particles and solid elements, the user needs to define node to surface contact for the interaction between those two parts. When ICPL = 1 and IOPT = 1, the newly generated SPH particles are bonded with solid elements as one part through the coupling, and the new material ID with different failure criteria can be applied to the newly generated SPH particles.

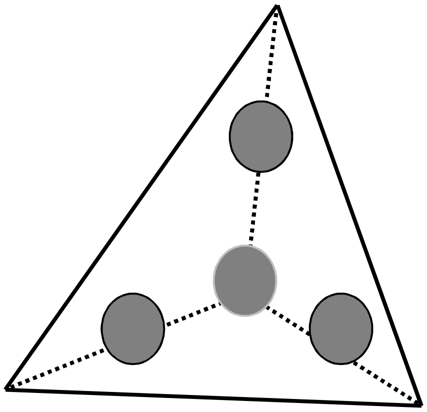
4. ICPL = 3 (combined with IOPT = 0) is used for pure thermal coupling between SPH part and Solid part only. User can define the coupling thermal conductivity value between SPH part and Solid part through CPCD parameter for more realistic thermal coupling between SPH part and Solid part.



Example of SPH nodes for hexahedron element with NQ = 2



Example of SPH nodes for pentahedron element with NQ = 2



Example of SPH nodes for a tetrahedron element with NQ = 2

DEFINE_BOX_*{OPTION}

Available options include:

<BLANK>

LOCAL

Purpose: Define a box-shaped volume. Two diagonally opposite corner points of a box are specified in global or local coordinates if the LOCAL option is active. The box volume is then used for various specifications for a variety of input options, e.g., velocities, contact, etc.

If the option, LOCAL, is active, a local coordinate system with two vectors, see Figure 15-7, is defined. The vector cross product, $z = x \times y$, determines the local z-axis. The local y-axis is then given by $y = z \times x$. A point, X in the global coordinate system is considered to lie with the volume of the box if the coordinate X - C, where C is the global coordinate offset vector defined on Card 3, lies within the box after transformation into the local system, $XC\_local = T \times (X - C)$. The local coordinate, *XC_local,* is checked against the minimum and maximum coordinates defined on Card 1 in the local system. For the *INCLUDE_TRANSFORM options that include translations and rotations, all box options are automatically converted from *DEFINE_BOX_*xxxx* to *DEFINE_BOX_*xxxx*_LOCAL in the DYNA.INC file. Here, *xxxx* represents the box options: ADAPTIVE, COARSEN, and SPH, which are defined below.

Card	1	2	3	4	5	6	7	8
Variable	BOXID	XMN	XMX	YMN	YMX	ZMN	ZMX	
Type	I	F	F	F	F	F	F	
Default	0	0.0	0.0	0.0	0.0	0.0	0.0	

Local Card 1. First additional card for LOCAL keyword option.

Card 2	1	2	3	4	5	6	7	8
Variable	XX	YX	ZX	XV	YV	ZV		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	0.0	0.0	0.0		

Local Card 2. Second additional card for LOCAL keyword option.

Card 3	1	2	3	4	5	6	7	8
Variable	CX	CY	CZ					
Type	F	F	F					
Default	0.0	0.0	0.0					

VARIABLE **DESCRIPTION**

BOXID Box ID. Define unique numbers.

XMN Minimum x-coordinate. Define in the local coordinate system if the option LOCAL is active.

XMX Maximum x-coordinate. . Define in the local coordinate system if the option LOCAL is active.

YMN Minimum y-coordinate. . Define in the local coordinate system if the option LOCAL is active.

YMX Maximum y-coordinate. . Define in the local coordinate system if the option LOCAL is active.

ZMN Minimum z-coordinate. . Define in the local coordinate system if the option LOCAL is active.

ZMX Maximum z-coordinate. . Define in the local coordinate system if the option LOCAL is active.

VARIABLE **DESCRIPTION**

XX X-coordinate on local x-axis. Origin lies at (0,0,0). Define if the LOCAL option is active.

YX Y-coordinate on local x-axis. Define if the LOCAL option is active.

ZX Z-coordinate on local x-axis. Define if the LOCAL option is active.

XV X-coordinate of local x-y vector. Define if the LOCAL option is active.

YV Y-coordinate of local x-y vector. Define if the LOCAL option is active.

ZV Z-coordinate of local x-y vector. Define if the LOCAL option is active.

CX X-global coordinate of offset vector to origin of local system. Define if the LOCAL option is active.

CY Y-global coordinate of offset vector to origin of local system. Define if the LOCAL option is active.

CZ Z-global coordinate of offset vector to origin of local system. Define if the LOCAL option is active.

DEFINE_BOX_ADAPTIVE_*{OPTION}

Available options include:

<BLANK>

LOCAL

Purpose: Define a box-shaped volume enclosing (1) the shells where the h-adaptive level (2) the solids where the tetrahedron r-adaptive mesh size is to be specified. If the midpoint of the element falls within the box, the h-adaptive level is reset. With the additions of LIDX/NDID, LIDY and LIDZ, the box can be made movable; it is also possible to define a fission box followed by a fusion box and the mesh could refine when deformed and coarsen when flattened. Shells falling outside of this volume use the value, MAXLVL, on the *CONTROL_ADAPTIVE control cards. Another related keyword includes: *DEFINE_CURVE_BOX_ADAPTIVITY.

Card 1	1	2	3	4	5	6	7	8
Variable	BOXID	XMN	XMX	YMN	YMX	ZMN	ZMX	
Type	I	F	F	F	F	F	F	
Default	none	0.0	0.0	0.0	0.0	0.0	0.0	

Card 2	1	2	3	4	5	6	7	8
Variable	PID	LEVEL	LIDX/NDID	LIDY	LIDZ	BRMIN	BRMAX	
Type	I	I	I	I	I	F	F	
Default	0	1	0	0	0	0.0	0.0	

Local Card 1. First additional card for LOCAL keyword option. See *DEFINE_BOX for a description of the LOCAL option.

Card 3	1	2	3	4	5	6	7	8
Variable	XX	YX	ZX	XV	YV	ZV		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	0.0	0.0	0.0		

Local Card 2. Second additional card for LOCAL keyword option.

Card 4	1	2	3	4	5	6	7	8
Variable	CX	CY	CZ					
Type	F	F	F					
Default	0.0	0.0	0.0					

VARIABLE	DESCRIPTION
BOXID	Box ID. Define unique numbers.
XMN	Minimum x-coordinate. Define in the local coordinate system if the option LOCAL is active.
XMX	Maximum x-coordinate. Define in the local coordinate system if the option LOCAL is active.
YMN	Minimum y-coordinate. Define in the local coordinate system if the option LOCAL is active.
YMX	Maximum y-coordinate. Define in the local coordinate system if the option LOCAL is active.
ZMN	Minimum z-coordinate. Define in the local coordinate system if the option LOCAL is active.
ZMX	Maximum z-coordinate. Define in the local coordinate system if the option LOCAL is active.

VARIABLE **DESCRIPTION**

PID

Deformable part ID. If zero, all active elements within box are considered.

LEVEL

Maximum number of refinement levels for elements that are contained in the box. Values of 1, 2, 3, 4,... allow a maximum of 1, 4, 16, 64, ... elements, respectively, to be created for each original element.

LIDX/NDID

Load curve ID/Node ID.

GT.0: load curve ID. Define adaptive box movement (displacement vs. time) in global X axis.

LT.0: absolute value is a node ID, whose movement will be followed by the moving adaptive box. The node ID can be on a moving rigid body.

EQ.0: no movement.

LIDY

Load curve ID.

GT.0: load curve ID. Define adaptive box movement (displacement vs. time) in global Y axis.

EQ.0: no movement.

LIDZ

Load curve ID.

GT.0: load curve ID. Define adaptive box movement (displacement vs. time) in global Z axis.

EQ.0: no movement.

BRMIN

Minimum mesh size in 3D tetrahedron adaptivity

BRMAX

Maximum mesh size in 3D tetrahedron adaptivity

XX

X-coordinate on local x-axis. Origin lies at (0,0,0). Define if the LOCAL option is active.

YX

Y-coordinate on local x-axis. Define if the LOCAL option is active.

ZX

Z-coordinate on local x-axis. Define if the LOCAL option is active.

XV

X-coordinate of local x-y vector. Define if the LOCAL option is active.

VARIABLE	DESCRIPTION
YV	Y-coordinate of local x-y vector. Define if the LOCAL option is active.
ZV	Z-coordinate of local x-y vector. Define if the LOCAL option is active.
CX	X-global coordinate of offset vector to origin of local system. Define if the LOCAL option is active.
CY	Y-global coordinate of offset vector to origin of local system. Define if the LOCAL option is active.
CZ	Z-global coordinate of offset vector to origin of local system. Define if the LOCAL option is active.

Remarks:

The moving adaptive box is very useful and efficient in situation where deformation progresses while happening only locally, such as roller hemming and incremental forming simulation. With the moving box feature, elements entering one box can be refined and fused together when they enter another box. Mesh fission outside of the moving box envelope is controlled by MAXLVL and other parameters under *CONTROL_ADAPTIVE. The fusion controls (NCFREQ, IADPCL) can be defined using *CONTROL_ADAPTIVE. Currently, only IADPCL = 1 is supported.

Only when one of the LCIDX/NDID, LICDY, or LCIDZ is defined, the adaptive box will be moving; otherwise it will be stationary.

For 3D tetrahedron r-adaptivity, the current implementation does not support LOCAL option. In card 2, PID, BRMIN and BRMAX are the only parameters currently supported in 3D r-adaptivity.

Example:

Referring to a partial input deck below, and Figure 15-2, a strip of sheet metal is being roller hemmed. The process consists of pre- and final hemming. Each pre- and final roller is defined with a moving adaptive box ID 2 and 3, respectively, with the box shapes shown in Figure 15-2. The first box, a fission box, was set at LEVEL = 3, while the second box, a fusion box, was set at LEVEL = 1. Elements outside of the volume envelope made by the moving boxes undergo no fission and fusion (MAXLVL = 1). This settings allows mesh fission when entering the moving box 2 (LEVEL = 3), fusion only when elements entering the moving box 3 (LEVEL = 1), no fusion/fusion (MAXLVL = 1) at all outside of the volume envelope created by the moving boxes. In the example, the boxes 2 and 3 are to be moved in global X direction for a distance of 398mm defined by load curve 11, and 450mm defined by load curve 12, respectively.

```
*CONTROL_TERMINATION
0.252
*CONTROL_ADAPTIVE
$  ADPFREQ    ADPTOL    ADPOPT    MAXLVL    TBIRTH    TDEATH     LCADP    IOFLAG
   8.05E-4  0.200000         2         1    0.0001.0000E+20         0         1
$  ADPSIZE    ADPASS    IREFLG    ADPENE     ADPTH    MEMORY    ORIENT     MAXEL
  0.300000         1         0       5.0
$  IADPN90              NCFREQ    IADPCL    ADPCTL    CBIRTH    CDEATH
        -1         0         1         1      10.0     0.000      10.30
*DEFINE_BOX_ADAPTIVE
$#   BOXID       XMN       XMX       YMN       YMX       ZMN       ZMX
         2 -10.00000 36.000000 -15.03000  3.991000  1.00E+00 48.758000
$#     PID     LEVEL LIDX/NDID      LIDY      LIDZ
         6         3        11
*DEFINE_BOX_ADAPTIVE
$#   BOXID       XMN       XMX       YMN       YMX       ZMN       ZMX
         3 -100.0000  -60.0000 -15.03000  3.991000  1.00E+00 48.758000
$#     PID     LEVEL LIDX/NDID      LIDY      LIDZ
         6         1        12
*DEFINE_CURVE
11
                 0.000                 0.0
            0.00100000                 1.0
            0.19900000               397.0
            0.20000000               398.0
                 1.000               398.0
                     ⋮                   ⋮
*DEFINE_CURVE
12
                   0.0                 0.0
                  0.05                 0.0
                 0.051                 1.0
                 0.251               401.0
                 0.252               450.0
                     ⋮                   ⋮
```

A moving box can also follow the movement of a node, which can be on a moving rigid body. In this case, load curves defining the boxes' movement can be skipped, instead, NDIDs for the boxes should be defined. For example, in Figure 15-2 and a partial keyword example below, box 2 is to follow a node (ID: 33865) on the pre-roller, and box 3 to follow another node (ID: 38265) on the final roller.

```
*DEFINE_BOX_ADAPTIVE
$#   BOXID       XMN       XMX       YMN       YMX       ZMN       ZMX
         2 -10.00000 36.000000 -15.03000  3.991000  1.00E+00 48.758000
$#     PID     LEVEL LIDX/NDID      LIDY       LID
         6         3    -33865
*DEFINE_BOX_ADAPTIVE
$#   BOXID       XMN       XMX       YMN       YMX       ZMN       ZMX
         3 -100.0000  -60.0000 -15.03000  3.991000  1.00E+00 48.758000
$#     PID     LEVEL LIDX/NDID      LIDY       LID
         6         3    -38265
```

Revision information:

The variables LIDX/NDID, LIDY, LIDZ are available in both SMP and MPP starting in Revision 98718.

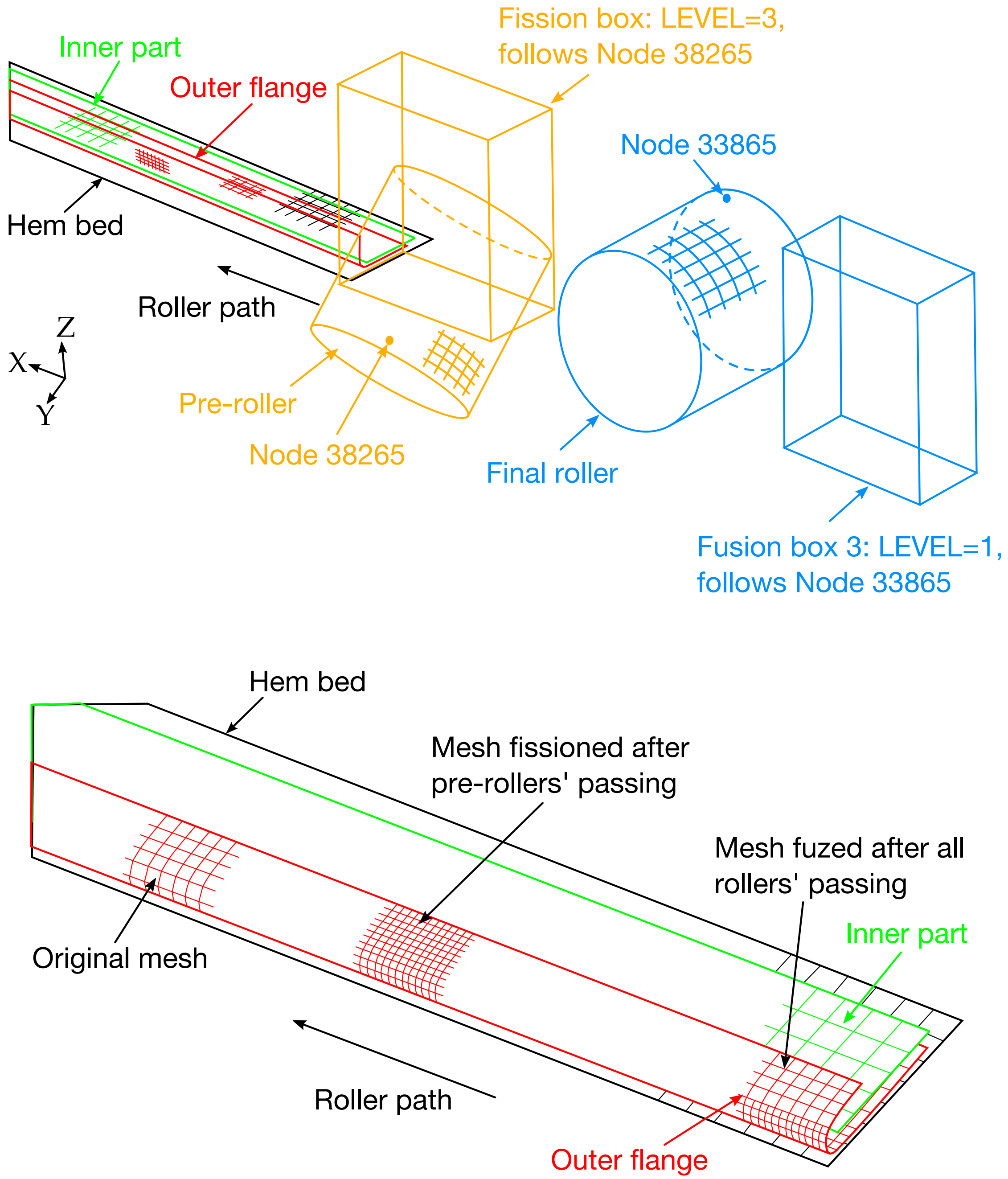


Figure 15-2. Defining mesh fission and fusion.

DEFINE_BOX_COARSEN_*{OPTION}

Available options include:

<BLANK>

LOCAL

Purpose: Define a specific box-shaped volume indicating elements which are protected from mesh coarsening. See also *CONTROL_COARSEN.

Card	1	2	3	4	5	6	7	8
Variable	BOXID	XMN	XMX	YMN	YMX	ZMN	ZMX	IFLAG
Type	I	F	F	F	F	F	F	I
Default	none	0.0	0.0	0.0	0.0	0.0	0.0	0

Local Card 1. First additional card for LOCAL keyword option. See *DEFINE_BOX for a description of the LOCAL option.

Card 2	1	2	3	4	5	6	7	8
Variable	XX	YX	ZX	XV	YV	ZV		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	0.0	0.0	0.0		

Local Card 2. Second additional card for LOCAL keyword option.

Card 3	1	2	3	4	5	6	7	8
Variable	CX	CY	CZ					
Type	F	F	F					
Default	0.0	0.0	0.0					

VARIABLE **DESCRIPTION**

BOXID Box ID. Define unique numbers.

XMN Minimum x-coordinate. Define in the local coordinate system if the option LOCAL is active.

XMX Maximum x-coordinate. . Define in the local coordinate system if the option LOCAL is active.

YMN Minimum y-coordinate. . Define in the local coordinate system if the option LOCAL is active.

YMX Maximum y-coordinate. . Define in the local coordinate system if the option LOCAL is active.

ZMN Minimum z-coordinate. . Define in the local coordinate system if the option LOCAL is active.

ZMX Maximum z-coordinate. . Define in the local coordinate system if the option LOCAL is active.

IFLAG Flag for protecting elements inside or outside of box.

EQ.0: elements inside the box cannot be coarsened

EQ.1: elements outside the box cannot be coarsened

XX X-coordinate on local x-axis. Origin lies at (0,0,0). Define if the LOCAL option is active.

YX Y-coordinate on local x-axis. Define if the LOCAL option is active.

ZX Z-coordinate on local x-axis. Define if the LOCAL option is active.

XV X-coordinate of local x-y vector. Define if the LOCAL option is active.

YV Y-coordinate of local x-y vector. Define if the LOCAL option is active.

ZV Z-coordinate of local x-y vector. Define if the LOCAL option is active.

CX X-global coordinate of offset vector to origin of local system. Define if the LOCAL option is active.

VARIABLE **DESCRIPTION**

CY Y-global coordinate of offset vector to origin of local system. Define if the LOCAL option is active.

CZ Z-global coordinate of offset vector to origin of local system. Define if the LOCAL option is active.

Remarks:

1. Many boxes may be defined. If an element is protected by any box then it may not be coarsened.

*DEFINE_BOX_DRAWBEAD

Purpose: Define a specific box or tube shaped volume around a draw bead. This option is useful for the draw bead contact. If box shaped, the volume will contain the draw bead nodes and elements between the bead and the outer edge of the blank. If tubular, the tube is centered around the draw bead. All elements within the tubular volume are included in the contact definition.

Card	1	2	3	4	5	6	7	8
Variable	BOXID	PID	SID	IDIR	STYPE	RADIUS	CID	
Type	I	F	F	F	I	F	I	
Default	0	0.0	0.0	0.0	4	0.0	0	
Remarks						optional	optional	

VARIABLE	DESCRIPTION
BOXID	Box ID. Define unique numbers.
PID	Part ID of blank.
SID	Set ID that defines the nodal points that lie along the draw bead. If a node set is defined, the nodes in the set must be consecutive along the draw bead. If a part or part set is defined, the set must consist of beam or truss elements. Within the part set, no ordering of the elements is assumed, but the number of nodes must equal the number of beam elements plus 1.
IDIR	Direction of tooling movement. The movement is in the global coordinate direction unless the tubular box option is active and CID is nonzero. In this latter case, the movement is in the local coordinate direction. EQ.1: tooling moves in x-direction, EQ.2: tooling moves in y-direction, EQ.3: tooling moves in z-direction.

VARIABLE	DESCRIPTION
STYPE	Set type: EQ.2: part set ID, EQ.3: part ID, EQ.4: node set ID.
RADIUS	The radius of the tube, which is centered around the draw bead. Elements of part ID, PID, that lie within the tube will be included in the contact. If the radius is not defined, a rectangular box is used instead. This option is recommended for curved draw beads and for draw beads that are not aligned with the global axes.
CID	Optional coordinate system ID. This option is only available for the tubular drawbead.

DEFINE_BOX_SPH_*{OPTION}

Available options include:

<BLANK>

LOCAL

Purpose: Define a box-shaped volume. Two diagonally opposite corner points of a box are specified in global coordinates. Particle approximations of SPH elements are computed when particles are located inside the box. The load curve describes the motion of the maximum and minimum coordinates of the box.

Card 1	1	2	3	4	5	6	7	8
Variable	BOXID	XMN	XMX	YMN	YMX	ZMN	ZMX	VID
Type	I	F	F	F	F	F	F	I
Default	none	0.0	0.0	0.0	0.0	0.0	0.0	0

Card 2	1	2	3	4	5	6	7	8
Variable	LCID	VD	NID					
Type	I	I	I					
Default	0	0	0					

Local Card 1. First additional card for LOCAL keyword option. See *DEFINE_BOX for a description of the LOCAL option

Card 3	1	2	3	4	5	6	7	8
Variable	XX	YX	ZX	XV	YV	ZV		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	0.0	0.0	0.0		

Local Card 2. Second additional card for LOCAL keyword option.

Card 4	1	2	3	4	5	6	7	8
Variable	CX	CY	CZ					
Type	F	F	F					
Default	0.0	0.0	0.0					

VARIABLE	DESCRIPTION
BOXID	Box ID. Define unique numbers.
XMN	Minimum x-coordinate. Define in the local coordinate system if the option LOCAL is active.
XMX	Maximum x-coordinate. Define in the local coordinate system if the option LOCAL is active.
YMN	Minimum y-coordinate. Define in the local coordinate system if the option LOCAL is active.
YMX	Maximum y-coordinate. Define in the local coordinate system if the option LOCAL is active.
ZMN	Minimum z-coordinate. Define in the local coordinate system if the option LOCAL is active.
ZMX	Maximum z-coordinate. Define in the local coordinate system if the option LOCAL is active.

VARIABLE **DESCRIPTION**

VID Vector ID for DOF, see *DEFINE_VECTOR.

LCID Load curve ID to describe motion value versus time, see *DEFINE_CURVE

VD Velocity/Displacement flag:

EQ.0: velocity,

EQ.1: displacement,

EQ.2: referential node

NID Referential nodal ID for VD = 2 (SPH box will move with this node).

XX X-coordinate on local x-axis. Origin lies at (0,0,0). Define if the LOCAL option is active.

YX Y-coordinate on local x-axis. Define if the LOCAL option is active.

ZX Z-coordinate on local x-axis. Define if the LOCAL option is active.

XV X-coordinate of local x-y vector. Define if the LOCAL option is active.

YV Y-coordinate of local x-y vector. Define if the LOCAL option is active.

ZV Z-coordinate of local x-y vector. Define if the LOCAL option is active.

CX X-global coordinate of offset vector to origin of local system. Define if the LOCAL option is active.

CY Y-global coordinate of offset vector to origin of local system. Define if the LOCAL option is active.

CZ Z-global coordinate of offset vector to origin of local system. Define if the LOCAL option is active.

DEFINE_CONNECTION_PROPERTIES_*{OPTION}

Available options include:

<BLANK>

ADD

Purpose: Define failure related parameters for solid element spot weld failure by *MAT_SPOTWELD_DAIMLERCHRYSLER. For each connection identifier, CON_ID, a separate *DEFINE_CONNECTION_PROPERTIES section must be included. The **ADD** option allows material specific properties to be added to an existing connection ID. See Remark 2.

Card 1	1	2	3	4	5	6	7	8
Variable	CON_ID	PRUL	AREAEQ		DGTYP	MOARFL		
Type	F	I	I		I	I		
Default	0	0	0		0	0		

Card 2	1	2	3	4	5	6	7	8
Variable		DSIGY	DETAN	DDGPR	DRANK	DSN	DSB	DSS
Type		F	F	F	F	F	F	F
Default		none	none	10^{10}	none	none	none	none

Card 3	1	2	3	4	5	6	7	8
Variable	DEXSN	DEXSB	DEXSS	DLCSN	DLCSB	DLCSS	DGFAD	DSCLMRR
Type	F	F	F	I	I	I	F	F
Default	1.0	1.0	1.0	0	0	0	none	1.0

Material Specific Data:

For each shell material with material specific data, define for this CON_ID the following *two* cards. Add as many pairs of cards as necessary. This input is terminated by the next keyword ("*") card.

Material Data Card 1.

Card 4	1	2	3	4	5	6	7	8
Variable	MID	SGIY	ETAN	DGPR	RANK	SN	SB	SS
Type	F	F	F	F	F	F	F	F
Default				10^{10}				

Material Data Card 2.

Card 5	1	2	3	4	5	6	7	8
Variable	EXSN	EXSB	EXSS	LCSN	LCSB	LCSS	GFAD	SCLMRR
Type	F	F	F	I	I	I	F	F
Default								1.0

VARIABLE	DESCRIPTION
CON_ID	Connection ID, referenced on *MAT_SPOTWELD_DAIMLER-CHRYSLER. Multiple sets of connection data may be used by assigning different connection IDs.

VARIABLE	DESCRIPTION
PRUL	The failure rule number for this connection. EQ.1: Use data of weld partner with lower RANK (default). GE.2: Use DEFINE_FUNCTION expressions to determine weld data depending on several values of both weld partners. Variables DSIGY, DETAN, DDGPR, DSN, DSB, DSS, DEXSN, DEXSB, DEXSS, and DGFAD must be defined as function IDs, see Remark 5.
AREAEQ	Area equation number for the connection area calculation. EQ.0: (default) $\text{Area}_{\text{true}} = \text{Area}_{\text{modelled}}$ EQ.1: millimeter form; see Remark 4 EQ.-1: meter form; see Remark 4
DGTYP	Damage type EQ.0: no damage function is used EQ.1: strain based damage EQ.2: failure function based damage EQ.3 or 4: fading energy based damage; see Remark 4
MOARFL	Modeled area flag EQ.0: $\text{Area}_{\text{modelled}}$ goes down with shear (default) EQ.1: $\text{Area}_{\text{modelled}}$ stays constant
DSIGY	Default yield stress for the spot weld element.
DETAN	Default tangent modulus for the spot weld element.
DDGPR	Default damage parameter for hyperbolic based damage function.
DRANK	Default rank value.
DSN	Default normal strength.
DSB	Default bending strength.
DSS	Default shear strength.
DEXSN	Default exponent on normal stress term.
DEXSB	Default exponent on bending stress term.

VARIABLE	DESCRIPTION
DEXSS	Default exponent on shear stress term.
DLCSN	Default curve ID for normal strength scale factor as a function of strain rate.
DLCSB	Default curve ID for bending strength scale factor as a function of strain rate.
DLCSS	Default curve ID for shear strength scale factor as a function of strain rate.
DGFAD	Default fading energy for damage type 3 and type 4.
DSCLMRR	Default scaling factor for torsional moment in failure function.
MID	Material ID of the shell material for which properties are defined.
SIGY	Yield stress to be used in the spot weld element calculation.
ETAN	Tangent modulus to be used in the spot weld element calculation.
DGPR	Damage parameter for hyperbolic based damage function.
RANK	Rank value. See Remark 4.
SN	Normal strength.
SB	Bending strength.
SS	Shear strength.
EXSN	Exponent on normal stress term.
EXSB	Exponent on bending stress term.
EXSS	Exponent on shear stress term.
LCSN	Curve ID for normal strength scale factor as a function of strain rate.
LCSB	Curve ID for bending strength scale factor as a function of strain rate.
LCSS	Curve ID for shear strength scale factor as a function of strain rate.

VARIABLE	DESCRIPTION
GFAD	Fading energy for damage type 3 and 4.
SCLMRR	Scaling factor for torsional moment in failure function.

Remarks:

1. **Restriction to *MAT_SPOTWELD_DAIMLERCHHRYSLER.** This keyword is used only with *MAT_SPOTWELD_DAIMLERCHRYSLER. The data input is used in a 3 parameter failure model. Each solid spot weld element connects shell elements that may have the same or different materials. The failure model assumes that failure of the spot weld depends on the properties of the welded materials, so this keyword allows shell material specific data to be input for the connection. The default data will be used for any spot weld connected to a shell material that does not have material specific data defined, so it is not necessary to define material specific data for all welded shell materials.

2. **ADD Option.** To simplify data input, the ADD keyword option allows material specific data to be added to an existing *DEFINE_CONNECTION_PROPERTIES table. To use the ADD option, omit cards 2 and 3, and input only CON_ID on card 1. Then use cards 4 and 5 to input material specific data. For each unique CON_ID, control parameters and default values must be input in one set of *DEFINE_CONNECTION_PROPERTIES data. The same CON_ID may be used for any number of sets of material specific data input with the ADD option.

3. **The Three Parameter Failure Function.** The three parameter failure function is

$$f = \left(\frac{\sigma_n}{\sigma_n^F}\right)^{m_n} + \left(\frac{\sigma_b}{\sigma_b^F}\right)^{m_b} + \left(\frac{\tau}{\tau^F}\right)^{m_\tau} - 1\ ,$$

where the three strength terms are SN, SB, and SS, and the three exponents are EXSN, EXSB, and EXSS. The strengths may be a function of strain rate by using the load curves, LCSN, LCSB, and LCSS. The peak stresses in the numerators are calculated from force resultants and simple beam theory.

$$\sigma_n = \frac{N_{rr}}{A}, \qquad \sigma_b = \frac{\sqrt{M_{rs}^2 + M_{rt}^2}}{Z}, \qquad \tau = \mathrm{SCLMRR} \times \frac{M_{rr}}{2Z} + \frac{\sqrt{N_{rs}^2 + N_{rt}^2}}{A}$$

where the area is the cross section area of the weld element and Z is given by:

$$Z = \pi \frac{d^3}{32}$$

where d is the equivalent diameter of the solid spot weld element assuming a circular cross section.

4. **Control Parameters PRUL, AREAQ. And DGTYP.** There are three control parameters that define how the table data will be used for the connection, PRUL, AREAEQ, and DGTYP. PRUL determines how the parameters will be used. Because each weld connects two shell surfaces, one weld can have two sets of failure data as well as two values for ETAN and SIGY. For PRUL=1 (default), a simple rule is implemented and the data with the lower RANK will be used. For PRUL=2 or 3, function expressions can be used to determine the data based on several input values from both weld partners (see Remark 5 for details).

   The second control parameter is AREAEQ which specifies a rule for calculating a true weld cross section area, A_true to be used in the failure function in place of the modeled solid element area, A. For AREAEQ = 1, A_true is calculated by

$$A_\text{true} = \frac{\pi}{4}\left(5\sqrt{t_\text{min shell}}\right)^2$$

   where $t_\text{min shell}$ is the thickness of the welded shell surface that has the smaller thickness. For AREAEQ = −1, A_true is calculated by

$$A_\text{true} = \frac{\pi}{4}\left(\frac{5}{1000}\sqrt{1000 \times t_\text{min shell}}\right)^2$$

   The equation for AREAEQ = 1 is valid only for a length unit of millimeters, and AREAEQ = −1 is valid only for a length unit of meters.

   The third control parameter, DGTYP, chooses from two available damage types. For DGTYP = 0, damage is turned off and the weld fails immediately when $f \geq 0$. For DGTYP > 0, damage is initiated when $f \geq 0$ and complete failure occurs when $\omega \geq 1$. For DGTYP = 1, damage growth is a function of plastic strain:

$$\omega = \frac{\varepsilon^p_\text{eff} - \varepsilon^p_\text{failure}}{\varepsilon^p_\text{rupture} - \varepsilon^p_\text{failure}}, \qquad \varepsilon^p_\text{failure} \leq \varepsilon^p_\text{eff} \leq \varepsilon^p_\text{rupture}$$

   where ε^p_eff is the effective plastic strain in the weld material. When the value of the failure function first exceeds zero, the plastic strain at failure$\varepsilon^p_\text{failure}$ is set to the current plastic strain, and the rupture strain is offset from the plastic strain at failure by

$$\varepsilon^p_\text{rupture} = \varepsilon^p_\text{failure} + \text{RS} - \text{EFAIL}$$

   where RS and EFAIL are the rupture strain and plastic strain at failure which are input on the *MAT_SPOTWELD_DAIMLERCHRYSLER card. If failure occurs when the plastic strain is zero, the weld material yield stress is reduced to the current effective stress such that damage can progress.

   For DGTYP = 2, damage is a function of the failure function, f:

$$f \geq 0 \Rightarrow \omega = \frac{f}{f_{\text{rupture}}}$$

where f_{rupture} is the value of the failure function at rupture which is defined by

$$f_{\text{rupture}} = \text{RS} - \text{EFAIL}$$

and RS and EFAIL are input on the *MAT_SPOTWELD_DAIMLERCHRYSLER card.

Because the DGTYP = 1 damage function is scaled by plastic strain, it will monotonically increase in time. The DGTYP = 2 damage function is forced to be a monotonically increasing function in time by using the maximum of the current value and the maximum previous value. For both DGTYP = 1 and DGTYP = 2, the stress scale factor is then calculated by

$$\hat{\sigma} = \frac{\text{DGPR} \times (1 - \omega)}{\omega\left(\frac{1}{2} + \sqrt{\frac{1}{4} + \text{DGPR}}\right) + \text{DGPR}}\sigma$$

This equation becomes nearly linear at the default value of DGPR which is 10^{10}.

For DGTYP = 3, damage is a function of total strain:

$$\omega = \frac{\Delta\varepsilon_n}{\Delta\varepsilon_{\text{fading}}}$$

where $\Delta\varepsilon_n$ is the accumulated total strain increment between moment of damage initiation (failure) and current time step t_n

$$\Delta\varepsilon_n = \Delta\varepsilon_{n-1} + \Delta t_n\sqrt{\frac{2}{3}\ \text{tr}(\dot{\boldsymbol{\varepsilon}}_n\dot{\boldsymbol{\varepsilon}}_n^T)}\ , \qquad \Delta\varepsilon|_{t_{\text{failure}}} = 0$$

and $\Delta\varepsilon_{fading}$ is the total strain increment for fading (reduction of stresses to zero)

$$\Delta\varepsilon_{\text{fading}} = \frac{2 \times \text{GFAD}}{\sigma_{\text{failure}}}$$

where GFAD is the fading energy from input and σ_{failure} is the effective stress at failure. The stress scale factor is then calculated by a linear equation

$$\hat{\sigma} = (1 - \omega)\sigma$$

where σ is the Cauchy stress tensor at failure and ω is the actual damage value. Problems can occur, if the loading direction changes after the onset of failure, since during the damage process, the components of the stress tensor are kept constant and hence represent the stress state at failure.

Therefore DGTYP = 4 should be used describing the damage behavior of the spotweld in a more realistic way. For DGTYP = 4, damage is a function of the internal work done by the spotweld after failure,

$$\hat{\sigma} = (1-\omega)\sigma^{\mathrm{ep}}, \quad \omega = \frac{G_{\mathrm{used}}}{2\times \mathrm{GFAD}}, \quad G_{\mathrm{used}} = G_{\mathrm{used}}^{n-1} + \det\left(F_{ij}\sigma_{ij}^{\mathrm{ep}}\Delta\varepsilon_{ij}\right)$$

Therein, F_{ij} is the deformation gradient. σ^{ep} is a scaled Cauchy stress tensor based on the undamaged Cauchy stress tensor σ^{wd} and scaled in such a way that the same internal work is done in the current time step as in the time step before (equipotential):

$$\sigma^{\mathrm{ep}} = \alpha\sigma^{\mathrm{wd}}, \quad \alpha = \frac{\sigma_{ij}^{n-1,\mathrm{ep}}\Delta\varepsilon_{ij}}{\sigma_{ij}^{\mathrm{wd}}\Delta\varepsilon_{ij}}$$

5. **Failure Rule from *DEFINE_FUNCTION.** The failure rule number PRUL = 2 or 3, is available starting with Release R7. To use this new option, 11 variables have to be defined as function IDs: DSIGY, DETAN, DDGPR, DSN, DSB, DSS, DEXSN, DEXSB, DEXSS, DGFAD, and DSCLMRR.

These functions depend on:

$(\mathrm{t1}, \mathrm{t2})$ = thicknesses of both weld partners

$(\mathrm{sy1}, \mathrm{sy2})$ = initial yield stresses at plastic strain

$(\mathrm{sm1}, \mathrm{sm2})$ = maximum engineering yield stresses

r = strain rate

a = spot weld area

For DSIGY = 100 Such a function could look like:

```
*DEFINE_FUNCTION
       100
 func(t1,t2,sy1,sy2,sm1,sm2,r,a)=0.5*(sy1+sy2)
```

All the listed arguments in their correct order must be included in the argument list. For PRUL = 2, the thinner part is the first weld partner. For PRUL = 3, the bottom part (nodes 1-2-3-4) is the first weld partner. Since material parameters have to be identified from both weld partners during initialization, this feature is only available for a subset of material models at the moment, namely material types 24, 120, 123, and 124. This new option eliminates the need for the ADD option.

*DEFINE_CONSTRUCTION_STAGES

Purpose: Define times and durations of construction stages.

Card	1	2	3	4	5	6	7	8
Variable	ISTAGE	ATS	ATE	ATR	RTS	RTE		
Type	I	F	F	F	F	F		
Default	none	0.0	0.0	none	ATS	ATE		

VARIABLE	DESCRIPTION
ISTAGE	Stage ID
ATS	Analysis time at start of stage
ATE	Analysis time at end of stage
ATR	Analysis time duration of ramp
RTS	Real time at start of stage
RTE	Real time at end of stage

Remarks:

See also *CONTROL_CONSTRUCTION_STAGES and *DEFINE_STAGED_CONSTRUCTION_PART.

The first stage should start at time zero. There must be no gaps between stages, i.e. ATS for each stage must be the same as ATE for the previous stage.

The ramp time allows gravity loading and part stiffening/removal to be applied gradually during the first time period ATR of the construction stage.

The analysis always runs in “analysis time” – typically measured in seconds. The “real time” is used only as a number to appear on output plots and graphs, and is completely arbitrary. A dynain file is written at the end of each stage.

*DEFINE_CONTACT_EXCLUSION

Purpose: Exclude tied nodes from being treated in specific contact interfaces. This keyword is currently only available in the MPP version.

Card 1	1	2	3	4	5	6	7	8
Variable	EID	Title						
Type	I	A70						

ID Card 1. This card sets the contact interface the ids of up to 7 tied interfaces.

Card 2	1	2	3	4	5	6	7	8
Variable	Target	C1	C2	C3	C4	C5	C6	C7
Type	I	I	I	I	I	I	I	I

Optional ID Cards. More tied interfaces. Include as many cards as necessary.

Card 2	1	2	3	4	5	6	7	8
Variable	C8	C9	C10	C11	C12	C13	C14	C15
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
EID	Exclusion ID
Title	Exclusion Title
Target	Contact interface from which tied nodes are to be excluded. This must be the ID of a SINGLE_SURFACE, NODE_TO_SURFACE, or SURFACE_TO_SURFACE contact with SOFT ≠ 2.

VARIABLE	DESCRIPTION
C*i*	The IDs of TIED contacts: 7 on the first card and 8 per additional card for as many cards as necessary. Any node which is a slave node in one of these interfaces, and is in fact tied, will not be processed (as a slave node) in the Target interface. Note that if a node is excluded from the Target by this mechanism, contact forces may still be applied to the node due to any slave or master nodes impacting the contact segments of which it is a part (no contact SEGMENTS are deleted, only contact NODES). If the Target contact is of type SURFACE_TO_SURFACE, any tied slave nodes are deleted from both the slave side (for the normal treatment) and the master side (for the symmetric treatment).

*DEFINE_CONTACT_VOLUME

Purpose: Define a rectangular, a cylindrical, or a spherical volume in a local coordinate system. The volume can be referenced by *SET_NODE_GENERAL for the purpose of defining a node set consisting of nodes inside the volume, or by *CONTACT_... for the purpose of defining nodes or segments on the slave side or the master side of the contact (see SBOXID and MBOXID on Card 1 of *CONTACT_...).

Card 1	1	2	3	4	5	6	7	8
Variable	CVID	CID	TYPE	XC	YC	ZC		
Type	I	I	I	F	F	F		
Default	0	0	0	0.	0.	0.		

Card 2 for Rectangular Prism. Use when type = 0.

Card 2	1	2	3	4	5	6	7	8
Variable	XMN	XMX	YMN	YMX	ZMN	ZMX		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	0.0	0.0	0.0		

Card 2 for Cylinder. Use when type = 1.

Card 2	1	2	3	4	5	6	7	8
Variable	LENGTH	RINNER	ROUTER	D_ANGC				
Type	F	F	F	F				
Default	0.0	0.0	0.0	0.0				

Card 2 for Sphere Use when type = 3.

Card 2	1	2	3	4	5	6	7	8
Variable	RINNER	ROUTER	D_ANGS					
Type	F	F	F					
Default	0.0	0.0	0.0					

VARIABLE	DESCRIPTION
CVID	Contact volume ID
CID	Coordinate system ID. Required for rectangular and cylindrical volumes
TYPE	Volume type. Set to 0 for rectangular, 1 for cylindrical, and 2 for spherical.
XC	x-coordinate which defines the origin of coordinate system or the center of the sphere for type = 3 referenced to the global coordinate system.
YC	y-coordinate which defines the origin of coordinate system or the center of the sphere for type = 3 referenced to the global coordinate system.
ZC	z-coordinate which defines the origin of coordinate system or the center of the sphere for type = 3 referenced to the global coordinate system.
XMN	Minimum x-coordinate in local coordinate system.
XMX	Maximum x-coordinate in local coordinate system.
YMN	Minimum y-coordinate in local coordinate system.
YMX	Maximum y-coordinate in local coordinate system.
ZMN	Minimum z-coordinate in local coordinate system.
ZMX	Maximum z-coordinate in local coordinate system.

VARIABLE	DESCRIPTION
LENGTH	Length of cylinder originating at (XC,YC,ZC) and revolving around the local x-axis.
RINNER	Inner radius of cylinder or sphere.
ROUTER	Outer radius of cylinder or sphere.
D_ANGC	If the included angle between the axis of the cylinder and the normal vector to the contact segment is *less* than this angle, the segment is deleted.
D_ANGS	If the included angle between a line draw from the center of the sphere to the centroid of the segment, and the normal vector to the contact segment is *greater* than this angle, the segment is deleted.

*DEFINE_COORDINATE_NODES

Purpose: Define a local coordinate system with three node numbers. The local cartesian coordinate system is defined in the following steps. If the primary direction is along the x-axis, then the z-axis is computed from the cross product of x and $\bar{y}$, (see Figure 15-3), $z = x \times \bar{y}$, then the y-axis is computed via $y = z \times x$. A similar procedure applies if the local axis is along the y or z axes.

Card	1	2	3	4	5	6	7	8
Variable	CID	N1	N2	N3	FLAG	DIR		
Type	I	I	I	I	I	A		
Default	0	0	0	0	0	X		

VARIABLE	DESCRIPTION
CID	Coordinate system ID. A unique number has to be defined.
N1	ID of node located at local origin.
N2	ID of node located along local x-axis if DIR = X, the y-axis if DIR = Y, and along the z axis if DIR = Z.
N3	ID of node located in local x-y plane if DIR = X, the local y-z plane if DIR = Y, and the local z-x plane if DIR = Z.
FLAG	Set to unity, 1, if the local system is to be updated each time step. Generally, this option when used with nodal SPC's is *not recommended* since it can cause excursions in the energy balance because the constraint forces at the node may go through a displacement if the node is partially constrained.
DIR	Axis defined by node N2 moving from the origin node N1. The default direction is the x-axis.

Remarks:

1. The nodes N1, N2, and N3 must be separated by a reasonable distance and not colinear to avoid numerical inaccuracies.

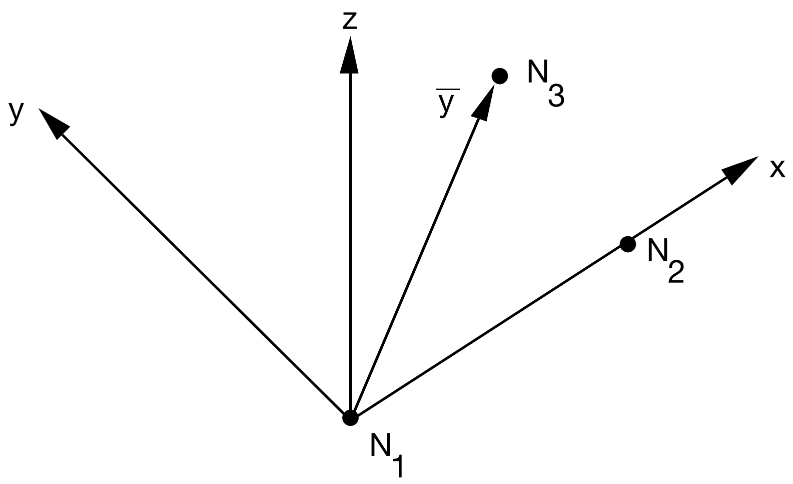


Figure 15-3. Definition of local coordinate system using three nodes when the node N2 lies along the x-axis.

DEFINE_COORDINATE_SYSTEM_*{OPTION}

Available options include:

<BLANK>

IGES

Purpose: Define a local coordinate system.

This card implements the same method as *DEFINE_COORDINATE_NODES; but, instead of reading coordinate positions from nodal IDs, it directly reads the three coordinates from its data cards as Cartesian triples.

When the IGES option is active, LS-DYNA will generate the coordinate system from an IGES file containing three straight curves representing the x, y, and z axes. See remark 4.

Card 1 for <BLANK> Keyword Option.

Card 1	1	2	3	4	5	6	7	8
Variable	CID	X0	Y0	Z0	XL	YL	ZL	CIDL
Type	I	F	F	F	F	F	F	I
Default	0	0.0	0.0	0.0	0.0	0.0	0.0	0

Card 2 for <BLANK> Keyword Option.

Card 2	1	2	3	4	5	6	7	8
Variable	XP	YP	ZP					
Type	F	F	F					
Default	0.0	0.0	0.0					

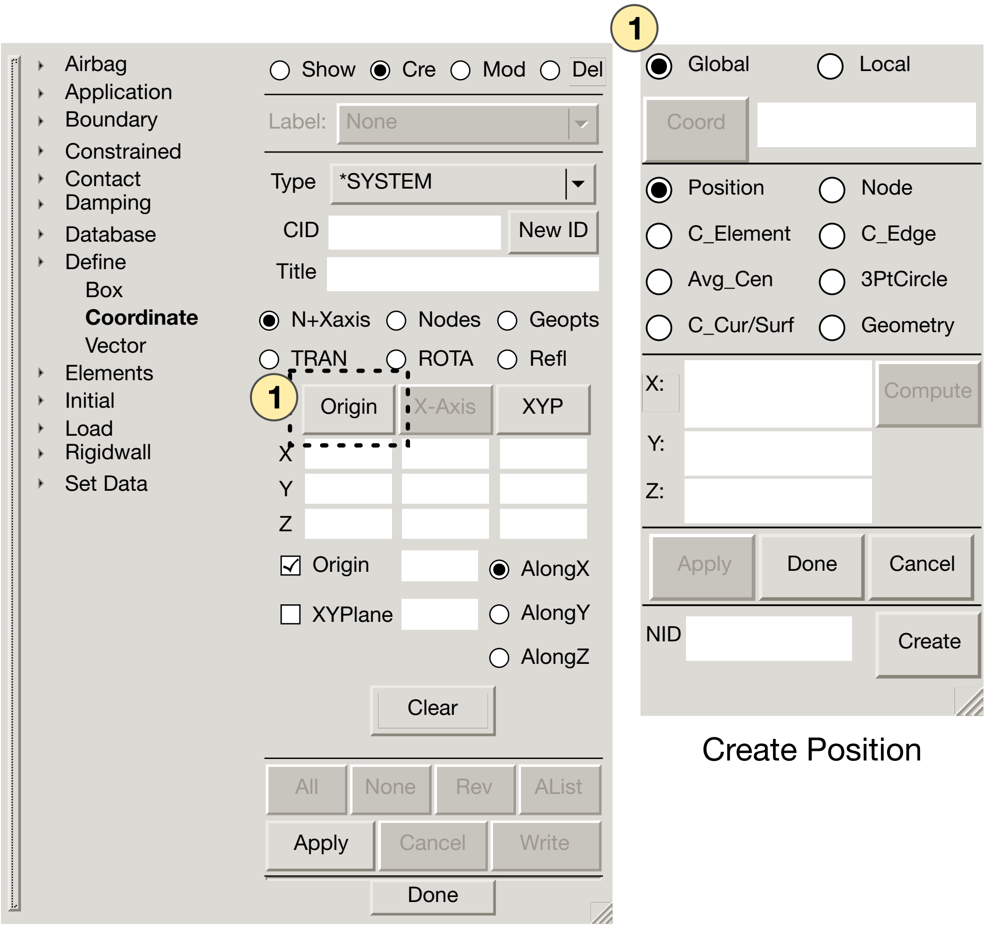
Card 1 for IGES Keyword Option.

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	C							
Default	none							

VARIABLE	DESCRIPTION
CID	Coordinate system ID. A unique number has to be defined.
XO	X-coordinate of origin.
YO	Y-coordinate of origin.
ZO	Z-coordinate of origin.
XL	X-coordinate of point on local x-axis.
YL	Y-coordinate of point on local x-axis.
ZL	Z-coordinate of point on local x-axis.
CIDL	Coordinate system ID applied to the coordinates used to define the current system. The coordinates X0, Y0, Z0, XL, YL, ZL, XP, YP, and ZP are defined with respect to the coordinate system CIDL.
XP	X-coordinate of point in local x-y plane.
YP	Y-coordinate of point in local x-y plane.
ZP	Z-coordinate of point in local x-y plane.
FILENAME	Name of the IGES file containing three curves (see **Remarks** below).

Remarks:

1. The coordinates of the points must be separated by a reasonable distance and not co-linear to avoid numerical inaccuracies.



Create Position

Create Entity

Figure 15-4. *LS-PrePost4.0* Dialog for defining a coordinate system.

2. Care must be taken to avoid chains of coordinate transformations because there is no guarantee that they will be executed in the correct order.

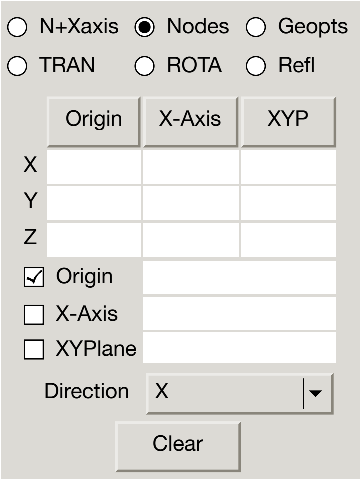
3. **LS-PrePost.** A coordinate system can be created using the dialog box located at *Model* (main window) → *CreEnt* → *Define* (see the left pane) → *Coordinate*. This will activate a *Define Coordinate* dialog in the right pane. Select the *Cre* radio button at the top of the right pane, and set the *type* dropdown to **SYSTEM*. The next set of radio buttons (below the title input box) sets the method used to define the coordinate system. See Figure 15-4.

   a) The *N+Xaxis* method generates a coordinate system from based on:

      i) a user specified origin,

      ii) one of the three global axes (this is a *severe* restriction), and

      iii) a 3rd point.



○ N+Xaxis ● Nodes ○ Geopts

○ TRAN ○ ROTA ○ Refl

☑ Origin

□ X-Axis

□ XYPlane

Direction X

Figure 15-5. Subset of *Create Entity* dialog for both *Nodes* and *Geopts* methods.

The 3rd point, together with the specified global axis defines the new system's x-y plane. The remaining axes are derived using orthogonality and right-handedness. This method requires the user to pick two points which involves the *Create Position* dialog box, as shown in the left frame of Figure 15-4.

NOTE: After defining each point in the *Create Position* dialog, it is *very important* to use the *done* button. The *Create Entity* dialog stays up and remains interactive while the *Create Position* dialog is also up and interactive. This can be confusing. Returning to the *Create Entity* dialog without choosing *done* is a common mistake.

b) The *node* method generates a coordinate system from three points:

i) The first point specifies the origin.

ii) The first and second points together specify the x-axis.

iii) The three points together specify the x-y plane of the new coordinate system. The y and z axis are derived from orthogonality and right-handedness.

c) The *Geopts* option generates the new coordinate system from a global axis and two points. With this method the new system's z-axis is set from the *Direction* drop-down. This new system's x-y plane is, then, orthogonal to the chosen direction. The remaining two points serve to define the origin and the x-axis (by projecting the second point). This option is useful for metal forming application, since, often times, only the z-axis is important while the while the x and y axes are not.

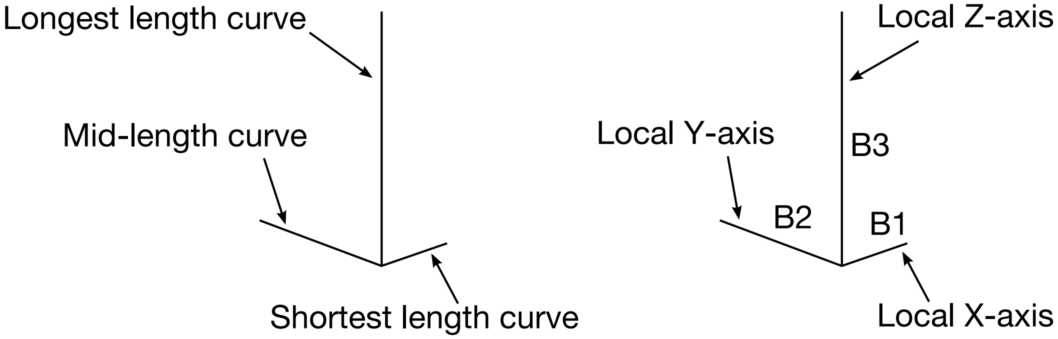


Figure 15-6. Input curves (left). The generated local coordinate system is written to the d3plot file as a part consisting of three beams (right).

4. **IGES.** When option, IGES, is used, three curves in the IGES format will be used to define a local coordinate system. IGES curve entity types 126, 110 and 106 are currently supported. Among the three curves, the longest length will be made as local Z-axis, the mid-length will be Y-axis and the shortest length X-axis. Suggested X, Y and Z-axis length is 100mm, 200mm and 300mm, respectively.

   All the three curves must have one identical point, and will be used for the origin of the new local coordinate system. The coordinate system ID for the local system will be based on the IGES file name. The IGES file name must start with a number, followed by an underscore "_", or by a dot. The number preceding the file name will be used as the new local coordinate system ID, which can then be referenced in *MAT_20 cards, for example.

   After the LS-DYNA run, three beam elements of a new PID will be created in place of the three curves representing the local X, Y, and Z-axis in the d3plot file for viewing in LS-PrePost. See Figure 15-6.

   The following partial input contains an example in which the keyword is used to create a local coordinate system (CID = 25) from IGES input. The IGES file named, 25_iges, contains three intersecting curves in one of the three supported IGES entity types. The example demonstrates using the IGES coordinate system (ID = 25) to specify the local coordinate system for a rigid body (PID = 2, MID = 2). The keyword, *BOUNDARY_PRESCRIBED_MOTION_RIGID_ LOCAL, then uses this local coordinate system to assign velocities from load curves 3 and 5 for the rigid body motion in the local x-direction.

```
*KEYWORD
*DEFINE_COORDINATE_SYSTEM_IGES_TITLE
Flanging OP25
25_iges
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+---
-8
*PART
punch
         2         2         2
```

```
*MAT_RIGID
$      MID        RO         E        PR         N    COUPLE         M
ALIAS
         2 7.830E-09 2.070E+05      0.28
$      CMO      CON1      CON2
        -1        25    011111
$LCO or A1        A2        A3        V1        V2        V3
25
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+---
-8
*BOUNDARY_PRESCRIBED_MOTION_RIGID_LOCAL
$   typeID       DOF       VAD      LCID        SF       VID     DEATH
BIRTH
         2         1         0         3      -1.0         0   0.00241
0.0
         2         1         0         5      -1.0         0 0.0115243
0.00241
```

The keyword can be repeated for each new coordinate system if multiple coordinate systems are needed.

Revision information:

This option is available starting in LS-DYNA Revision 62798.

*DEFINE_COORDINATE_VECTOR

Purpose: Define a local coordinate system with two vectors, see Figure 15-7. The vector cross product, $z = x \times xy$, determines the z-axis. The y-axis is then given by $y = z \times x$. If this coordinate system is assigned to a nodal point, then at each time step during the calculation, the coordinate system is incrementally rotated using the angular velocity of the nodal point to which it is assigned.

Card	1	2	3	4	5	6	7	8
Variable	CID	XX	YX	ZX	XV	YV	ZV	NID
Type	I	F	F	F	F	F	F	I
Default	0	0.0	0.0	0.0	0.0	0.0	0.0	0.

VARIABLE	DESCRIPTION
CID	Coordinate system ID. A unique number has to be defined.
XX	X-coordinate on local x-axis. Origin lies at (0,0,0).
YX	Y-coordinate on local x-axis
ZX	Z-coordinate on local x-axis
XV	X-coordinate of local x-y vector
YV	Y-coordinate of local x-y vector
ZV	Z-coordinate of local x-y vector
NID	Optional nodal point ID. The coordinate system rotates with the rotation of this node. If the node is not defined, the coordinate system is stationary.

Remarks:

1. These vectors should be separated by a reasonable included angle to avoid numerical inaccuracies.

2. Ideally, this nodal point should be attached to a rigid body or a structural part where the nodal point angular velocities are meaningful. It should be noted that angular velocities of nodes may not be meaningful if the nodal point is

attached only to solid elements and even to shell elements where the drilling degree of freedom may be singular, which is likely in flat geometries.

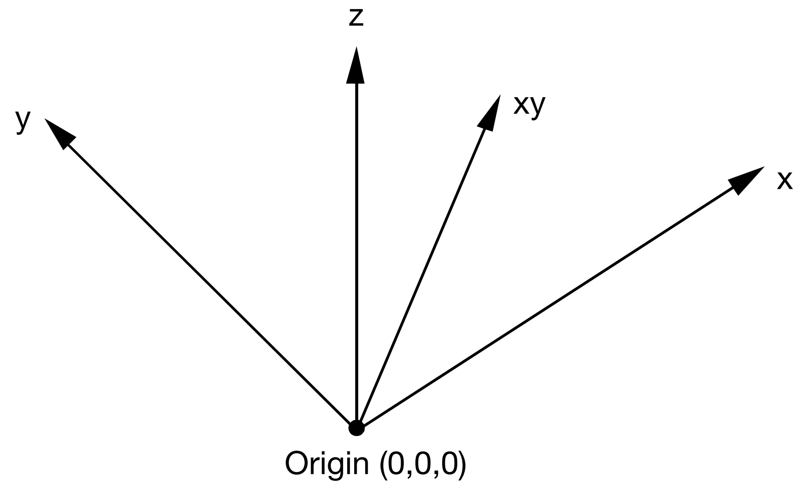


Figure 15-7. Definition of the coordinate system with two vectors.

*DEFINE_CPM_BAG_INTERACTION

Purpose: To model energy flow from a master airbag to a slave airbag. The master must be an active particle airbag and the slave a control volume (CV) airbag converted from a particle bag.

To track the flow of energy, LS-DYNA automatically determines which vent parts are common to both airbags. At each time step the energy that is vented through the common vents is subtracted from the master and added to the slave. In turn, the slave bag's pressure provides the downstream pressure value for the master bag's venting equation. While this model accounts for energy flow from master to slave it ignores flow from slave to master.

If CHAMBER is used for slave CV bag, see remark 1.

Card 1	1	2	3	4	5	6	7	8
Variable	Bag ID1	Bag ID2						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
Bag ID1	Airbag ID of master CPM particle bag
Bag ID2	Airbag ID of slave CV bag switched from CPM bag

Remarks:

1. Due to the complexity of the bookkeeping, the slave may have several chambers but only one of the chambers is allowed to interact with the master bag. This chamber will be searched automatically through the commonly shared parts.

*DEFINE_CPM_CHAMBER

Purpose: To define airbag chambers for air particle initialization or chamber interaction.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	NCHM						
Type	I	I						
Default	none	0						

Chamber Definition Card Sets:

Add NCHM chamber definition card sets. Each chamber definition card set consists of a Chamber Definition Card followed by NINTER Interaction Cards.

Chamber Definition Card.

Card 2	1	2	3	4	5	6	7	8
Variable	SID1	SID2	NINTER	CHM_ID				
Type	I	I	I	I				
Default	none	0	0	0				

Interaction Cards. Add NINTER of these. If NINTER = 0, skip this card.

Card 3	1	2	3	4	5	6	7	8
Variable	SID3	ITYPE3	TOCHM					
Type	I	I	I					
Default	none	none	none					

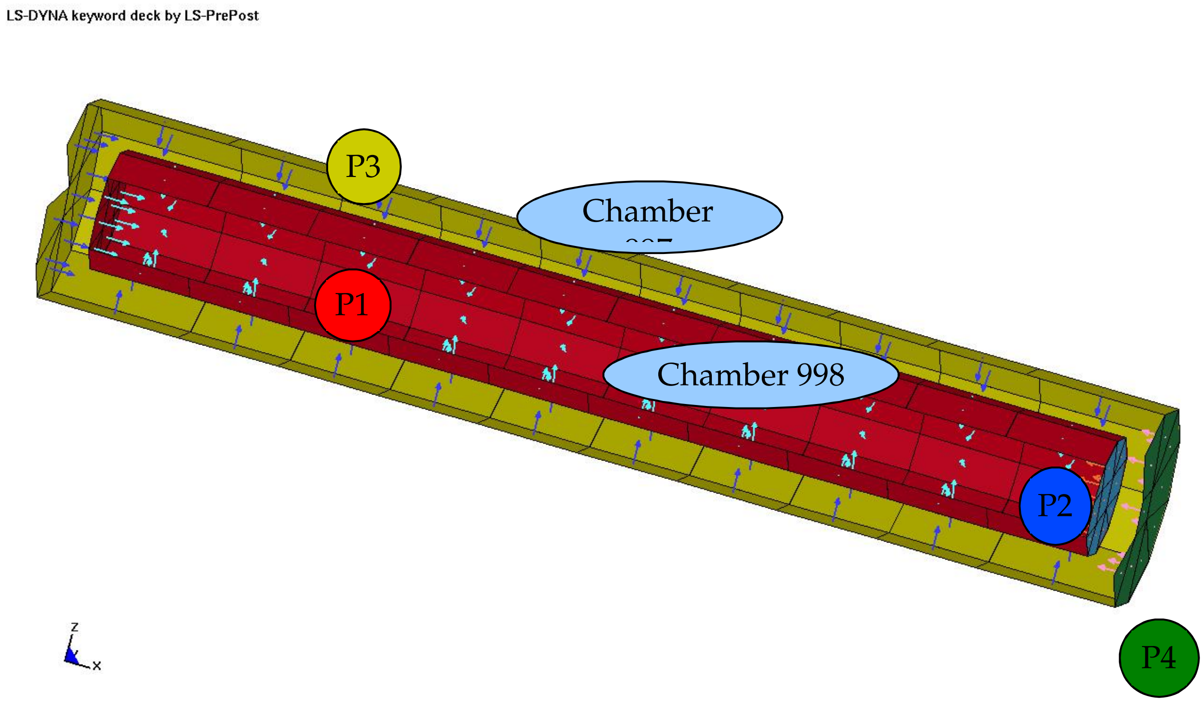


Figure 15-8.

VARIABLE	DESCRIPTION
ID	Unique ID for this card
NCHM	Number of chambers defined in this card
SID1	Part set defining all parts that constitute the chamber volume (See Remark 1)
SID2	Part set defining the parts whose shell normals need to be flipped (eg. separation walls between chambers) (See Remark 1)
NINTER	Number of vent hole definition for chamber interaction.
CHM_ID	Chamber ID (see Remark 2).
SID3	Set defining interaction between chambers
ITYPE3	Set type EQ.0: Part EQ.1: Part set
TOCHM	The chamber ID of the connected chamber.

Remarks:

1. Each chamber's volume is calculated based on the part normals pointed inwards. So SID1 would normally have parts with their shell normals pointing inwards. But in some cases, parts may be shared by more than one chamber. In this case, the shell orientation of certain part(s) may need to be flipped for the other chambers in question. In such cases, SID2 can be used to flip the shell-normals for specific parts.

```
*SET_PART_LIST
$#     sid
         1
$#    pid1      pid2      pid3      pid4
         1         2         3         4
*SET_PART_LIST
$#     sid
        20
$#    pid1      pid2
         1         2
*DEFINE_CPM_CHAMBER
$#      id      nchm
      1234         2
$#    sid1      sid2    ninter    chm_id
        20         0         1       998
$#    sid3    itype3     tochm
         2         0       997
$#    sid1      sid2    ninter    chm_id
         1        20         1       997
$#    sid3    itype3     tochm
         2         0       998
```

2. Particles with different chamber ID will not interact in particle to particle collision. This feature will allow program to distinguish particles separated by a thin wall.

3. All chambers data are output to lsda binout database. The utility “l2a” can convert it into abstat_chamber ASCII file and process with lsprepost under abstat format

*DEFINE_CPM_GAS_PROPERTIES

Purpose: To define extended gas thermodynamic properties

Card 1	1	2	3	4	5	6	7	8
Variable	ID	Xmm	Cp0	Cp1	Cp2	Cp3	Cp4	
Type	I	F	F	F	F	F	F	
Default	none	none	0.	0.	0.	0.	0.	

Card 1	1	2	3	4	5	6	7	8
Variable	µt0	µt1	µt2	µt3	µt4	Chm_ID	Vini	
Type	F	F	F	F	F	I	F	
Default	0.	0.	0.	0.	0.	0	0.	

VARIABLE	DESCRIPTION
ID	Unique ID for this card
Xmm	Molar mass
Cp0, …, Cp4	Coefficients of temperature dependent specific heat with constant pressure $C_p(T) = C_{p0} + C_{p1}\,T + C_{p2}\,T^2 + C_{p3}\,T^3 + C_{p4}\,T^4$
µt0, …, µt4	Coefficients of temperature dependent Joule-Thomson effect $\mu_t(T) = \mu_{t0} + \mu_{t1}\,T + \mu_{t2}\,T^2 + \mu_{t2}\,T^3 + \mu_{t2}\,T^4$
Chm_ID	Chamber ID (remark 1)
Vini	Initial volume for user defined inflator (remark 1)

Example:

```
*AIRBAG_PARTICLE
$====1====$====2====$====3====$====4====$====5====$====6====$====7====$====8====
      1010         1      1011         1         0       0.0       0.0         1
    100000         0         1     300.0   1.0e-04         1
         1         1         1
        61         0       1.0         0         0         1       0.0
   1.0E-04     300.0     -9900
       651       653     -9910
   3000001       1.0
$=====================================================
*DEFINE_CPM_GAS_PROPERTIES
$====1====$====2====$====3====$====4====$====5====$====6====$====7====$====8====
      9900 2.897E-02 2.671E+01 7.466E-03-1.323E-06
      9910   4.0E-03     20.79
   -610.63   -0.0926
```

Remark:

1.If Chm_ID and Vini are defined. This gas property will be used in the user_ inflator routine which is provided in the dyn21b.f of general usermat package. The code will give current chamber volume, pressure, temperature and time step and expect returning value of change of chamber, burned gas temperature and mass flow rate to feedback to the code for releasing particles. All state data for this chamber will be output binout under abstat_chamber subdirectory.

*DEFINE_CPM_VENT

Purpose: To define extended vent hole options

Card 1	1	2	3	4	5	6	7	8
Variable	ID	C23	LCTC23	LCPC23	ENH_V	PPOP	C23UP	IOPT
Type	I	F	I	I	I	F	F	
Default	none	none	none	none	none	none	none	

Card 2	1	2	3	4	5	6	7	8
Variable	JT	IDS1	IDS2	IOPT1	PID1	IPD2	VANG	
Type	I	I	I	I	I	I	F	
Default	0	none	none	none	none	none	0.	

VARIABLE	DESCRIPTION
ID	Unique ID for this card
C23	Vent hole coefficient. This is the Wang-Nefske leakage parameter. (Default 1.0)
LCTC23	Load curve defining vent hole coefficient as a function of time.
LCPC23	Load curve defining vent hole coefficient as a function of pressure.
ENH_V	Enhance venting option. (Default 0). However if Joule-Thomson effect is considered, the option will set to 1 automatically. EQ.0: disable EQ.1: enable
PPOP	Pressure difference between interior and ambient pressure to open the vent hole. Once the vent is open then it will stay open.

VARIABLE DESCRIPTION

C23UP

Scale factor of C23 while switching from CPM to uniform pressure calculation.

IOPT

Directional venting:

EQ.1: In shell normal

EQ.2: Against shell normal

One-way venting:

EQ.10: In shell normal

EQ.20: Against shell normal

Special vent option::

EQ.100: Enable compression seal vent. Vent area is adjusted according to the formula below. See Remark 1.

$$A_{\text{vent}} = \max(A_{\text{current}} - A_0, 0)$$

EQ.200: Enable push-out vent. Particle remains active while going through this external vent within the range of 2 times of its characteristic length, l_{vent}.

$$l_{\text{vent}} = \sqrt{A_{\text{vent}}}$$

JT

Include the Joule-Thomson effect. When the Joule-Thomson effect is enabled ENH_V is automatically set to 1 (enable).

EQ.0: disable

EQ.1: use part pressure

EQ.2: use chamber pressure

IDS1

JT's up stream condition part ID/chamber ID

IDS2

JT's downstream condition part ID/chamber ID

IOPT1

Upstream chamber ID for one-way vent hole. This will help the code to determine the probability function.

PID1, PID2

When specified the vent probability function is evaluated from the difference of local part pressures (between PID1 and PID2) instead of the usual calculation involving the chamber pressure. This option is usually used for vents near a long sleeve which causing unrealistic venting using chamber pressure alone.

VARIABLE **DESCRIPTION**

VANG

Cone angle in degrees. Particle goes through this vent will be redirection based on this angle. This option is only valid with internal vent.

GT.0: cone angle (maximum 270)

EQ.0: disabled (Default)

LT.0: direction follows the vent normal

Remarks:

1. **Compression Seal Vent Model.** In order to evaluate bag state variables correctly, the CPM domain needs to be a closed surface for the volume to be well-defined. If the model contains a flap vent which is free to open and close, this option will correctly maintain the bag's integrity.

Example:

```
*AIRBAG_PARTICLE
$====1====$====2====$====3====$====4====$====5====$====6====$====7====$====8====
      1010         1      1011         1         0       0.0       0.0         1
    100000         0         1     300.0   1.0e-04         1
         1         1         1
        61         0     -9910
   1.0E-04     300.0  2.897E-2  2.671E+1  7.466E-3 -1.323E-6
      1000      1001    4.0E-3     20.79
   3000001       1.0
$====================================================
*DEFINE_CPM_VENT
$====1====$====2====$====3====$====4====$====5====$====6====$====7====$====8====
      9910       1.0         0         0         1       0.0
         1        51         2
```

DEFINE_CURVE_*{OPTION}

Purpose: Define a curve [for example, load (ordinate value) versus time (abscissa value)], often loosely referred to as a load curve. The ordinate may represent something other than a load however, as in the case of curves for constitutive models.

In the case of constitutive models, *DEFINE_CURVE curves are rediscretized internally with equal intervals along the abscissa for fast evaluation. Rediscretization is *not* used when evaluating loading conditions such as pressures, concentrated forces, or displacement boundary conditions (see Remark 1 for more details).

The curve rediscretization algorithm was enhanced for the 2005 release of version 970. In certain cases the new load-curve routines changed the final results enough to disrupt benchmarks. For validated models, such as barriers and occupants, requiring numerical consistency, there are keyword options for reverting to the older algorithms.

Available options include:

<OPTION>

3858

5434a

which correspond to the first releases of version 970 and the 2005 release, respectively.

Since input errors and wrong results are sometimes related to load curve usage, a "*Load curve usage*" table is printed in the d3hsp file after all the input is read. This table should be checked to ensure that each curve ID is referenced by the option for which the curve is intended.

Card 1	1	2	3	4	5	6	7	8
Variable	LCID	SIDR	SFA	SFO	OFFA	OFFO	DATTYP	LCINT
Type	I	I	F	F	F	F	I	I
Default	none	0	1.	1.	0.	0.	0	0

Point Cards. Put one pair of points per card (2E20.0). Input is terminated at the next keyword ("*") card.

Card 2…	1	2	3	4	5	6	7	8
Variable	A1		01					
Type	E20.0		E20.0					
Default	0.0		0.0					

VARIABLE	DESCRIPTION

LCID
Load curve ID. Tables (see *DEFINE_TABLE) and load curves may not share common ID's. LS-DYNA allows load curve ID's and table ID's to be used interchangeably. A unique number has to be defined.

SIDR
Flag controlling use of curve during dynamic relaxation. SIDR set to 1 or 2 will activate a dynamic relaxation phase unless IDRFLG = -999 in *CONTROL_DYNAMIC_RELAXATION.

EQ.0: load curve used in normal analysis phase only or for other applications,

EQ.1: load curve used in dynamic relaxation phase but not the normal analysis phase,

EQ.2: load curve applies to both dynamic relaxation phase and normal analysis phase.

SFA
Scale factor for abscissa value. This is useful for simple modifications.

EQ.0.0: default set to 1.0.

SFO
Scale factor for ordinate value (function). This is useful for simple modifications.

EQ.0.0: default set to 1.0.

OFFA
Offset for abscissa values, see explanation below.

OFFO
Offset for ordinate values (function), see explanation below.

VARIABLE **DESCRIPTION**

DATTYP Data type. This affects how offsets are applied (see description below).

EQ.-2: for fabric stress vs. strain curves (*MAT_FABRIC) as described below.

EQ.0: general case for time dependent curves, force versus displacement curves and stress strain curves

EQ.1: for general (x, y) data curves whose abscissa values do not increase monotonically

EQ.6: for general (r, s) data (coordinates in a 2D parametric space) whose values do not increase monotonically. Use for definition of trimming polygons for trimmed NURBS (*ELEMENT_SHELL_NURBS_PATCH, NL.GT.0)

LCINT The number of discretization points to use for this curve. If 0 is input, the value of LCINT from *CONTROL_SOLUTION will be used.

A1, A2, … Abscissa values. See remarks below.

O1, O2, … Ordinate (function) values. See remarks below.

Remarks:

1. ***Warning* Concerning Rediscretization.** For constitutive models, LS-DYNA internally rediscretizes the curve with uniform spacing to bypass searching during evaluations. The major drawback of this algorithm is that any detail in the curve on a scale finer than the uniform rediscretization grid will be smoothed-out and lost. *It is, therefore, important to avoid placing a single point off at some value approaching infinity.* The lone point at infinity will cause the resolution of the uniform grid to be coarse relative to the other points, causing the rediscretized curve to be, possibly, featureless.

   Therefore, when defining curves for constitutive models, points should be spaced as uniformly as possible. Also, since the constitutive model curves are extrapolated, it is important to ensure that extrapolation does not lead to physically meaningless values, such as a negative flow stress. Conversely, extrapolation can be exploited to control the results of evaluations at points far from the input data.

The number of points in each rediscretized curve is controlled by the parameter LCINT in *CONTROL_SOLUTION. By changing LCINT to a value greater than the default of 100, the rediscretized curves may better resemble the input curves. The data points of the rediscretized curves are written to messag and d3hsp if the parameter IPCURV is set to 1 in *CONTROL_OUTPUT.

2. **Scaling.** The load curve values are scaled after the offsets are applied, i.e.:

$$\text{Abscissa value} = \text{SFA} \times (\text{Defined value} + \text{OFFA})$$

$$\text{Ordinate value} = \text{SFO} \times (\text{Defined value} + \text{OFFO})$$

3. **DATTYP.** The DATTYP field controls how the curve is processed during the calculation.

   a) For DATTYP = 0 positive offsets may be used when the abscissa represents time, since two additional points are generated automatically at time zero and at time 0.999 × OFFA with the function values set to zero.

   b) If DATTYP = 1, then the offsets do not create these additional points. Negative offsets for the abscissa simply shifts the abscissa values without creating additional points.

   c) For *MAT_FABRIC material with FORM = 4, 14, -14, or 24, set DATYP = -2 to define stress vs. strain curves using engineering stress and strain rather the 2nd Piola-Kirchhoff stress and Green strain.

4. **Context Dependent Extrapolation.** Load curves are not extrapolated by LS-DYNA for applied loads such as pressures, concentrated forces, displacement boundary conditions, etc. Function values are set to zero if the time, etc., goes off scale. Therefore, extreme care must be observed when defining load curves. In the constitutive models, extrapolation is employed if the values on the abscissa go off scale.

5. **Restart.** The curve offsets and scale factors are ignored during restarts if the curve is redefined. See *CHANGE_CURVE_DEFINITION in the restart section.

*DEFINE_CURVE_BOX_ADAPTIVITY

Purpose: To define a polygon adaptive box in sheet metal forming, applicable to shell elements. This keyword is used together with *CONTROL_ADAPTIVE. Other related keyword is *DEFINE_BOX_ADAPTIVE.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	PID	LEVEL	DIST1				
Type	I	I	I	F				
Default	none	none	none	none				

Point Cards. Include as many as necessary. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	X		Y		Z			
Type	F		F		F			
Default	none		none		none			

VARIABLE	DESCRIPTION
ID	Curve ID; must be unique. The curve must be closed: its first and last point *must* coincide.
PID	Sheet blank Part ID, as in *PART.
LEVEL	Adaptive refinement levels, similar to 'MAXLVL' in *CONTROL_ADAPTIVE.

VARIABLE	DESCRIPTION
DIST1	Extended depths in Z for a polygon box defined by Card 2, 3, 4, etc. Currently this variable must be input as a negative value. The box depth in Z will be extended in –Z direction by Z_{min}-abs(DIST1) and in +Z direction by Z_{max}+abs(DIST1). The XYZ data pairs formed with Card 2, 3, 4, etc. will be automatically closed to create the polygon box. Zmin and Zmax are the minimum and maximum Z-coordinates in all the data pairs.
X	X-coordinate of a point on the curve.
Y	Y-coordinate of a point on the curve.
Z	Z-coordinate of a point on the curve.

Remarks:

Within the polygon, the variable LEVEL has priority over MAXLVL in *CONTROL_-ADAPTIVE but limited to minimum element size controlled by ADPSIZE. A larger LEVEL (than MAXLVL) value will enable more mesh refinement within the polygon, up to the size defined by ADPSIZE, while meshes outside of the box refined less by a smaller MAXLVL value. However, mesh refinement when LEVEL > MAXLVL is not recommended. The appropriate way of using this keyword (and *DEFINE_BOX_-ADAPTIVE) is to define the polygon box excluding the local areas of interest so refinement inside the local areas will be controlled by MAXLVL while outside of the area to be controlled by LEVEL, and in this case MAXLVL > LEVEL, as shown in Figure 15-9. The advantage of using this keyword is obvious when compared with multiple boxes needed when defining local adaptive refinement with keyword *DEFINE_BOX_-ADPATIVE (Figure 15-10). It is noted that ADPSIZE is a “global” variable, meaning final refined element sized, regardless of the values set for MAXLVL or LEVEL, cannot be smaller than what is defined by ADPSIZE.

The 3-D curve (closed polygon) defined by XYZ data pairs should be near the sheet blank in Z after the blank is auto-positioned in the beginning of a simulation. Similar to *DEFINE_BOX_ADAPTIVE, only the elements on the sheet blank initially within the polygon will be considered for use with this keyword. Local coordinate system is not supported at the moment.

The 3-D curve can be converted from IGES format to format required here following the procedure outlined in keyword *INTERFACE_BLANKSIZE_{OPTION}.

A partial keyword example is provided below, where inside the polygon mesh has no refinement (LEVEL = 1), while outside of the box the mesh is refined 5 levels

(MAXLVL = 5). The final minimum element size is defined as 0.4. It is noted that the first point and last point of the polygon are the same, closing the polygon box.

```
*CONTROL_ADAPTIVE
$  ADPFREQ    ADPTOL    ADPOPT    MAXLVL    TBIRTH    TDEATH     LCADP    IOFLAG
   &adpfq1       5.0         2         5       0.0 1.000E+20                   1
$  ADPSIZE    ADPASS    IREFLG    ADPENE     ADPTH    MEMORY    ORIENT     MAXEL
       0.4         1         0   &lookfd       0.0                   0         0
$  IADPE90              NCFREQ    IADPCL    ADPCTL    CBIRTH    CDEATH     LCLVL
        -1
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*DEFINE_CURVE_BOX_ADAPTIVITY
$       ID       PID     LEVEL     DIST1
        99         1         1     -25.0
$(3E20.0):
          -59.573399           -6.698870          -40.224651
          -90.728516           24.456253          -40.224651
...
          -23.213169           18.088070          -10.954337
           14.353654           16.130911          -10.954337
          -31.070744           -5.785467          -40.487387
          -59.573399           -6.698870          -40.224651
```

Revision information:

This feature is available in SMP only, and in LS-DYNA Revision 81918 and later releases.

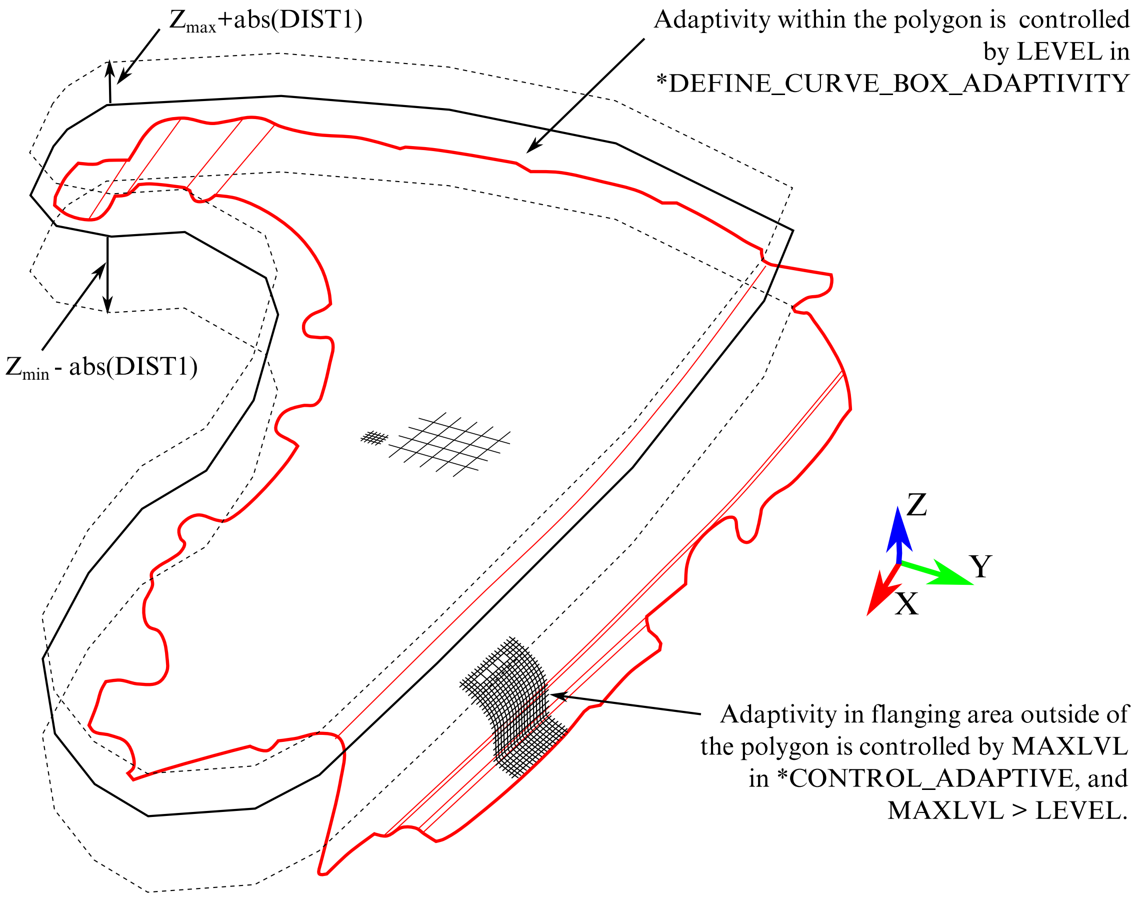
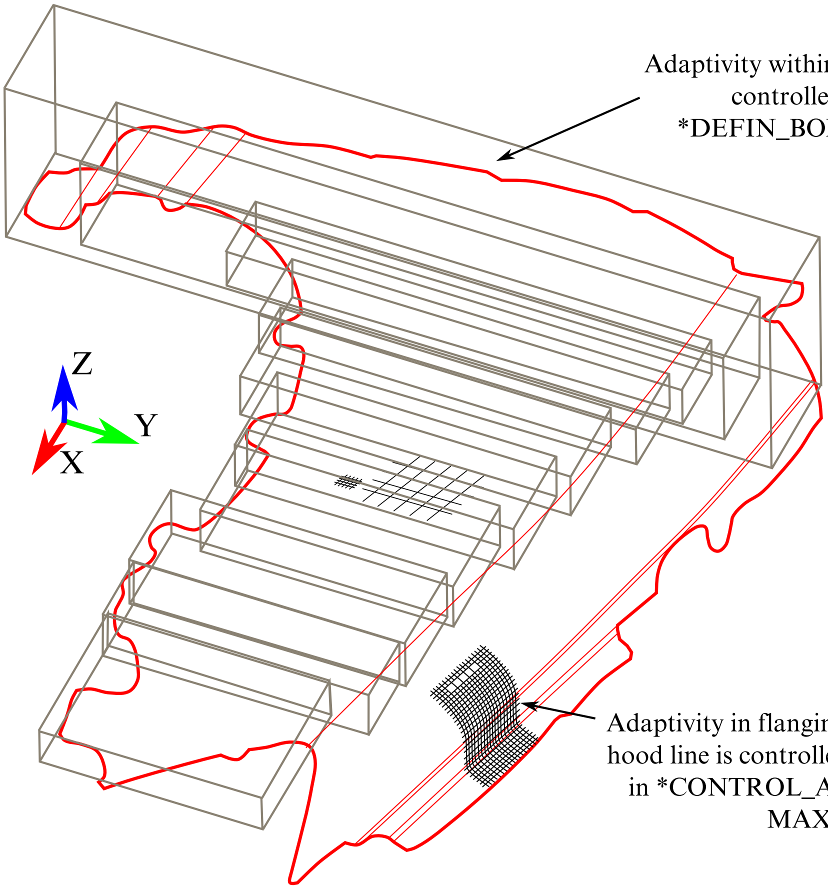

Z_{max}+abs(DIST1)
Adaptivity within the polygon is controlled by LEVEL in *DEFINE_CURVE_BOX_ADAPTIVITY
Z_{min} - abs(DIST1)
Z
Y
X
Adaptivity in flanging area outside of the polygon is controlled by MAXLVL in *CONTROL_ADAPTIVE, and MAXLVL > LEVEL.

Figure 15-9. Defining an adaptive polygon box



Adaptivity within these boxes is controlled by LEVEL in *DEFIN_BOX_ADAPTIVE

Adaptivity in flanging area along the hood line is controlled by MAXLVL in *CONTROL_ADAPTIVE, and MAXLVL > LEVEL.

Figure 15-10. Defining adaptive boxes

DEFINE_CURVE_COMPENSATION_CONSTRAINT_*OPTION

Purpose: This keyword with the two options allows for the definition of a localized die face region for springback compensation of stamping tools.

Options available include:

BEGIN

END

NOTE: *DEFINE_CURVE_COMPENSATION_CONSTRAINT_BEGIN and *DEFINE_CURVE_COMPENSATION_CONSTRAINT_END are not valid in the context of a general keyword input deck. Instead, they may only be used inside of an *INCLUDE_COMPENSATION_CURVE include file.

The required option, which must be either BEGIN or END, distinguishes between two different closed curves, which, when taken together identify a portion of the die wherein springback compensation is applied, and a transition region for which compensation smoothly tapers off.

Card 1	1	2	3	4	5	6	7	8
Variable	CRVID	INOUT	TYPE					
Type	I	I	I					
Default	0	0	none					

Point Cards. Include as many as necessary (3E16.0). This input ends at the next keyword ("*") card. Only the projection of this curve onto the *x*-*y* plane is used.

Card 2	1	2	3	4	5	6	7	8
Variable	X		Y		Z			
Type	F		F		F			
Default	0.0		0.0		0.0			

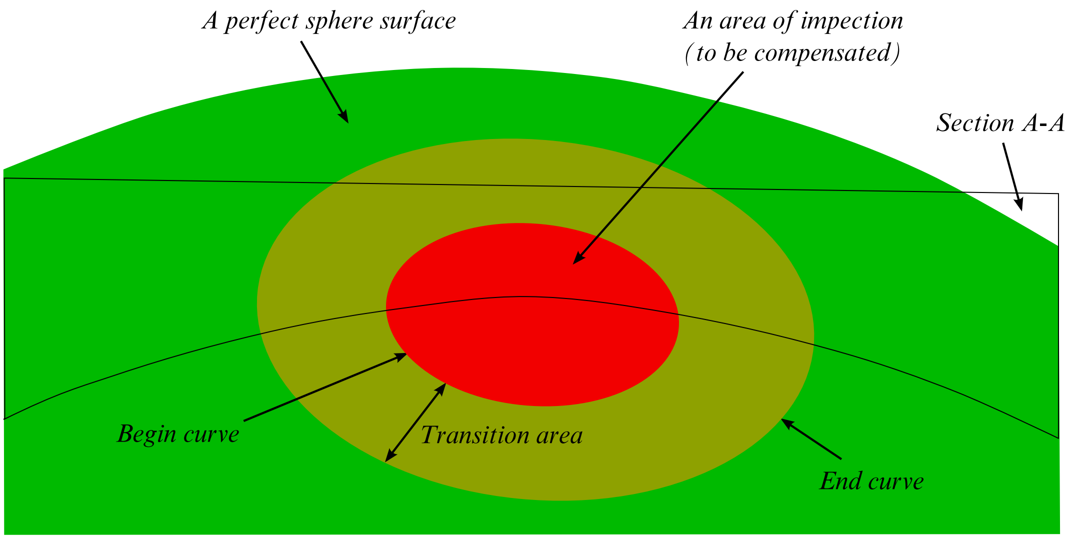


Figure 15-11. Local area compensation.

VARIABLE	DESCRIPTION
CRVID	Curve ID; must be unique. The curve must be closed: its first and last point *must* coincide.
INOUT	Flag to indicate local area to be compensated: EQ.0: For this option, the compensated region of the die consists of all points for which the projection onto the *x*-*y* plane is exterior to the projection of the BEGIN curve. The projection of the END curve is assumed exterior to the BEGIN curve. The transition region, then, consists of all die points for which the projection is between the BEGIN and END curves. All other points on the die are uncompensated. EQ.1: For this option, the compensated region of the die consists of all points for which the projection onto the *x*-*y* plane is interior to the projection of the BEGIN curve. The projection of the END curve is assumed exterior to the BEGIN curve. The transition region, then, consists of all die points for which the projection is between the BEGIN and END curves. All other points on the die are uncompensated. See Figure 15-11.
TYPE	Type code - must be “0”.
X	*x*-coordinate of a point on the curve.
Y	*y*-coordinate of a point on the curve.

VARIABLE	DESCRIPTION
Z	*z*-coordinate of a point on the curve.

Motivation:

Sometimes springback occurs in a localized region of the die face. Since other parts of the die face are better left undisturbed, a localized compensation makes the most sense to bring the part shape back to the design intent. A typical such example will be the front portion along the grill and headlamp, or the rear portion along the windshield of a trimmed hood inner panel. A decklid (or trunk lid) inner also exhibits the similar needs. Once the localized areas are identified, iterative compensation scheme may be employed within this localized region to bring the springback panel back to design shape.

Modeling details:

Referring to Figure 15-11, the keywords *COMPENSATION_CONSTRAINT_BEGIN and *COMPENSATION_CONSTRAINT_END must be used together in a file, which in turn will be included in keyword *INCLUDE_COMPENSATION_CURVE. The keyword "BEGIN" precedes the keyword "END", each is defined by discrete points. In addition, each curve must form a closed loop. The area formed between the two curves is a transition area, and will be affected in the compensated tooling. LS-PrePost4.0 under Curve → Merge → Multiple Method, multiple disconnected curves can be joined together, and output in ".xyz" format required here.

The curve can be a 3-D piecewise linear curve with coordinates in *x*, *y* and *z*. However, *z*-coordinates are ignored; meaning the tooling to be compensated must be positioned so draw direction is in global *z*; otherwise error will occur. In addition, it is assumed that both "blank before springback" and "blank after springback" will be smaller than rigid tools in dimension. It is further noted the rigid tool meshes should be discretized fine enough to provide enough degrees of freedom for the compensation.

Application example – single region:

A complete input deck is provided below for a local compensation simulation. The keyword files state1.k and state2.k consist model (nodes and elements) information of the blank before and after springback, respectively. It is noted here that if the blank is adaptively refined, the adaptive constraints must be included in the keyword files. The keyword file tools.k consists the stamping tools (with PID 1, 2, 3 and 4) all positioned in home position. The keyword file curvesxy.xyz consists keywords "BEGIN" and "END" defining the two closed-loop curves used to define a localized area.

```
*KEYWORD
```

```
*TITLE
LS-Dyna971 Compensation Job
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*INTERFACE_COMPENSATION_NEW
$   METHOD        SL        SF     ELREF     PSIDm     UNDCT     ANGLE  NOLINEAR
         6    10.000     0.700         0         1         0         0         1
*INCLUDE_COMPENSATION_BLANK_BEFORE_SPRINGBACK
state1.k
*INCLUDE_COMPENSATION_BLANK_AFTER_SPRINGBACK
state2.k
*INCLUDE_COMPENSATION_DESIRED_BLANK_SHAPE
state1.k
*INCLUDE_COMPENSATION_COMPENSATED_SHAPE
state1.k
*INCLUDE_COMPENSATION_CURRENT_TOOLS
tools.k
*INCLUDE_COMPENSATION_CURVE
curvesxy.xyz
*SET_PART_LIST
         1
1,2,3,4
*END
```

A portion of the file *curvesxy.xyz* is shown below,

```
*KEYWORD
*DEFINE_CURVE_COMPENSATION_CONSTRAINT_BEGIN
$      CID    IN/OUT      TYPE
         1         1         0
    -1.86925e+02     1.83338e+03    -1.55520e+01
    -1.83545e+02     1.83003e+03    -1.55469e+01
    -1.80162e+02     1.82668e+03    -1.55428e+01
    -1.91811e+02     1.83884e+03    -1.56014e+01
    -1.90187e+02     1.83701e+03    -1.55852e+01
    -1.88560e+02     1.83519e+03    -1.55688e+01
    -1.86925e+02     1.83338e+03    -1.55520e+01
*DEFINE_CURVE_COMPENSATION_CONSTRAINT_END
$      CID    IN/OUT      TYPE
         2         1         0
    -4.07730e+02     1.61371e+03    -8.04858e+01
    -3.84480e+02     1.59890e+03    -7.99169e+01
    -3.61193e+02     1.58423e+03    -7.93471e+01
    -3.37832e+02     1.56984e+03    -7.87756e+01
    -4.49289e+02     1.67556e+03    -8.04582e+01
    -4.35672e+02     1.65473e+03    -8.05162e+01
    -4.21764e+02     1.63396e+03    -8.05530e+01
    -4.07730e+02     1.61371e+03    -8.04858e+01
*END
```

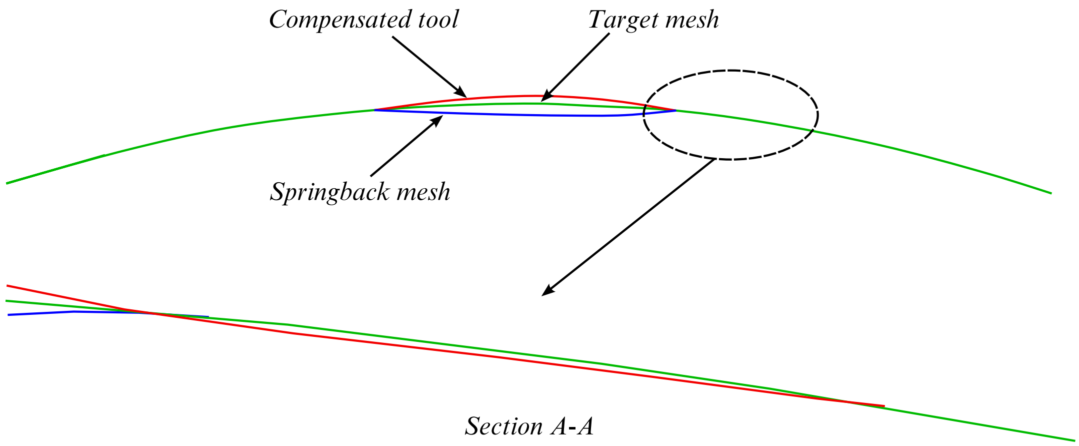


Figure 15-12. Local compensation details.

It is noted the first point and last point are exactly the same, forming a closed loop. In Figure 15-11, local area compensation is to be performed in the center portion of a rigid sphere. Based on springback and target meshes, the compensated tool mesh is obtained and smooth transition areas are achieved, Figure 15-12. Here the compensation scale factor of 0.7 is used.

Application example – multiple regions:

Multi-region localized compensation is also possible by defining multiple pairs of the BEGIN and END versions of this keyword, each forming a localized region. For example, for localized compensation of two regions, the file curvesxy.xyz will read as follows,

```
*KEYWORD
*DEFINE_CURVE_COMPENSATION_CONSTRAINT_BEGIN
$       CID     IN/OUT       TYPE
          1          1          0
     3.67967e+02     1.63423e+03    -6.98532e+01
     3.60669e+02     1.62992e+03    -6.92921e+01
     3.53586e+02     1.62525e+03    -6.88777e+01
          ⋮                ⋮                ⋮
*DEFINE_CURVE_COMPENSATION_CONSTRAINT_END
$       CID     IN/OUT       TYPE
          2          1          0
     4.12534e+02     1.75537e+03    -5.83975e+01
     3.98853e+02     1.75264e+03    -5.58860e+01
     3.85292e+02     1.74921e+03    -5.35915e+01
          ⋮                ⋮                ⋮
*DEFINE_CURVE_COMPENSATION_CONSTRAINT_BEGIN
$       CID     IN/OUT       TYPE
          3          1          0
    -4.37478e+02     2.67393e+03    -1.70421e+02
    -4.45605e+02     2.67209e+03    -1.71724e+02
    -4.53649e+02     2.66985e+03    -1.72894e+02
          ⋮                ⋮                ⋮
*DEFINE_CURVE_COMPENSATION_CONSTRAINT_END
$       CID     IN/OUT       TYPE
          4          1          0
    -4.49426e+02     2.79057e+03    -2.18740e+02
    -4.63394e+02     2.78749e+03    -2.20955e+02
```

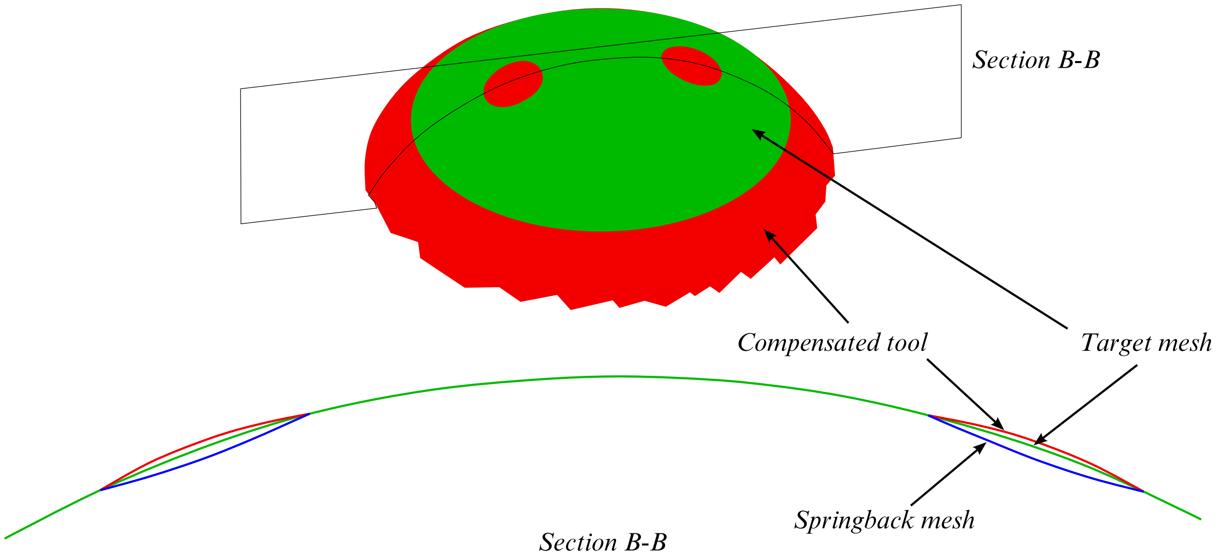


Figure 15-13. Multi-region local compensation.

```
    -4.77223e+02      2.78370e+03    -2.22938e+02
         ⋮                 ⋮                ⋮
*END
```

Figure 15-13 (top) shows an example of two localized areas of the sphere to be compensated. The compensation results are shown in Figure 15-13 (bottom). Again, a compensation scale factor of 0.7 was used and smooth transition areas are achieved.

Revision information:

This feature is available in double precision version of LS-DYNA starting in Revision 62038. Multi-region localized compensation is available starting in Revision 66129 and later releases. In addition, prior to Revision 66129, all keywords must be capitalized. Also, official release version starting in R7.1.1 (double precision) can be used.

*DEFINE_CURVE_DRAWBEAD

Purpose: This keyword simplifies the definition of a draw bead, which previously required the use of many keywords.

> **NOTE:** This option has been deprecated in favor of *DEFINE_MULTI_DRAWBEADS_IGES.

Card 1	1	2	3	4	5	6	7	8
Variable	CID	TCTYPE	VID	PID	BLKID	PERCT	LCID	
Type	I	I	I	I	I	F	I	
Default	none	none	none	none	none	0.0	none	

Point Cards. For TCTYPE = 1 define points on the curve. Input is terminated at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	CX	CY	CZ					
Type	F	F						
Default	0.0	0.0						

IGES Card. For TCTYPE = 2 set an IGES file.

Card 2	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	C							
Default	none							

VARIABLE	DESCRIPTION
CID	Draw bead curve ID; must be unique.
TCTYPE	Flag to indicate input curve data format: EQ.1: XYZ data, EQ.2: IGES format data.
VID	Vector ID, as defined by *DEFINE_VECTOR. This vector is used to project the supplied curves to the rigid tool, defined by the PID below.
PID	Part ID of a rigid tool to which the curves are projected and attached.
BLKID	Part ID of the blank.
PERCT	Draw bead lock percentage or draw bead force. GT.0: Percentage of the full lock force for the bead defined. This is the ratio of desired restraining force over the full lock force. The value should be between 0.0 and 100.0. LT.0: Absolute value is the draw bead force.
LCID	Load curve ID defining material hardening curve of the sheet blank, BLKID.
CX, CY, CZ	Points on the curve.
FILENAME	IGES file name.

Remarks:

1. This feature implements the following input algorithm for drawbeads:

   a) It reads a draw bead curve in either XYZ or IGES format

   b) projects the curve to the rigid tool specified

   c) creates extra node set and attaches it to the rigid tool.

   d) With supplied material hardening curve (LCID), full lock force is calculated.

   There is no need to define *CONTACT_DRAWBEAD and *CONSTRAINED_-RIGID_BODIES since they are treated internally within the code.

2. The “curve” menu in LS-PrePost can be used to break or join multiple disconnected curves, and output in either ‘XYZ’ or IGES format.

3. The following partial keyword example defines a draw bead curve ID 98 (IGES file “bead1.iges”) to restrain blank part ID 63. Full lock force is calculated from the strain hardening curve ID 400. The draw bead is projected along vector ID 991, and is attached to a rigid tool of part ID 3.

```
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+---
-8
*KEYWORD
*DEFINE_VECTOR
991,0.0,0.0,0.0,0.0,0.0,10.0
*DEFINE_CURVE_DRAWBEAD
$      CID    TCTYPE       VID       PID     BLKID     PERCT      LCID
        98         2       991         3        63    52.442       400
bead1.iges
*MAT_037
$      MID        R0         E        PR      SIGY      ETAN         R
HCLID
         1  7.89E-09  2.00E+05       0.3     240.0                 1.6
400
*DEFINE_CURVE
400
0.0,240.0
0.02,250.0
...
1.0, 350.0
*END
```

Revision information:

This feature is available starting in LS-DYNA R5 Revision 62464.

*DEFINE_CURVE_DUPLICATE

Purpose: Define a curve by optionally scaling and offsetting the abscissa and ordinates of another curve defined by the *DEFINE_CURVE keyword.

Card 1	1	2	3	4	5	6	7	8
Variable	LCID	RLCID	SFA	SFO	OFFA	OFFO		
Type	I	I	F	F	F	F		
Default	none	none	1.	1.	0.	0.		

VARIABLE	DESCRIPTION
LCID	Load curve ID. Tables (see *DEFINE_TABLE) and load curve ID's must be unique.
RLCID	Reference load curve ID.
SFA	Scale factor for abscissa value of curve ID, RLCID. This value scales the SFA value defined for RLCID. EQ.0.0: default set to 1.0.
SFO	Scale factor for ordinate value (function) of curve ID, RLCID. This value scales the SFO value defined for RLCID. EQ.0.0: default set to 1.0.
OFFA	Offset for abscissa values. This value is added to the OFFA value defined for RLCID.
OFFO	Offset for ordinate values (function). This value is added to the OFFO value defined for RLCID.

*DEFINE_CURVE_ENTITY

Purpose: Define a curve of straight line segments and circular arcs that defines an axisymmetric surface. This curve can only be used with the keyword, *CONTACT_ENTITY for the load curve entity, GEOTYP = 11.

Card	1	2	3	4	5	6	7	8
Variable	LCID	SFA	SFO	SFR	OFFA	OFFO	OFFR	
Type	I	F	F	F	F	F	F	
Default	none	1.	1.	1.	0.	0.	0.	

Point Cards. Put one point per card (3E20.0,I20). Include as many cards as needed Input is terminated when a "*" card is found.

Card	1	2	3	4	5	6	7	8
Variable	Ai		Oi		Ri		IFLAG	
Type	F		F		F		I	
Default	0.0		0.0		optional		Required if |R1| > 0	

VARIABLE	DESCRIPTION
LCID	Load curve ID. Tables (see *DEFINE_TABLE) and load curves may not share common ID's. LS-DYNA allows load curve ID's and table ID's to be used interchangeably. A unique number has to be defined.
SFA	Scale factor for axis value. This is useful for simple modifications. EQ.0.0: default set to 1.0.
SFO	Scale factor for radius values. This is useful for simple modifications. EQ.0.0: default set to 1.0.

VARIABLE	DESCRIPTION
SFR	Scale factor for circular radius. This is useful for simple modifications. EQ.0.0: default set to 1.0.
OFFA	Offset for axis values, see explanation below.
OFFO	Offset for radius values, see explanation below.
OFFR	Offset for circular radius, see explanation below.
Ai	Z-axis coordinates along the axis of rotation.
Oi	Radial coordinates from the axis of rotation
Ri	Radius of arc between points (Ai,Oi) and (Ai+1,Oi+1). If zero, a straight line segment is assumed.
IFLAG	Defined if |Ri| > 0. Set to 1 if center of arc is inside axisymmetric surface and to -1 if the center is outside the axisymmetric surface.

Remarks:

1. The load curve values are scaled after the offsets are applied, i.e.:

$$\text{Axis value} = \text{SFA} \times (\text{Defined value} + \text{OFFA})$$

$$\text{Radius value} = \text{SFO} \times (\text{Defined value} + \text{OFFO})$$

$$\text{Circular value} = \text{SFR} \times (\text{Defined value} + \text{OFFR})$$

*DEFINE_CURVE_FEEDBACK

Purpose: Define information that is used as the solution evolves to scale the ordinate values of the specified load curve ID. This keyword is usually used in connection with sheet metal forming calculations.

Card 1	1	2	3	4	5	6	7	8
Variable	LCID	PID	BOXID	FLDID				
Type	I	I	I	I				
Default	none	none	0	none				

Card 2	1	2	3	4	5	6	7	8
Variable	FSL	TSL	SFF	SFT	BIAS			
Type	F	F	F	F	F			
Default	none	none	1.0	1.0	0.0			

VARIABLE	DESCRIPTION
LCID	ID number for load curve to be scaled.
PID	Active part ID for load curve control
BOXID	Box ID. Elements of specified part ID contained in box are checked. If the box ID is set to zero the all elements of the active part are checked.
FLDID	Load curve ID which defines the flow limit diagram as shown in Figure 15-14.
FSL	If the ratio, $r = \varepsilon_{\mathrm{major}_{\mathrm{workpiece}}} / \varepsilon_{\mathrm{major}_{\mathrm{fld}}}$ exceeds FSL, then scale factor for flow, SFF, is active.
TSL	Thickness strain limit. If the thickness strain limit is exceeded, then the scale factor for thickening, SFT, is active.

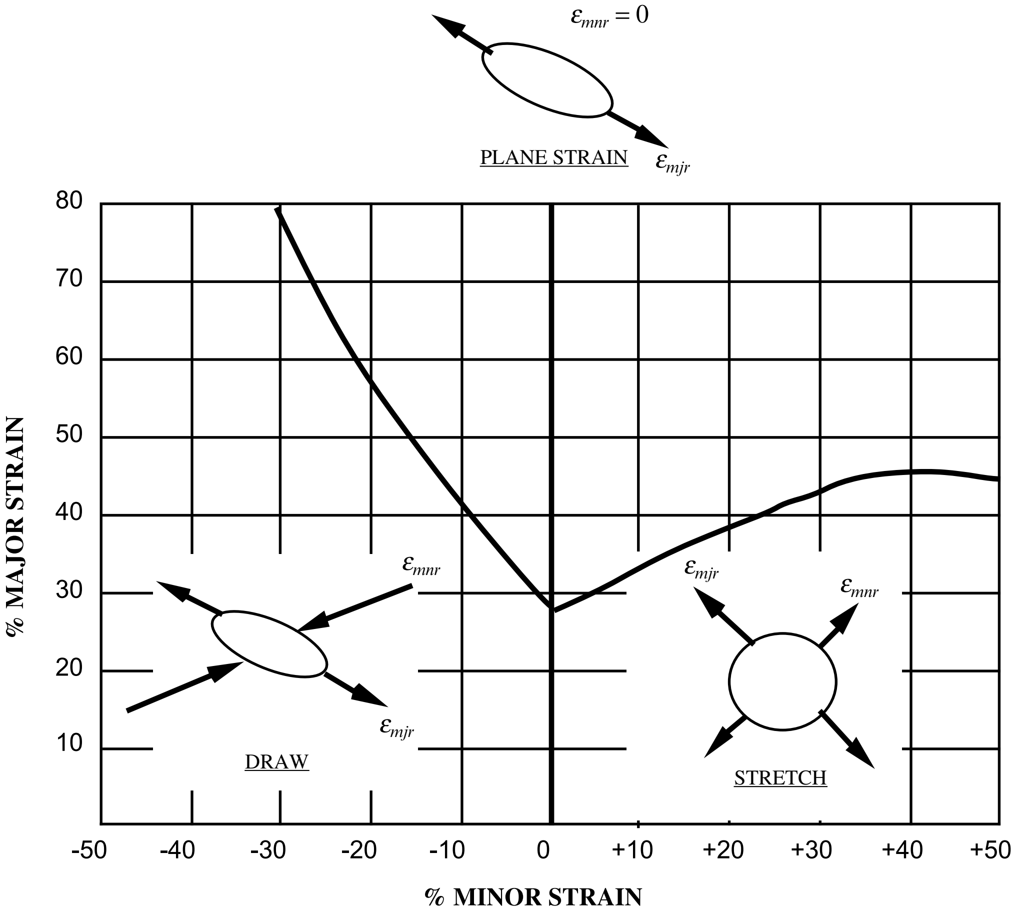


Figure 15-14. Flow limit diagram.

VARIABLE	DESCRIPTION
SFF	Scale factor for the flow limit diagram.
SFT	Scale factor for thickening.
BIAS	Bias for combined flow and thickening. Bias must be between -1 and 1.

Remarks:

This feature scales the ordinate values of a load curve according to a computed scale factor, S_f, that depends on both the major strain, r, and the through thickness, t. At each time step the load curve is scaled by S_f according to,

$$S^{n+1}_{\text{scaled load curve}} = \mathrm{S_f}(r, t) \times S^{n}_{\text{load curve}},$$

where the superscript denotes the time step. The scale factor depends on r, which is a strain measure defined as,

$$r = \frac{\varepsilon_{\mathrm{major}_{\mathrm{workpiece}}}}{\varepsilon_{\mathrm{major}_{\mathrm{fld}}}}.$$

The scale factor, then, is given by,

$$S_f = \begin{cases} 1 & r < \mathrm{FSL}, \quad t < \mathrm{TSL} \\ \mathrm{SFF} & r > \mathrm{FSL}, \quad t < \mathrm{TSL} \\ \mathrm{SFT} & r < \mathrm{FSL}, \quad t > \mathrm{TSL} \\ \frac{1}{2}(1 - \mathrm{BIAS})\times\mathrm{SFF} + \frac{1}{2}(1 + \mathrm{BIAS}) \times \mathrm{SFT} & r > \mathrm{FSL}, \quad t > \mathrm{TSL} \end{cases}$$

Usually SFF is slightly less than unity and SFT is slightly greater than unity so that $S_{\mathrm{load\ curve}}$ changes insignificantly from time step to time step.

*DEFINE_CURVE_FLC

Purpose: This keyword allows for defining Forming Limit Diagram (FLD) using sheet metal thickness '*t*' and strain hardening value '*n*', applicable to shell elements only.

This feature is available in LS-DYNA Revision 61435 and later releases.

Card 1	1	2	3	4	5	6	7	8
Variable	LCID	TH	TN					
Type	I	F	F					
Default	none	0.0	0.0					

VARIABLE	DESCRIPTION
LCID	Load curve ID.
TH	Sheet metal thickness.
TN	Strain hardening value of the sheet metal, as in power law (Swift).

Remarks:

1. This keyword is used in conjunction with keyword *MAT_TRANSVERSELY_ANISOTROPIC_ELASTIC_PLASTIC_NLP_FAILURE, and for shell elements only. For detailed formula of calculating the FLD based on sheet metal thickness and n-value, please refer to the following paper: *Ming F. Shi, Shawn Gelisse, "Issues on the AHSS Forming Limit Determination", IDDRG 2006*.

2. It is noted that this FLD calculation method is limited to sheet metal steels with thickness equal to or less than 2.5 mm, and it is not suitable for aluminum sheets.

3. In a validation example shown in Figure 15-15, a single shell element is stretched in three typical strain paths (linear): uniaxial, plane strain and equibiaxial. Strain limits for each path are recovered when the history variable (Formability Index limit in *MAT_037) reaches 1.0, shown in Figure 15-16. The top most point (strain limit) of each strain path coincides with the FLC curve calculated according to the paper, indicating the FLC defined by this keyword is working correctly. As shown in a partial keyword file below, the FLC is defined using a thickness value of 1.5 and n-value of 0.159. The 'LCID' of 891 is

used to define a variable ‘ICFLD’ in keyword *MAT_TRANSVERSELY_ANISOTROPIC_ELASTIC_PLASTIC_NLP_FAILURE.

```
*MAT_TRANSVERSELY_ANISOTROPIC_ELASTIC_PLASTIC_NLP_FAILURE
$      MID        RO         E        PR      SIGY      ETAN         R
HLCID
         1 7.830E-09 2.070E+05      0.28       0.0       0.0    -0.864
200
$      IDY        EA       COE     ICFLD
                                     891
*DEFINE_CURVE_FLC
$ LCID, TH, TN
891,1.5,0.159
$ DP600 NUMISHEET'05 Xmbr, Power law fitted
*DEFINE_CURVE
200
0.000,395.000
0.001,425.200
0.003,440.300
...
```

4. For aluminum sheets, *DEFINE_CURVE can be used to input the FLC for the variable ‘ICFLD’ in *MAT_TRANSVERSELY_ANISOTROPIC_ELASTIC_PLASTIC_NLP_FAILURE.

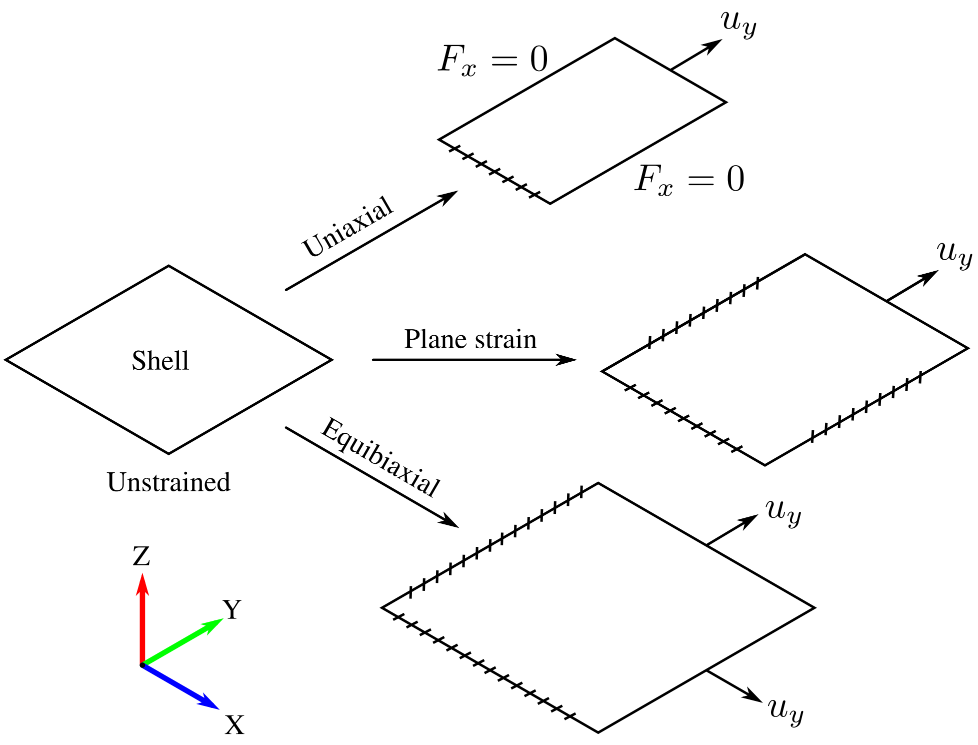


Figure 15-15. A single shell strained in three different strain paths

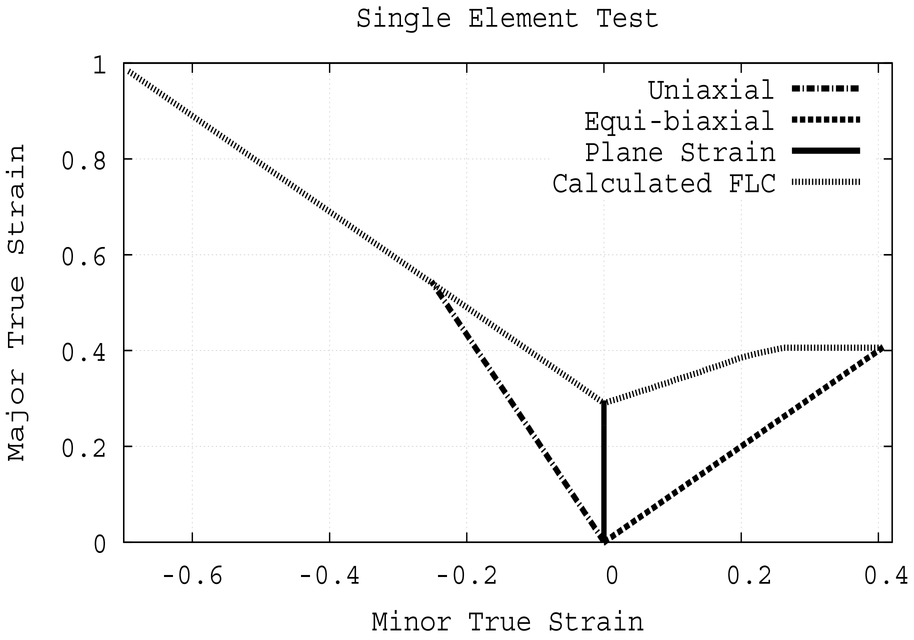


Figure 15-16. Validation of the FLC defined by this keyword

*DEFINE_CURVE_FUNCTION

Purpose: Define a curve [for example, load (ordinate value) versus time (abscissa value)] where the ordinate is given by a function expression. The function can reference other curve definition, kinematical quantities, forces, interpolating polynomials, intrinsic functions, and combinations thereof. Please note that many functions require the definition of a local coordinate system (see Remark 1 below). To output the curve to an ASCII database, see *DATABASE_CURVOUT. This command is not for defining curves for material models. Note that arguments appearing in square brackets "[ ]" are optional.

Card	1	2	3	4	5	6	7	8
Variable	LCID	SIDR						
Type	I	I						
Default	none	0						

Function Cards. Insert as many cards as needed. These cards are combined to form a single line of input. The next keyword ("*") card terminates this input.

Card	1	2	3	4	5	6	7	8
Variable	FUNCTION							
Type	A80							
Remarks	1							

VARIABLE	DESCRIPTION
LCID	Load curve ID. Tables (see *DEFINE_TABLE) and load curves may not share common ID's. LS-DYNA allows load curve ID's and table ID's to be used interchangeably. A unique number has to be defined.

VARIABLE **DESCRIPTION**

SIDR

Stress initialization by dynamic relaxation:

EQ.0: load curve used in transient analysis only or for other applications,

EQ.1: load curve used in stress initialization but not transient analysis,

EQ.2: load curve applies to both initialization and transient analysis.

FUNCTION

Arithmetic expression involving a combination of the following possibilities.

Constants and Variables:

FUNCTION	DESCRIPTION
TIME	Current simulation time
TIMESTEP	Current simulation time step
PI	Proportionality constant relating the circumference of a circle to its diameter
DTOR	Degrees to radians conversion factor ($\pi/180$)
RTOD	Radians to degrees conversion factor ($180/\pi$)

Intrinsic Functions:

FUNCTION	DESCRIPTION
ABS(a)	Absolute value of a
AINT(a)	Nearest integer whose magnitude is not larger than a
ANINT(a)	Nearest whole number to a
MOD(a_1, a_2)	Remainder when a_1 is divided by a_2
SIGN(a_1, a_2)	Transfer sign of a_2 to magnitude of a_1
MAX(a_1, a_2)	Maximum of a_1 and a_2
MIN(a_1, a_2)	Minimum of a_1 and a_2

FUNCTION	DESCRIPTION
SQRT(a)	Square root of a
EXP(a)	e raised to the power of a
LOG(a)	Natural logarithm of a
LOG10(a)	Log base 10 of a
SIN(a)	Sine of a
COS(a)	Cosine of a
TAN(a)	Tangent of a
ASIN(a)	Arc sine of a
ACOS(a)	Arc cosine of a
ATAN(a)	Arc tangent of a
ATAN2(a_1,a_2)	Arc tangent of a_1/a_2
SINH(a)	Hyperbolic sine of a
COSH(a)	Hyperbolic cosine of a
TANH(a)	Hyperbolic tangent of a

Load Curves:

FUNCTION	DESCRIPTION
LCn	Ordinate value of curve n defined elsewhere (see *DEFINE_CURVE)

FUNCTION	DESCRIPTION
DELAY$(\mathrm{LC}n, t_{\mathrm{delay}}, y_{\mathrm{def}})$	Delays curve n, defined by *DEFINE_CURVE_FUNCTION, *DEFINE_FUNCTION or DEFINE_CURVE, by T_{delay} when simulation time ≥ T_{delay}, and sets the delayed curve value to Y_{def} when time < T_{delay}, i.e., $f_{\mathrm{delay}}(t) = \begin{cases} f(t - t_{\mathrm{delay}}) & t \geq t_{\mathrm{delay}} \\ y_{\mathrm{def}} & t < t_{\mathrm{delay}} \end{cases}$ For a nonlinear curve, a T_{delay} equal to more than 5,000 time steps might compromise the accuracy and must be used with caution. When T_{delay} is a negative integer, delay time is input in terms of time step. $\lvert T_{\mathrm{delay}} \rvert$ is the number of delay time steps. In such case, $\lvert T_{\mathrm{delay}} \rvert$ is limited to a maximum of 100. For example, T_{delay} = -2 delays the curve by 2 time steps.

Coordinate Functions:

FUNCTION	DESCRIPTION
CX(n)	Value of x-coordinate for node n.
CY(n)	Value of y-coordinate for node n.
CZ(n)	Value of z-coordinate for node n.

Displacement Functions:

FUNCTION	DESCRIPTION
DM$(n_1[, n_2])$	Magnitude of translational displacement of node n_1 relative to node n_2. Node n_2 is optional and if omitted the displacement is computed relative to ground.
DMRB(n)	Magnitude of translational displacement of rigid body having a part ID of n

FUNCTION	DESCRIPTION
DX(n_1[,n_2,n_3])	x-translational displacement of node n_1 relative to node n_2 expressed in the local coordinate system of node n_3. In other words, at any time t, the function returns the component of relative displacement that lies in the x-direction of the local coordinate system at time = t. If node n_2 is omitted it defaults to ground. If node n_3 is not specified the displacement is reported in the global coordinate system.
DY(n_1[,n_2,n_3])	y-translational displacement of node n_1 relative to node n_2 expressed in the local coordinate system of node n_3. In other words, at any time t, the function returns the component of relative displacement that lies in the y-direction of the local coordinate system at time = t. If node n_2 is omitted it defaults to ground. If node n_3 is not specified the displacement is reported in the global coordinate system.
DZ(n_1[,n_2,n_3])	z-translational displacement of node n_1 relative to node n_2 expressed in the local coordinate system of node n_3. In other words, at any time t, the function returns the component of relative displacement that lies in the z-direction of the local coordinate system at time = t. If node n_2 is omitted it defaults to ground. If node n_3 is not specified the displacement is reported in the global coordinate system.
DXRB(n)	x-translational displacement of rigid body having a part ID of n
DYRB(n)	y-translational displacement of rigid body having a part ID of n
DZRB(n)	z-translational displacement of rigid body having a part ID of n
AX(n_1[,n_2])	Rotation displacement of node n_1 about the local x-axis of node n_2. If n_2 is not specified then it defaults to ground. In computing this value it is assumed the rotation about the other two axes (y-, z-axes) of node n_2 is zero.
AY(n_1[,n_2])	Rotation displacement of node n_1 about the local y-axis of node n_2 . If n_2 is not specified then it defaults to ground. In computing this value it is assumed the rotation about the other two axes (x-, z-axes) of node n_2 is zero. See Remark 1.
AZ(n_1[,n_2])	Rotation displacement of node n_1 about the local z-axis of node n_2. If n_2 is not specified then it defaults to ground. In computing this value it is assumed the rotation about the other two axes (x-, y-axes) of node n_2 is zero. See Remark 1.

FUNCTION	DESCRIPTION
PSI(n_1[,n_2])	First angle in the body2:313 Euler rotation sequence which orients node n_1 in the frame of node n_2. If n_2 is omitted it defaults to ground. See Remark 1.
THETA(n_1[,n_2])	Second angle in the body2:313 Euler rotation sequence which orients node n_1 in the frame of node n_2. If n_2 is omitted it defaults to ground. See Remark 1.
PHI(n_1[,n_2])	Third angle in the body2:313 Euler rotation sequence which orients node n_1 in the frame of node n_2. If n_2 is omitted it defaults to ground. See Remark 1.
YAW(n_1[,n_2])	First angle in the body3:321 yaw-pitch-roll rotation sequence which orients node n_1 in the frame of node n_2. If n_2 is omitted it defaults to ground. See Remark 1.
PITCH(n_1[,n_2])	Second angle in the body3:321 yaw-pitch-roll rotation sequence which orients node n_1 in the frame of node n_2. If n_2 is omitted it defaults to ground. See Remark 1.
ROLL(n_1[,n_2])	Third angle in the body3:321 yaw-pitch-roll rotation sequence which orients node n_1 in the frame of node n_2. If n_2 is omitted it defaults to ground. See Remark 1.

Velocity Functions:

FUNCTION	DESCRIPTION
VM(n_1[,n_2])	Magnitude of translational velocity of node n_1 relative to node n_2. Node n_2 is optional and if omitted the velocity is computed relative to ground.
VR(n_1[,n_2])	Relative radial translational velocity of node n_1 relative to node. If node n_2 is omitted it defaults to ground.
VX(n_1[,n_2,n_3])	x-component of the difference between the translational velocity vectors of node n_1 and node n_2 in the local coordinate system of node n_3. If node n_2 is omitted it defaults to ground. Node n_3 is optional and if not specified the global coordinate system is used.

FUNCTION **DESCRIPTION**

VY(n_1[,n_2,n_3]) y-component of the difference between the translational velocity vectors of node n_1 and node n_2 in the local coordinate system of node n_3. If node n_2 is omitted it defaults to ground. Node n_3 is optional and if not specified the global coordinate system is used.

VZ(n_1[,n_2,n_3]) z-component of the difference between the translational velocity vectors of node n_1 and node n_2 in the local coordinate system of node n_3. If node n_2 is omitted it defaults to ground. Node n_3 is optional and if not specified the global coordinate system is used.

WM(n_1[,n_2]) Magnitude of angular velocity of node n_1 relative to node n_2. Node n_2 is optional and if omitted the angular velocity is computed relative to ground.

WX(n_1[,n_2,n_3]) x-component of the difference between the angular velocity vectors of node n_1 and node n_2 in the local coordinate system of node n_3. If node n_2 is omitted it defaults to ground. Node $n_3$is optional and if not specified the global coordinate system is used. See Remark 1.

WY(n_1[,n_2,n_3]) y-component of the difference between the angular velocity vectors of node n_1 and node $n_2$in the local coordinate system of node n_3. If node n_2 is omitted it defaults to ground. Node $n_3$is optional and if not specified the global coordinate system is used. See Remark 1.

WZ(n_1[,n_2,n_3]) z-component of the difference between the angular velocity vectors of node n_1 and node n_2 in the local coordinate system of node n_3. If node $n_2$is omitted it defaults to ground. Node $n_3$is optional and if not specified the global coordinate system is used. See Remark 1.

Acceleration Functions:

FUNCTION **DESCRIPTION**

ACCM(n_1[,n_2]) Magnitude of translational acceleration of node n_1 relative to node n_2. Node n_2 is optional and if omitted the acceleration is computed relative to ground. See Remark 1.

FUNCTION	DESCRIPTION
ACCX(n_1[,n_2,n_3])	x-component of the difference between the translational acceleration vectors of node n_1 and node n_2 in the local coordinate system of node n_3. If node n_2 is omitted it defaults to ground. Node n_3 is optional and if not specified the global coordinate system is used. See Remark 1.
ACCY(n_1[,n_2,n_3])	y-component of the difference between the translational acceleration vectors of node n_1 and node n_2 in the local coordinate system of node n_3. If node n_2 is omitted it defaults to ground. Node $n_3$is optional and if not specified the global coordinate system is used. See Remark 1.
ACCZ(n_1[,n_2,n_3])	z-component of the difference between the translational acceleration vectors of node n_1 and node n_2 in the local coordinate system of node n_3. If node n_2 is omitted it defaults to ground. Node n_3 is optional and if not specified the global coordinate system is used. See Remark 1.
WDTM(n_1[,n_2])	Magnitude of angular acceleration of node n_1 relative to node n_2. Node n_2 is optional and if omitted the angular acceleration is computed relative to ground. See Remark 1.
WDTX(n_1[,n_2,n_3])	x-component of the difference between the angular acceleration vectors of node n_1 and node n_2 in the local coordinate system of node n_3. If node n_2 is omitted it defaults to ground. Node n_3 is optional and if not specified the global coordinate system is used. See Remark 1.
WDTY(n_1[,n_2,n_3])	y-component of the difference between the angular acceleration vectors of node $n_1$and node n_2 in the local coordinate system of node n_3. If node n_2 is omitted it defaults to ground. Node n_3 is optional and if not specified the global coordinate system is used. See Remark 1.
WDTZ(n_1[,n_2,n_3])	z-component of the difference between the angular acceleration vectors of node n_1 and node n_2 in the local coordinate system of node n_3. If node n_2 is omitted it defaults to ground. Node n_3 is optional and if not specified the global coordinate system is used. See Remark 1.

Generic Force Functions:

FUNCTION **DESCRIPTION**

FM($n_1[,n_2]$)

Magnitude of the SPC force acting on node n_1 minus the force acting on node n_2. Node n_2 is optional and if omitted the force that acting only on n_1 Is returned. See Remark 1.

FX($n_1[,n_2,n_3]$)

x-component of SPC force acting on node n_1 as computed in the optional local system of node n_3. If n_2 is specified, the force acting on n_2 is subtracted from the force acting on n_1. See Remark 1.

FY($n_1[,n_2,n_3]$)

y-component of SPC force acting on node n_1 as computed in the optional local system of node n_3. If n_2 is specified, the force acting on n_2 is subtracted from the force acting on n_1. See Remark 1.

FZ($n_1[,n_2,n_3]$)

z-component of SPC force acting on node n_1 as computed in the optional local system of node n_3. If n_2 is specified, the force acting on n_2 is subtracted from the force acting on n_1. See Remark 1.

TM($n_1[,n_2]$)

Magnitude of SPC torque acting on node n_1 minus the torque acting node n_2. Node n_2 is optional and if omitted the torque that acting only on n_1. See Remark 1.

TX($n_1[,n_2,n_3]$)

x-component of the SPC torque acting on node n_1 as computed in the optional local system of node n_3. If n_2 is specified, the torque acting on n_2 is subtracted from the torque acting on n_1. See Remark 1.

TY($n_1[,n_2,n_3]$)

y-component of the SPC torque acting on node n_1 as computed in the optional local system of node n_3. If n_2 is specified, the torque acting on n_2 is subtracted from the torque acting on n_1. See Remark 1.

TZ($n_1[,n_2,n_3]$)

z-component of the SPC torque acting on node n_1 as computed in the optional local system of node n_3. If n_2 is specified, the torque acting on n_2 is subtracted from the torque acting on n_1. See Remark 1.

Sensor Functions:

FUNCTION	DESCRIPTION
SENSOR(n)	Returns a value of 1.0 if *SENSOR_CONTROL of control ID n has a status of “on”. If the sensor has a status of “off”, then the returned value is equal to the value of the TYPEID field on *SENSOR_CONTROL when the TYPE field is set to “function,” otherwise SENSOR(n) returns zero.
SENSORD($n, idflt$)	Returns the current value of *SENSOR_DEFINE sensor having ID n. Idflt is the optional filter ID defined using *DEFINE_-FILTER.

Contact Force Functions:

FUNCTION	DESCRIPTION
RCFORC(*id*, *ims*, *comp*, *local*)	Returns the component *comp* (see description below) of contact interface *id* (see *CONTACT_..._ID) as calculated in the local coordinate system *local* (see *DEFINE_COORDINATE_...). If *local* equals zero then forces are reported in the global coordinate system. Forces are reported for the slave side when *ims* = 1 or master side when *ims* = 2. Following are the admissible values of *comp* and their corresponding force component. comp.EQ.1: x force component comp.EQ.2: y force component comp.EQ.3: z force component comp.EQ.4: resultant force

Element Specific Functions:

FUNCTION	DESCRIPTION
BEAM(*id*, *jflag*, *comp*, *rm*)	Returns the force component *comp* (see description below) of beam *id* as calculated in the local coordinate system *rm*. Forces are reported in the global coordinate system if *rm* is zero. If *rm* equals -1 the beam’s *r*, *s*, and *t* force/moment is returned. If *jflag* is set to zero

FUNCTION **DESCRIPTION**

then the force/torque acting on n_1 end of the beam is returned, otherwise if *jflag* is set to 1 the force/torque on the n_2 end of the beam is returned. See *ELEMENT_BEAM for the nodal connectivity rule defining n_1 and n_2.

Admissible values of *comp* are 1-8 and correspond to the following components.

comp.EQ.1: force magnitude

comp.EQ.2: x force (axial r-force, rm = -1)

comp.EQ.3: y force (s-shear force, rm = -1)

comp.EQ.4: z force (t-shear force, rm = -1)

comp.EQ.5: torque magnitude

comp.EQ.6: x torque (torsion, rm = -1)

comp.EQ.7: y torque (s-moment, rm = -1)

comp.EQ.8: z torque (t-moment, rm = -1)

ELHIST(*eid*, *etype*, *comp*, *ipt*, *local*)

Returns the elemental quantity *comp* (see description below) of element e*id* as calculated in the local coordinate system *local*. Quantities are reported in the global coordinate system if *local* is zero. The parameter *ipt* specifies whether the quantity is for particular integration point or maximum, minimum, or averaging is applied across the integration points.

The following element classes, specified with *etype*, are supported.

etype.EQ.0: solid

etype.EQ.2: thin shell

Following are admissible values of *comp* and the corresponding elemental quantity.

comp.EQ.1: x stress

comp.EQ.2: y stress

comp.EQ.3: z stress

comp.EQ.4: xy stress

FUNCTION	DESCRIPTION
	comp.EQ.5: *yz* stress comp.EQ.6: *zx* stress comp.EQ.7: effective plastic strain comp.EQ.8: hydrostatic pressure comp.EQ.9: effective stress comp.EQ.11: *x* strain comp.EQ.12: *y* strain comp.EQ.13: *z* strain comp.EQ.14: *xy* strain comp.EQ.15: *yz* strain comp.EQ.16: *zx* strain
	Integration point options, specified with *ipt*, follow. ipt.GE.1: quantity is reported for integration point number ipt ipt.EQ.-1: maximum of all integration points (default) ipt.EQ.-2: average of all integration points ipt.EQ.-3: minimum of all integration points ipt.EQ.-4: lower surface integration point ipt.EQ.-5: upper surface integration point ipt.EQ.-6: middle surface integration point
	The local coordinate option *local* currently defaults to the global coordinate system for solid elements and other coordinate system options are unavailable. In the case of thin shell elements the quantity is reported only in the element local coordinate system. local.EQ.1: global coordinate system (solid elements) local.EQ.2: element coordinate system (thin shell elements)

FUNCTION **DESCRIPTION**

JOINT(*id*, *jflag*, *comp*, *rm*)

Returns the force component *comp* (see description below) due to rigid body joint *id* as calculated in the local coordinate system *rm*. If *jflag* is set to zero then the force/torque acting on n_1 end of the joint is returned. The force/torque on the n_2 end of the joint is returned if jflag is set to 1. See *CONSTRAINED_JOINT for the rule defining n1 and n_2.

Admissible values of *comp* are 1-8 and correspond to the following components.

comp.EQ.1: force magnitude

comp.EQ.2: x force

comp.EQ.3: y force

comp.EQ.4: z force

comp.EQ.5: torque magnitude

comp.EQ.6: x torque

comp.EQ.7: y torque

comp.EQ.8: z torque

Nodal Specific Functions:

FUNCTION	DESCRIPTION
TEMP(n)	Returns the temperature of node n

General Functions

FUNCTION **DESCRIPTION**

CHEBY($x, x_0, a_0, \ldots, a_{30}$)

Evaluates a Chebyshev polynomial at the user specified value x. The parameters x_0, a_0, a_1, …, a_{30} are used to define the constants for the polynomial defined by:

$$C(x) = \sum a_j T_j(x - x_0)$$

where the functions T_j is defined recursively as

FUNCTION **DESCRIPTION**

$$T_j(x-x_0) = 2(x-x_0)T_{j-1}(x-x_0) - T_{j-2}(x-x_0)$$

where

$$T_0(x-x_0) = 1$$

$$T_1(x-x_0) = x - x_0$$

$\mathrm{FORCOS}(x, x_0, \omega[, a_0, \dots, a_{30}])$

Evaluates a Fourier cosine series at the user specified value x. The parameters x0, a0, a1, …, a30 are used to define the constants for the series defined by:

$$F(x) = \sum a_j T_j(x-x_0)$$

where

$$T_j(x-x_0) = \cos[j\omega(x-x_0)]$$

$\mathrm{FORSIN}(x, x_0, \omega[, a_0, \dots, a_{30}])$

Evaluates a Fourier sine series at the user specified value x. The parameters x_0, a_0, a_1, …, a_{30} are used to define the constants for the series defined by:

$$F(x) = \sum a_j T_j(x-x_0)$$

where

$$T_j(x-x_0) = \sin[j\omega(x-x_0)]$$

$\mathrm{IF}(a_1, a_2, a_3, a_4)$

Arithmetic if conditional where a_i can be a constant or any legal expression described in *DEFINE_CURVE_FUNCTION. For example, a_1=*'CX(100)'* sets the first argument to be the x-coordinate of node 100.

$$\mathrm{IF} = \begin{cases} \text{the value of } a_2 & \text{if the value of } a_1 < 0 \\ \text{the value of } a_3 & \text{if the value of } a_1 = 0 \\ \text{the value of } a_4 & \text{if the value of } a_1 > 0 \end{cases}$$

PIDCTL(*meas, ref, kp, ki, kd, tf, ei0, sint, umin, umax*)

Evaluates the control signal of a PID controller

$$u(t) = kp \times e(t) + ki \times \int_0^t e(\tau)d\tau + kd \times \frac{\mathrm{d}e(t)}{\mathrm{d}t}$$

where $e(t)$ is the control error defined as the difference between the reference value *ref* and the measured value, *meas*

$$e(t) = ref - meas$$

FUNCTION	DESCRIPTION
	The control parameters are proportional gain *kp*, integral gain *ki*, derivative gain *kd* and low-pass filter *tf* for the derivative calculation $\frac{\mathrm{d}e(t_n)}{\mathrm{d}t} = \frac{\mathrm{d}e(t_{n-1})}{\mathrm{d}t}\frac{tf}{\Delta t + tf} + \frac{\Delta t}{\Delta t + tf} \times \frac{e(t_n) - e(t_{n-1})}{\Delta t}$ *Ei0* is the initial integral value at time = 0. *Sint* is the sampling interval. *Umin* and *umax*, the lower and upper limit of a control signal, can be used to represent the saturation limits of an actuator. When the signal is not within the limits, it is clipped to the saturation limit, i.e., integration is skipped to avoid integrator wind-up. Input parameter can be a constant or any legal expression described in *DEFINE_CURVE_FUNCTION. For example, *meas='CX(100)'* measures the x-coordinate of node 100, *ref* ='LC200' uses curve 200 as the reference value.
$\mathrm{POLY}(x, x_0, a_0, \dots, a_{30})$	Evaluates a standard polynomial at the user specified value x. The parameters x_0, a_0, a_1, …, a_{30} are used to define the constants for the polynomial defined by: $P(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)^2 + \cdots + a_n(x - x_0)^n$
$\mathrm{SHF}(x, x_0, a, \omega[, \phi, b])$	Evaluates a Fourier sine series at the user specified value x. The parameters x_0, a_0, a_1, …, a_{30} are used to define the constants for the series defined by: $\mathrm{SHF} = a\sin[\omega(x - x_0) - \varphi] + b$
$\mathrm{STEP}(x, x_0, h_0, x_1, h_1)$	Approximates the Heaviside function with a cubic polynomial using the equation:

$$\mathrm{STEP} = \begin{cases} h_0 & \text{if } x \le x_0 \\ h_0 + (h_1 - h_0)\left[\frac{(x - x_0)}{(x_1 - x_0)}\right]^2 \left\{3 - 2\left[\frac{(x - x_0)}{(x_1 - x_0)}\right]\right\} & \text{if } x < x < x_1 \\ h_1 & \text{if } x \ge x_1 \end{cases}$$

Electromagnetic solver (EM) Functions

FUNCTION	DESCRIPTION
EM_ELHIST(*iele*, *ifield*, *idir*)	Returns the elemental quantity of element *iele* in the global reference frame.
EM_NDHIST(*inode*, *ifield*, *idir*)	Returns the nodal quantity of node *inode* in the global reference frame.
RM_PAHIST(*ipart*, *ifield*, *idir*)	Returns the value integrated over the whole part given by *ipart*. *ifield* can be 7, 8 and 11 only.

Admissible values of *ifield* are 1-10 and correspond to the following variables.

comp.EQ.1: scalar potential

comp.EQ.2: vector potential

comp.EQ.3: electric field

comp.EQ.4: **B** field

comp.EQ.5: **H** field

comp.EQ.6: current density

comp.EQ.7: Lorentz force

comp.EQ.8: current density

comp.EQ.9: Lorentz force

comp.EQ.10: relative permeability

comp.EQ.11: magnetic energy (in the conductor only)

Admissible values of *idir* are 1-4 and correspond to the following components.

comp.EQ.1: *x*-component

comp.EQ.2: *y*-component

comp.EQ.3: *z*-component

comp.EQ.4: Norm

Remarks:

1. **Local Coordinate Systems Required for Rotational Motion.** A local coordinate system *must* be attached to nodes if they are referenced by functions

involving rotational motion, for example, angular displacement or angular velocity. The local coordinate system is attached to the node using *DEFINE_-COORDINATE_NODES and FLAG=1 is a requirement. Furthermore, the three nodes which comprise the coordinate system must lie on the same body. Similarly, a local coordinate system must also be attached to node n_3 if n_3 is referenced in functions: DX, DY, DZ, VX, VY, VZ, WX, WY, WZ, ACCX, ACCY, ACCZ, WDTX, WDTY, WDTZ, FX, FY, FZ, TX, TY, or TZ.

2. **Default is Radians.** Unless otherwise noted units of radians are always used for the arguments and output of functions involving angular measures.

3. The following examples serve only as an illustration of syntax.

Example 1:

Define a curve 10 whose ordinate is,

$$f(x) = \frac{1}{2}(\text{ordinate of load curve 9}) \times (\text{magnitude of translation velocity at node 22})^3.$$

```
*DEFINE_CURVE_FUNCTION
10
0.5*lc9*vm(22)**3
```

Example 2:

Define a curve 101 whose ordinate is,

$$f(x) = -2(\text{z translational displacement of node 38}) \times \sin(20\pi t).$$

```
*DEFINE_CURVE_FUNCTION
101
-2.*dz(38)*sin(2.*pi*10.*time)
```

Example 3:

Define a curve 202 whose ordinate is,

$$f(x) = \begin{cases} \cos(4\pi t)\ if\ t \le 5. \\ \quad 0.\ if\ t\ > 5. \end{cases}$$

```
*DEFINE_CURVE_FUNCTION
202
If(TIME-5.,COS(4.*PI*TIME),COS(4.*PI*TIME),0.)
```

*DEFINE_CURVE_SMOOTH

Purpose: Define a smoothly varying curve using few parameters. This shape is useful for velocity control of tools in metal forming applications.

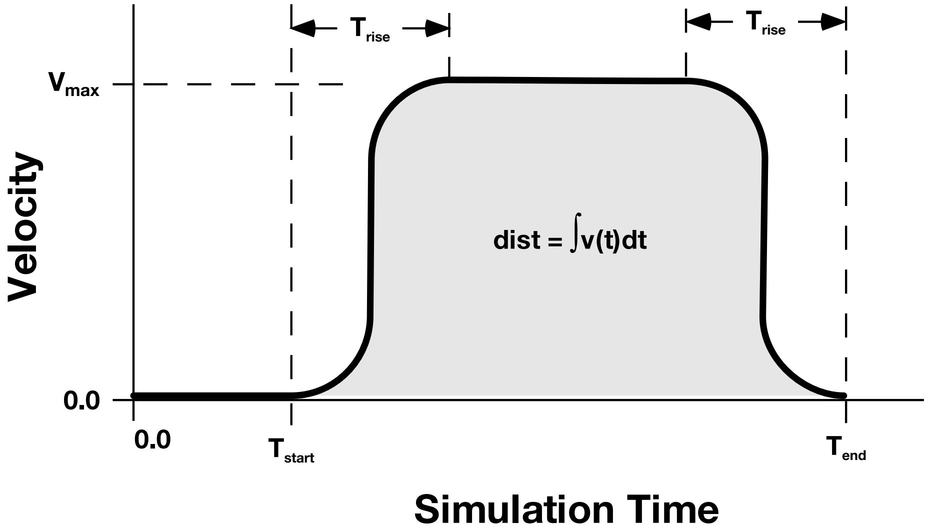


Figure 15-17. Smooth curve created automatically using *DEFINE_CURVE_-SMOOTH. This shape is commonly used to control velocity of tools in metal forming applications as shown in the above graph, but can be used for other applications in place of any standard load curve.

Card	1	2	3	4	5	6	7	8
Variable	LCID	SIDR	DIST	TSTART	TEND	TRISE	VMAX	
Type	I	I	F	F	F	F	F	
Default	none	none	none	none	none	none	none	

VARIABLE	DESCRIPTION
LCID	Load curve ID, must be unique.

VARIABLE **DESCRIPTION**

SIDR

Stress initialization by dynamic relaxation:

EQ.0: load curve used in transient analysis only or for other applications,

EQ.1: load curve used in stress initialization but not transient analysis,

EQ.2: load curve applies to both initialization and transient analysis.

DIST

Total distance tool will travel (area under curve).

TSTART

Time curve starts to rise

TEND

Time curve returns to zero. If TEND is nonzero, VMAX will be computed automatically to satisfy required travel distance DIST. Input either TEND or VMAX.

TRISE

Rise time

VMAX

Maximum velocity (maximum value of curve). If VMAX is nonzero, TEND will be computed automatically to satisfy required travel distance DIST. Input either TEND or VMAX.

DEFINE_CURVE_TRIM_*{OPTION}

Available options include:

<BLANK>

3D

NEW

Purpose: This keyword is developed to define curves and controls for sheet blank trimming in sheet metal forming. It can also be used to define mesh adaptivity along a curve prior to the start of a simulation, using variable TCTOL and keyword *CONTROL_ADAPTIVE_CURVE.

When the option 3D is used, the trimming is processed based on the element normal rather than a vector. The option NEW is used to trim in a fixed direction specified by a vector, and is also called 2D trimming. Currently this keyword applies to:

- 2D and 3D trimming of shell elements,
- 2D and 3D trimming of solids,
- 2D and 3D adaptive trimming of adaptive-meshed sandwiched parts (limit to a core of one layer of solid elements with outer layers of shell elements, see “IFSAND” under *CONTROL_ADAPTIVE),
- 2D and 3D trimming of non-adaptive sandwiched parts (a core of multiple layers of solid elements with outer layers of shell elements), and,
- 2D trimming of thick shell elements (TSHELL).

This keyword is not applicable to axisymmetric solids or 2D plane strain/stress elements. Related keywords include *ELEMENT_TRIM, *CONTROL_FORMING_TRIMMING, *CONTROL_ADAPTIVE_CURVE, *INCLUDE_TRIM, and *INCLUDE. Another closely related keyword is *CONTROL_FORMING_TRIM_MERGE, which automatically closes an open trim curve with a user-specified tolerance.

Trimming of shell and solid elements are supported starting in *LS-PrePost 4.0* and *LS-PrePost 4.3*, respectively, under *Application → MetalForming → Easy Setup*.

Card 1	1	2	3	4	5	6	7	8
Variable	TCID	TCTYPE	TFLG	TDIR	TCTOL	TOLN / IGB	NSEED1	NSEED2
Type	I	I	I	I	F	F	I	I
Default	none	1	none	0	0.25	2.0 / 0	none	none
Remarks				Fig. 15-18	Fig. 15-19			

Point Cards. Additional cards for TCTYPE = 1. Put one point per card (2E20.0). Input is terminated at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	CX		CY		CZ			
Type	F		F		F			
Default	0.0		0.0		0.0			

IGES File Card. Additional card for TCTYPE = 2.

Card 3	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	C							

VARIABLE	DESCRIPTION
TCID	ID number for trim curve.

VARIABLE	DESCRIPTION
TCTYPE	Trim curve type: EQ.1: Curve data in XYZ format, obtained following procedures outlined in Figures under *INTERFACE_BLANKSIZE. In addition, only this format is allowed in *INTERFACE_COMPENSATION_NEW. EQ.2: IGES trim curve.
TFLG	Element removal option (applies to option **NEW** only): EQ.-1: remove material outside curve; EQ.1: remove material inside curve.
TDIR	If the option **NEW** is used, this is the ID of a vector (*DEFINE_VECTOR) giving the trim direction (see Figure 15-18). EQ.0: default vector (0.0,0.0,1.0) is used. Curve is defined in global XY plane, and projected onto mesh in global Z-direction to define trim line. If the option **3D** is used, TDIR is used to indicate whether the trim curve is near the top or bottom surface of the solids or laminates (>Revision 101964), see Trimming. EQ.1: trim curve is located near the top surface (default). EQ.-1: trim curve is located near the bottom surface.
TCTOL	Tolerance limiting size of small elements created during trimming (see Figure 15-19). LT.0: "simple" trimming, producing jagged edge mesh When used together with *CONTROL_ADAPTIVE_CURVE, it is a distance from the curve out (both sides). Within this distance the blank mesh will be refined, as stated in remarks below.

VARIABLE **DESCRIPTION**

TOLN / IGB

If the option **3D** is used, TOLN represents the maximum gap between the trimming curve and the mesh. If the gap is bigger than this value, this section in the curve will not be used.

If the option **NEW** is used, then the variable IGB is defined as follows:

IGB.EQ.0: trimming curve is defined in local coordinate system. This is the default value. If this value is chosen for IGB, then the variable TDIR and the keyword *DEFINE_VECTOR need to be defined according to Figure 15-18.

IGB.EQ.1: trimming curve is defined in global coordinate system.

NSEED1/ NSEED2

A node ID on the blank in the area that remains after trimming, applicable to both options 3D or NEW.

LT.0: positive number is a node ID, which may not necessarily be from the blank, referring to remarks below.

CX, CY, CZ

X, Y, Z-coordinate of trim curve. Define if and only if TCTYPE = 1.

FILENAME

Name of IGES database containing trim curve(s). Define if and only if TCTYPE = 2.

Trimming capability summary:

This keyword and its options deal with trimming of the following scenarios:

	2D (along one direction)	**3D (element normal)**	**2D & 3D Double Trim**	**Adaptive mesh**
Shell	Yes	Yes	Yes	Yes
Solids	Yes	Yes	Yes	N/A
Laminates	Yes	Yes	Yes	One layer of solids only; Multiple layers of solids okay for non-adaptive mesh.

TSHELL	Yes	N/A	N/A	N/A

About the options and trim curves:

The option NEW activates a new searching algorithm, which enables a much faster trimming operation compared with option 3D. For big models, the improvement in computational efficiency of the NEW option is significant. In addition, users are required to pick a seed node (or position coordinates), as is the case with the 3D option. Both options are now available under the "*Trimming*" feature of *LS-PrePost 4.0*'s eZ-Setup for metal forming application (http://ftp.lstc.com/anonymous/outgoing/lsprepost/4.0/metalforming/). Only IGES entities 110 and 106 are supported when using the TCTYPE of 2. The eZ-Setup for trimming function ensures correct IGES files are written for trimming simulation.

For the option NEW, Revision 68643 and later releases enable trimming of a part where trim lines go beyond the part boundary. This is illustrated in Figure 15-24.

Enclosed trimming curves (same start and end points) are required for all options. Furthermore, for each enclosed trimming curve, only one curve segment is acceptable for the option 3D; while several curve segments are acceptable with the option NEW. Curves can be manipulated through the use of *Merge* and *break* features in *LS-PrePost4.0*, found under *Curve/Merge* (always select *piecewise* under *Merge*) and *break*.

In case of **3D** trimming, trim curves need to be sufficiently close to the part. A feature of curve projection to the mesh in *LS-PrePost* can be used to process the trim curves. The feature is accessible under *GeoTol/Project/Closest Proj/Project to Element/By Part*. Double precision LS-DYNA executable may also help in this situation.

Choice of 2D or 3D trimming depends on the to-be-trimmed part geometry. 2D trimming can be used if trimming is to be done on the relatively flat area of the part, while 3D trimming must be selected for trimming on a inclined or vertical draw wall area for precise trimming.

Seed node definition:

This keyword in combination with *ELEMENT_TRIM trims the requested parts before a job starts (pre-trimming), and can handle adaptive mesh. If the keyword *ELEMENT_TRIM does not exist the parts are trimmed after the job is terminated (post-trimming).

Seed node is used to define which side of the drawn panel to be kept after the trimming. With the frequent application of adaptive re-meshing, the seed node for trimming is often unknown until the draw forming is complete. With the negative NSEED variable, an extra node unrelated to the blank and tools can be created for the definition of the seed node, enabling trimming process independent of the previous process simulation results. The extra node can be defined using keyword *NODE. A partial keyword input example for the trimming of a double-attached *NUMISHEET2002 fender outer* with the option NEW is listed below, where a 2D trimming is performed with IGES file *doubletrim.iges* in the global Z-axis, with two nodes of negative ID 43356 and 18764 assigned to the variables NSEED1 and NSEED2, respectively. The two seed nodes are defined off the stationary lower post, and do not necessarily need to be a part of the post, as shown in Figure 15-20. The drawn panels in wire frame are shown in Figure 15-21, along with the thickness/thinning contour (Figure 15-22). In Figure 15-23, the drawn panels are trimmed and separated.

```
*KEYWORD
*CONTROL_TERMINATION
0.000
*CONTROL_SHELL
......
*CONTROL_OUTPUT
......
*DATABASE_BINARY_D3PLOT
......
*DATABASE_EXTENT_BINARY
......
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*SET_PART_LIST
......
*PART
Blank
......
*SECTION_SHELL
......
*MAT_3-PARAMETER_BARLAT
......
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*INCLUDE_TRIM
drawn.dynain
*ELEMENT_TRIM
         1
*DEFINE_CURVE_TRIM_NEW
$#    TCID    TCTYPE      TFLG      TDIR     TCTOL      TOLN    NSEED1    NSEED2
         1         2                   0     0.250         1    -43356    -18764
doubletrim.iges
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*NODE
18764,-184.565,84.755,78.392
43356,-1038.41,119.154,78.375
*INTERFACE_SPRINGBACK_LSDYNA
......
*END
```

If the seed node is too far away from the blank it will be projected to the blank and the new position will be used as the seed node. Typically, this node can be selected from the stationary tool in its home position.

Alternatively, if the variable NSEEDs are not defined, the seeds can be defined using *DEFINE_TRIM_SEED_POINT_COORDINATES. A partial keyword input is provided below for trimming of the same double-attached fender outer.

```
*INCLUDE_TRIM
drawn.dynain
*ELEMENT_TRIM
         1
*DEFINE_CURVE_TRIM_NEW
$#    TCID    TCTYPE      TFLG      TDIR     TCTOL      TOLN    NSEED1    NSEED2
         1         2                   0     0.250         1
doubletrim.iges
*DEFINE_TRIM_SEED_POINT_COORDINATES
$    NSEED        X1        Y1        Z1        X2        Y2        Z2
         2  -184.565    84.755    78.392  -1038.41   119.154    78.375
```

Again, selecting a seed node is quite easy in *"Trimming"* process of *LS-PrePost4.0* eZSetup for metal forming application.

General adaptive re-meshing and element fixing during trimming:

In case of large element size along the trim curves, the blank mesh can be pre-adapted along the trim curves before trimming by adding the keyword *CONTROL_ADAPTIVE_CURVE to the above example for a better quality trim edge. The following indicates refining meshes for part set ID 1 no more than three levels along the trim curves, or until element size reaches 3.0: Care should be taken since too small of the value SMIN and too large value of N could result in excessive amount of elements to be generated.

```
*CONTROL_ADAPTIVE_CURVE
$#   IDSET     ITYPE         N      SMIN   ITRIOPT
         1         2         3       3.0         0
```

Sometimes it is helpful to conduct a check of the trimmed mesh along the edge in the same trimming input deck using the keyword *CONTROL_CHECK_SHELL. This is especially useful for the next continued process simulation. For detailed usage, check for an updated remarks for the keyword.

The trimming tolerance TCTOL limits the size of the smallest element created during trimming. A value of 0.0 places no limit on element size. A value of 0.5 restricts new elements to be at least half of the size of the parent element. A value of 1.0 allows no new elements to be generated, only repositioning of existing nodes to lie on the trim curve. A negative tolerance value activates "simple" trimming, where entire elements are removed, leaving a jagged edge.

TCTOL used as mesh refinement width along a curve:

The variable TCTOL can be used to control the mesh refinement along a curve when used together with *CONTROL_ADAPTIVE_CURVE. In this scenario, it is the distance from both sides of the curve within which the mesh will be refined. The mesh will be refined in the beginning of the simulation. This method offers greater control on the number of elements to be generated during mesh refinement, as compared to that without using this variable. A detailed description and example is provided under the manual section under *CONTROL_ADAPTIVE_CURVE.

The keyword *INCLUDE_TRIM is recommended to be used at all times, either for trimming or for mesh refinement purpose, except in case where to-be-trimmed sheet blank has no stress and strain information (no *INITIAL_STRESS_SHELL, and *INITIAL_STRAIN_SHELL cards present in the sheet blank keyword or dynain file), the keyword *INCLUDE must be used. A check box to indicate that the blank is free of stress and strain information is provided in the "Trimming" process in the eZ-Setup for users to set up a trimming input deck under the circumstance.

2D and 3D trimming of solid elements and laminates:

Trimming curve preparation, file inclusion, etc.

The requirement for trimming curves definition of solids in case of 3D trimming is different from that of shell trimming. The trim curve should be created based on solid element normal. If trim curve is created closer to the top surface, the variable TDIR should be set to 1; if closer to the bottom surface, set to -1, see Figure 15-27.

Normal directions of solid elements can be viewed using *LS-PrePost* starting in *version 4.2* with the menu option of *EleTol* → *Normal* → *Entity Type: Solid* → *By Part*, and set a large *V-Size*. In addition, when defining a trim curve for the 3D trimming of both solids and laminates, the curve should be as close to either the top or bottom side of the part as possible to enable a successful trimming. This is especially true if wrinkles are present in the panels to be trimmed. *LS-PrePost* can be used to project the curves to the part, via menu option: *GeoTol* → *Project* → *Closest proj* → *Project to Element*. Either top or bottom side of the part can be selected as "*Element*" *by part*. The curves may need to be refined with more points before projection, using menu option: *Curves* → *Method: Respace* → *By number*. Sufficient number of points may be entered to capture the sheet metal surface contour.

For solid element trimming, only *INCLUDE_TRIM (not *INCLUDE) is supported to include the dynain file from a previous process (for example, forming) simulation.

2D trimming of solid elements

As of Revision 92088, 2D **(**option **NEW**) trimming in any direction (defined by a vector) of solid elements is available. An illustration of the 2D trim is shown in Figure 15-25. A partial keyword example is provided below, where trim curves trimcurves2d.iges is being used to perform a solid element trimming along a vector defined along the global Z-axis.

```
*KEYWORD
*INCLUDE_TRIM
incoming.dynain
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*PARAMETER_EXPRESSION
...
```

```
*CONTROL_TERMINATION
$   ENDTIM
0.0
*CONTROL_OUTPUT
...
*DATABASE_XXX
...
*PART
 Solid Blank
$#     pid     secid       mid
  &blk1pid  &blk1sec  &blk1mid
*SECTION_Solid
&blk1sec,&elform
*MAT_PIECEWISE_LINEAR_PLASTICITY
...
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$   Trim cards
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*CONTROL_FORMING_TRIMMING
$     PSID
   &blksid
*DEFINE_TRIM_SEED_POINT_COORDINATES
$ NSEED,X1,Y1,Z1,X2,Y2,Z2
1,&seedx,&seedy,&seedz
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*DEFINE_CURVE_TRIM_NEW
$#    tcid    tctype      tflg      tdir     tctol      toln    nseed1    nseed2
         2         2         0         1   0.10000  1.000000         0         0
$# filename
trimcurves2d.iges
*DEFINE_VECTOR
$#     vid        xt        yt        zt        xh        yh        zh       cid
         1     0.000     0.000     0.000     0.000     0.000  1.000000         0
*INTERFACE_SPRINGBACK_LSDYNA
$     PSID
&blksid,&nshv
*END
```

Currently, 2D trimming of solids in some cases may be approximate. The trimming will trim the top and bottom sides of the elements, not crossing over to the other sides. This can be seen, for instance, trimming involving a radius.

3D (normal) trimming of solid elements

As of Revision 93467 3D trimming (option **3D**) of solid elements is available. From the previous input, in case of 3D trimming, the option **NEW** is changed to **3D**, and trim curves trimcurves3d.iges is used. In the example below, the variable TDIR is set to "1" since the trim curve is on the positive side of the element normal (Figure 15-27). Since 3D trimming are along the element normal directions, *DEFINE_VECTOR card is no longer needed.

```
*DEFINE_CURVE_TRIM_3D
$#    tcid    tctype      tflg      tdir     tctol      toln    nseed1    nseed2
         2         2         0         1   0.10000  1.000000         0         0
$# filename
trimcurves3d.iges
```

Again, the projection of trim curves onto either the top or bottom surface of the blank is important to ensure a smooth and successful trimming.

2D and 3D trimming of non-adaptive-meshed sandwiched parts (laminates)

2D and 3D trimming of non-adaptive-meshed laminates are available starting in Revision 92289. Trimming of the laminates can have multiple layers of solid elements, sandwiched by a top and a bottom layer of shell elements. Note that the nodes of shell elements must share the nodes with solid elements at the top and bottom layers. An illustration of the trim is shown in Figure 15-26. The input deck is similar to those used for trimming of solid elements, except the variable ITYP under *CONTROL_FORMING_TRIMMING should be set to "1" to activate the trimming of laminates in both 2D and 3D conditions:

```
*CONTROL_FORMING_TRIMMING
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$     PSID                ITYP
   &blksid                   1
```

Again, in the case of 3D trimming, the projection of trim curves onto either the top or bottom surface of the blank is important to ensure a smooth and successful trim.

2D and 3D trimming of adaptive-meshed sandwiched parts (laminates)

2D and 3D trimming of adaptive-meshed laminates are available starting in Revision 108770. Trimming of the laminates is limited to one core layer of solid elements, sandwiched by a top and a bottom layer of shell elements. Shell elements share the same nodes as the solid elements.

Note elements are refined automatically along the trim curves until no slave nodes would be cut by the trim curves. In addition, the keyword *CONTROL_ADAPTIVE_CURVE must not be used, since it is only applied to shell elements, and would cause error termination otherwise. Furthermore, unlike the mesh refinement along the trim curve during shell element trimming, this trimming requires no additional adaptivity-related keyword inputs. An example of the trimming on the 2005 NUMISHEET Cross Member is shown in Figure 15-28.

Again, in the case of 3D trimming, the projection of trim curves onto either the top or bottom surface of the blank is important to ensure a smooth and successful trim.

Summary – trimming of solids and laminates

In summary, the trimming input files between solids/laminates and shells are different in a few ways. For solids, *SECTION_SOLID is needed in place of *SECTION_SHELL. For laminates, in addition to setting the ITYP = 1 in *CONTROL_FORMING_TRIMMING, both *SECTION_SHELL and *SECTION_SOLID need to be defined. The position of a trim curve for 3D trimming of solids needs to be defined according to Figure 15-27.

Adaptive-meshed sandwiched parts (limit to a core of one layer of solid elements with outer layers of shell elements) can be 2D or 3D trimmed with adaptive mesh refinement along the trim curves.

Non-adaptive-meshed sandwiched parts (a core of multiple layers of solid elements with outer layers of shell elements) can be 2D and 3D trimmed.

In all trimming of solids and laminates, only dynain file is written out (no d3plot files will be output), and finally, *INCLUDE_TRIM is to be used.

In the case of 3D trimming, the projection of trim curves onto either the top or bottom surface of the blank is important to ensure a smooth and successful trim.

2D trimming of thick shell elements (TSHELL):

2D trimming of TSHELL is supported starting from Revision 107957. Note by definition, TSHELL has only one layer of solid elements, and is defined by keyword *SECTION_TSHELL. Note also *INCLUDE_TRIM (not *INCLUDE) must be used to include the dynain file to be trimmed.

Input deck for 2D trimming of TSHELL is similar to what is used for trimming of solid elements.

2D and 3D trimming of double-attached solids and laminates:

These features are available starting in Revision 110140. Both seed point coordinates can be specified in *DEFINE_TRIM_SEED_POINT_COORDINATES to define a seed coordinate for each part, as shown below:

```
*DEFINE_TRIM_SEED_POINT_COORDINATES
$    NSEED        X1          Y1          Z1          X2          Y2          Z2

         2  -184.565     84.755     78.392   -1038.41    119.154     78.375
```

In Figure 15-29, a 2D double-trimming example on a sandwiched part is shown using the 2005 NUMISHEET cross member. Two trim curves, two seed nodes are defined for each to be trimmed portion. The coordinates for seed node #1 is (-184.565, 84.755, 78.392) and is (-1038.41, 119.154, 78.375) for seed node #2. Trimmed results are satisfactory.

Revision information:

Revision information are as follows:

1. Revision 54608 and 52312: negative seed node option is available for the options **3D** and **NEW**, respectively.

2. Revision 65630: Use of TCTOL as a distance for mesh refinement (when used together with *CONTROL_ADAPTIVE_CURVE) is available.

3. Revision 68643: trimming is enabled for those trim lines going beyond the part boundary.

4. Revision 92088: 2D **(**option **NEW**) trimming of solid elements.

5. Revision 92289: 2D and 3-D (option **3D**) trimming of laminates.

6. Revision 93467: 3D trimming of solid elements.

7. Revision 101964: TDIR definition activated for 3D trimming of top and bottom surfaces of solid elements and laminates.

8. Revision 107957: 2D trimming of TSHELL.

9. Revision 109047: 2D and 3D trimming of adaptive-meshed sandwiched part.

10. Revision 110140: 2D and 3D trimming of solids and laminates of double-attached parts.

11. Latter Revisions may incorporate more improvements and are suggested to be used for trimming.

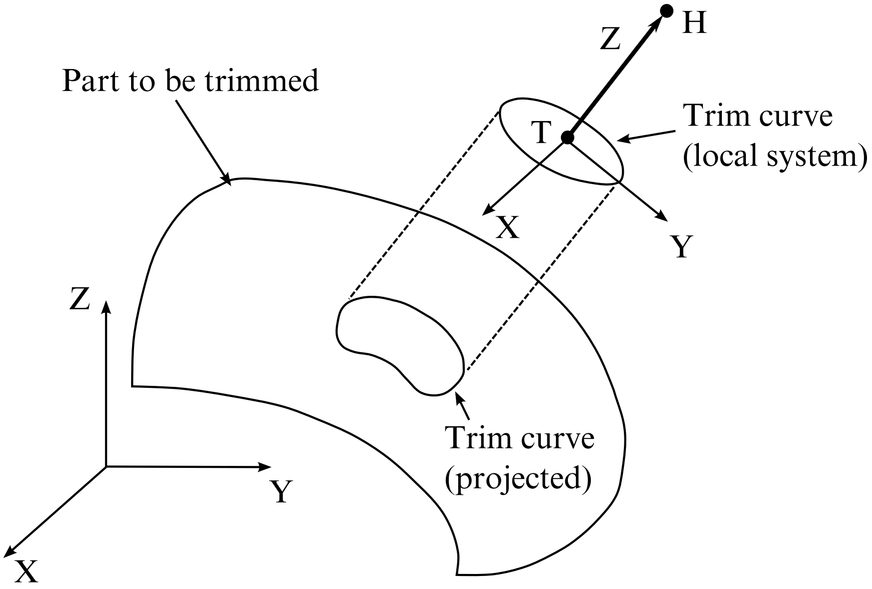


Figure 15-18. Trimming Orientation Vector. The tail (**T**) and head (**H**) points define a local coordinate system (x,y,z). The global coordinate system is named (X,Y,Z). The local x-direction is constructed in the Xz plane. If X and z nearly coincide (|X • z| > 0.95), then the local x-direction is instead constructed in the Yz plane. Trim curve data is input in the x-y plane, and projected in the z-direction onto the deformed mesh to obtain the trim line.

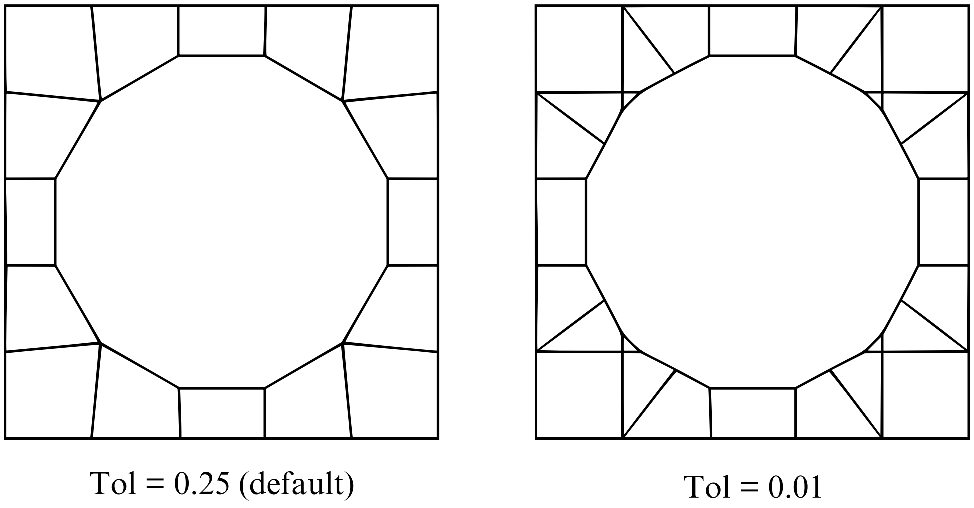


Figure 15-19. Trimming Tolerance. The tolerance limits the size of the small elements generated during trimming. The default tolerance (left) produces large elements. Using a tolerance of 0.01 (right) allows smaller elements, and more detail in the trim line.

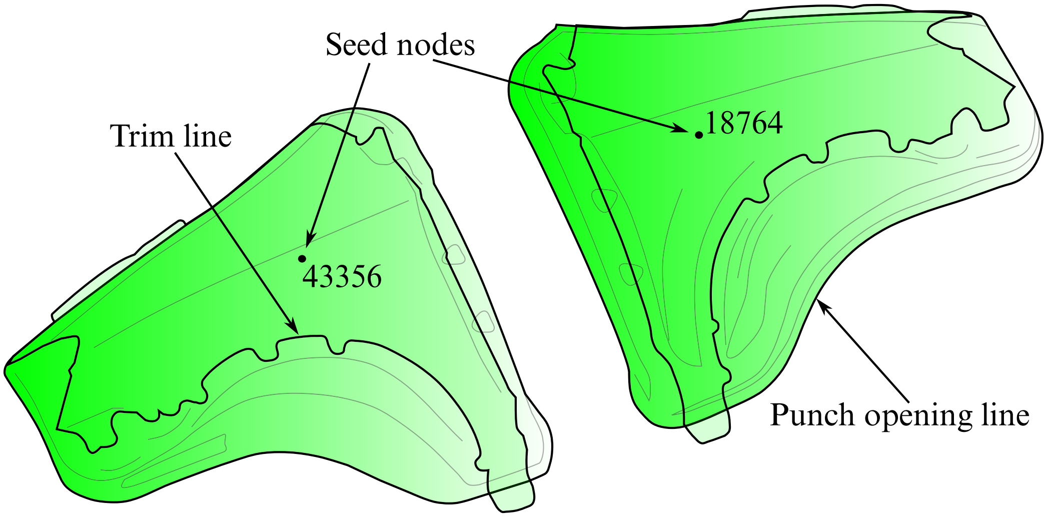


Figure 15-20. Trimming of a double-attached part (NUMISHEET2002 Fender Outer).

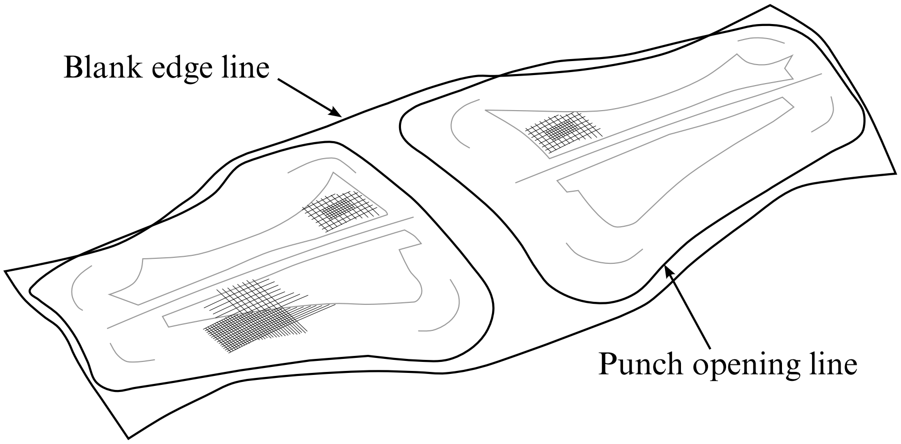


Figure 15-21. The fender outer (draw complete) in wireframe mode.

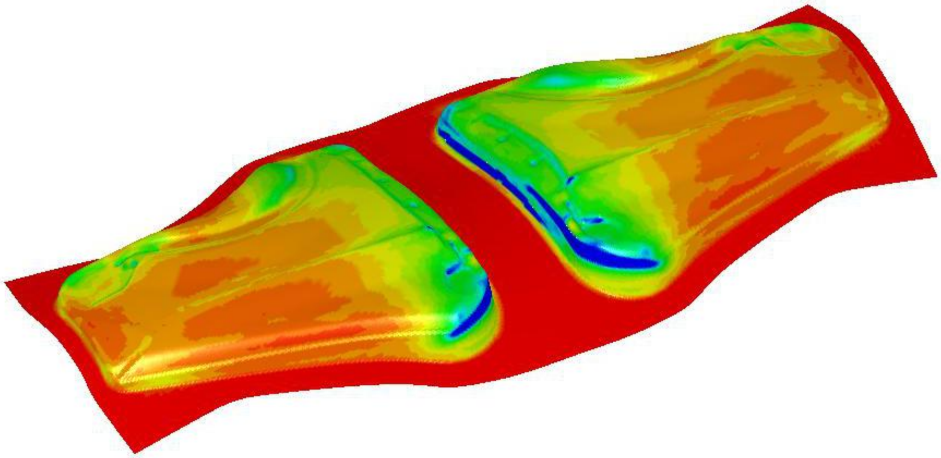


Figure 15-22. The fender outer - thickness/thinning plot on the drawn panel.

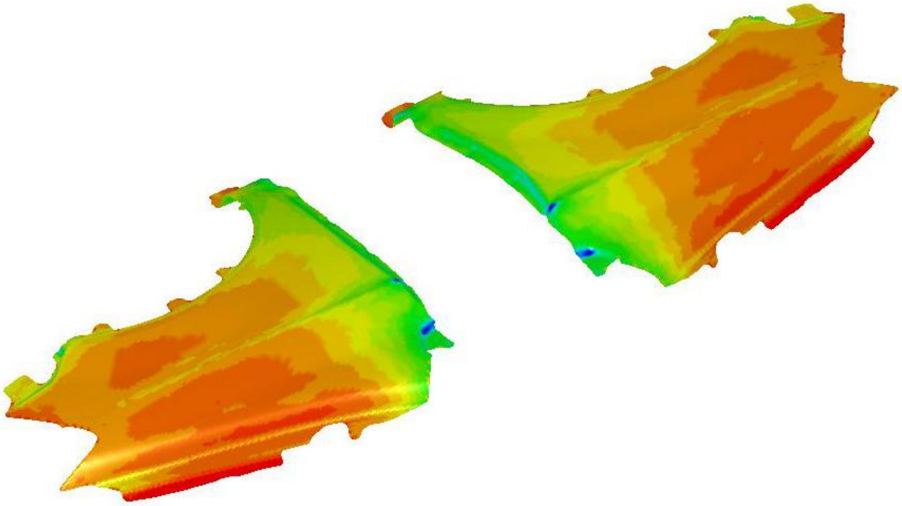


Figure 15-23. The fender outer trim complete using the NSEED1/NSEED2 feature.

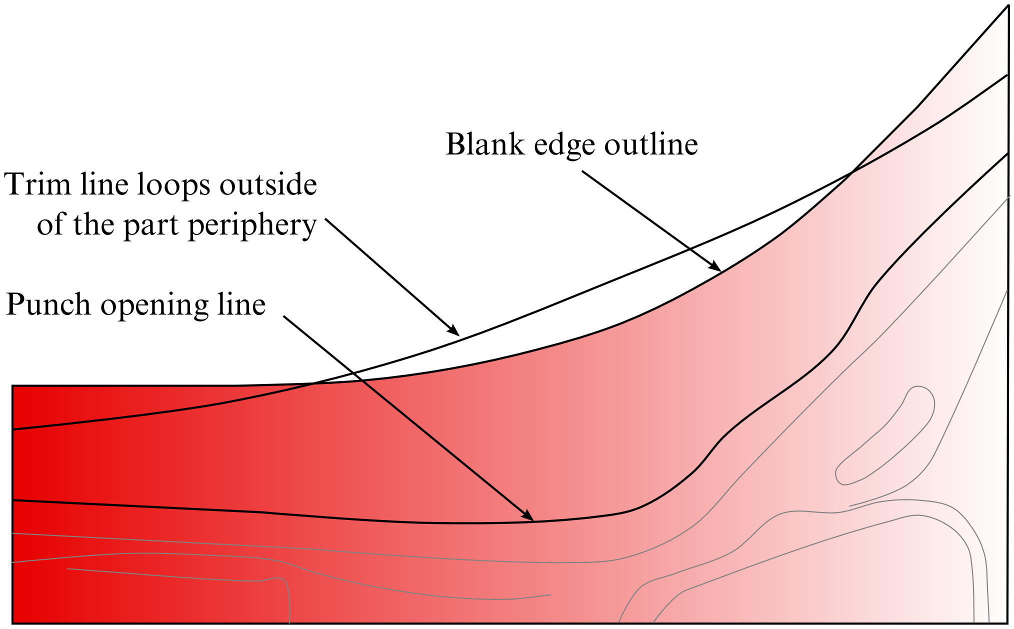


Figure 15-24. Revision 68643 deals with trim curves going beyond part boundary.

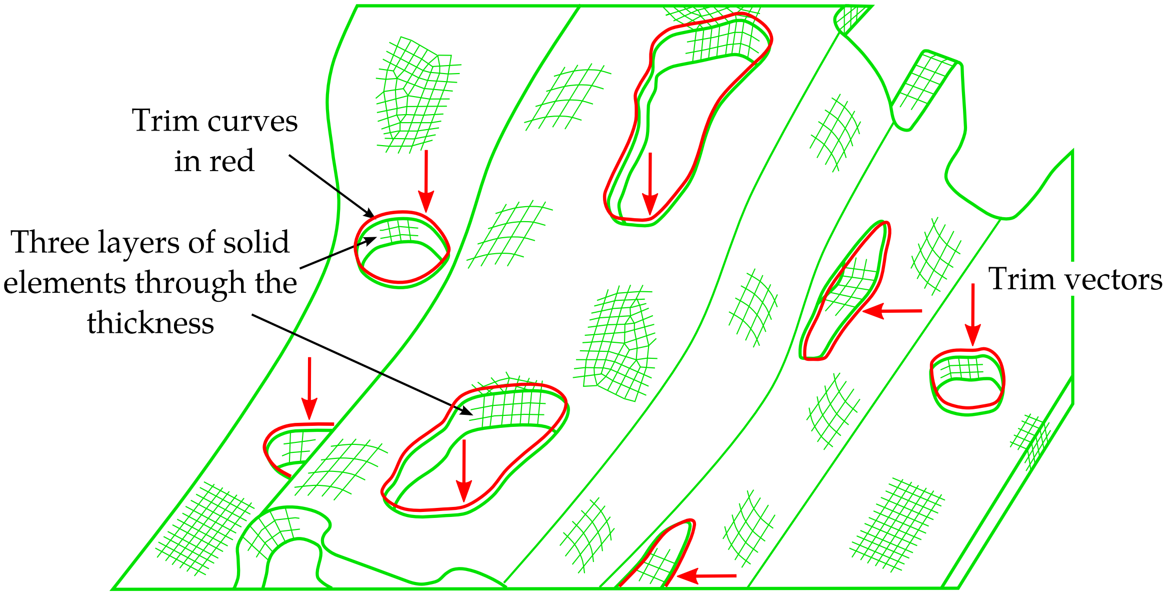


Figure 15-25. 2-D trimming of solids using *DEFINE_CURVE_TRIM_NEW

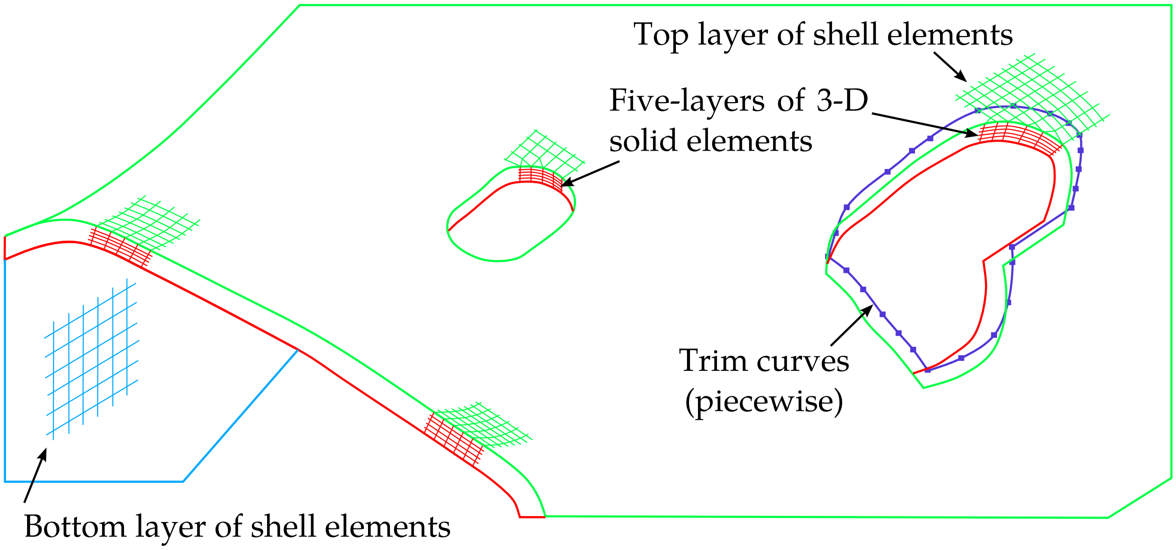
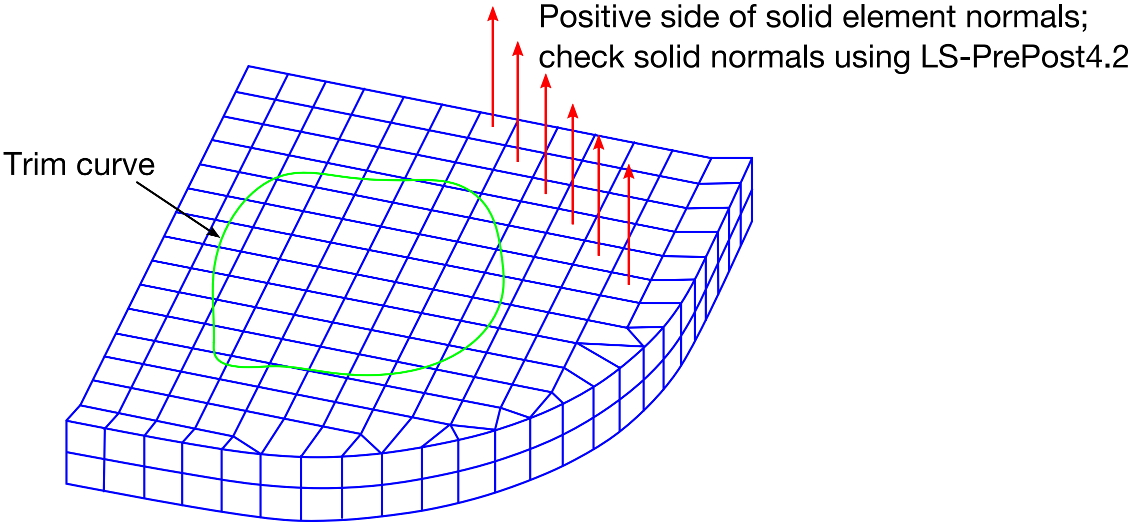

Top layer of shell elements
Five-layers of 3-D solid elements
Trim curves (piecewise)
Bottom layer of shell elements

Figure 15-26. 3-D trimming of laminates (a core of multiple-layers of solid elements with top and bottom layers of shell elements) using *DEFINE_CURVE_TRIM_3D. Note that shell elements must share the same nodes with the solid elements at the top and bottom layer.


Positive side of solid element normals; check solid normals using LS-PrePost4.2
Trim curve

All solid element normals must be consistent. If trim curve is close to the positive normal side, set TDIR=1; otherwise set TDIR=-1. Respacing the curve with more points, project the respaced curve to the top or bottom solid surface may help the trimming.

Figure 15-27. Define trim curve for 3D trimming of solid and laminates.

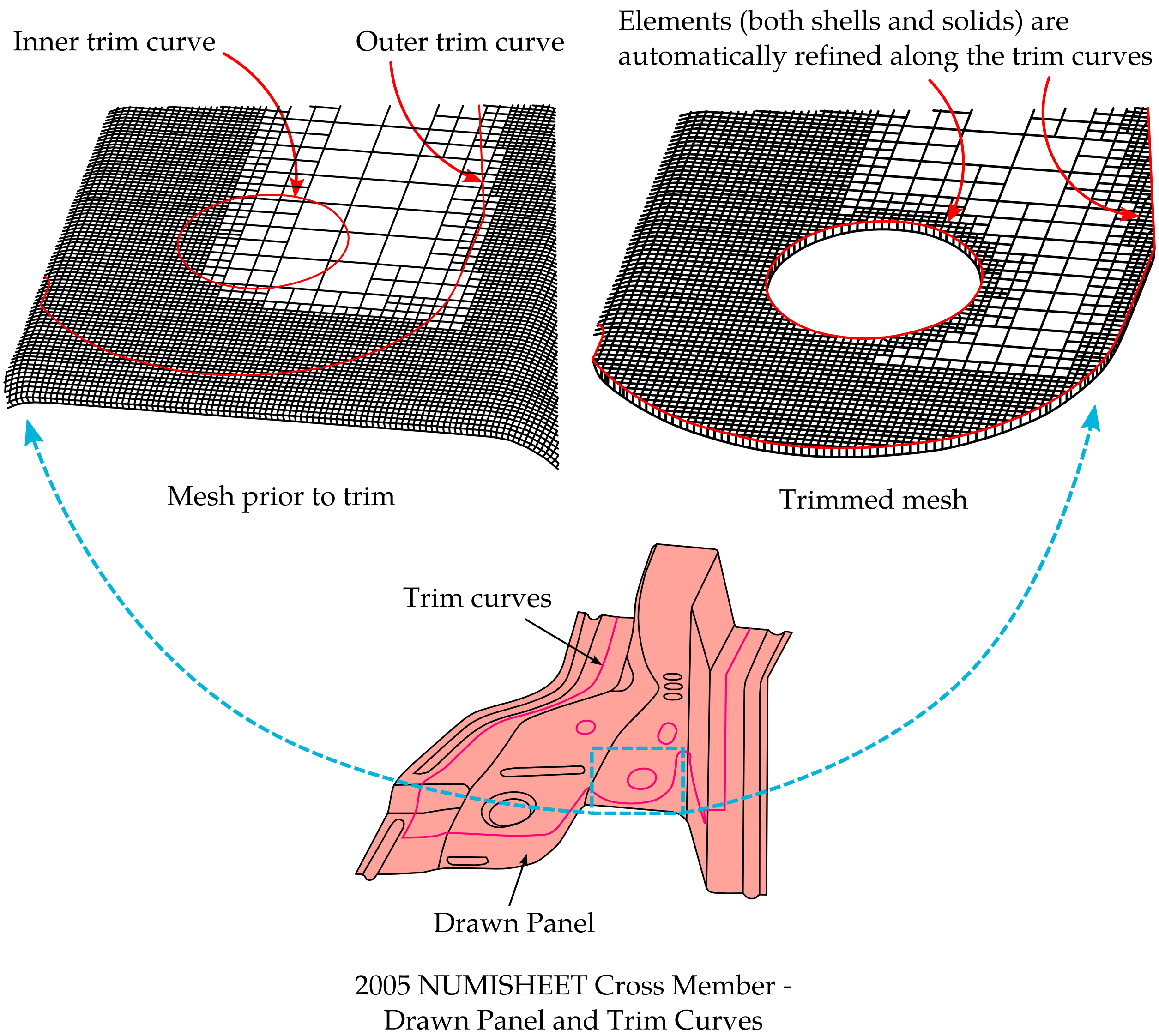

Inner trim curve
Outer trim curve
Elements (both shells and solids) are automatically refined along the trim curves
Mesh prior to trim
Trimmed mesh
Trim curves
Drawn Panel
2005 NUMISHEET Cross Member -
Drawn Panel and Trim Curves

Figure 15-28. Trimming of an adaptive mesh on a sandwiched part. Meshes are automatically refined along the trim curves. Note that only one layer of solid element as a core is allowed for adaptive-meshed sandwich parts. Also top and bottom layer of shells must share the same nodes as the solid elements. **The keyword *CONTROL_ADAPTIVE_CURVE must not be used.**

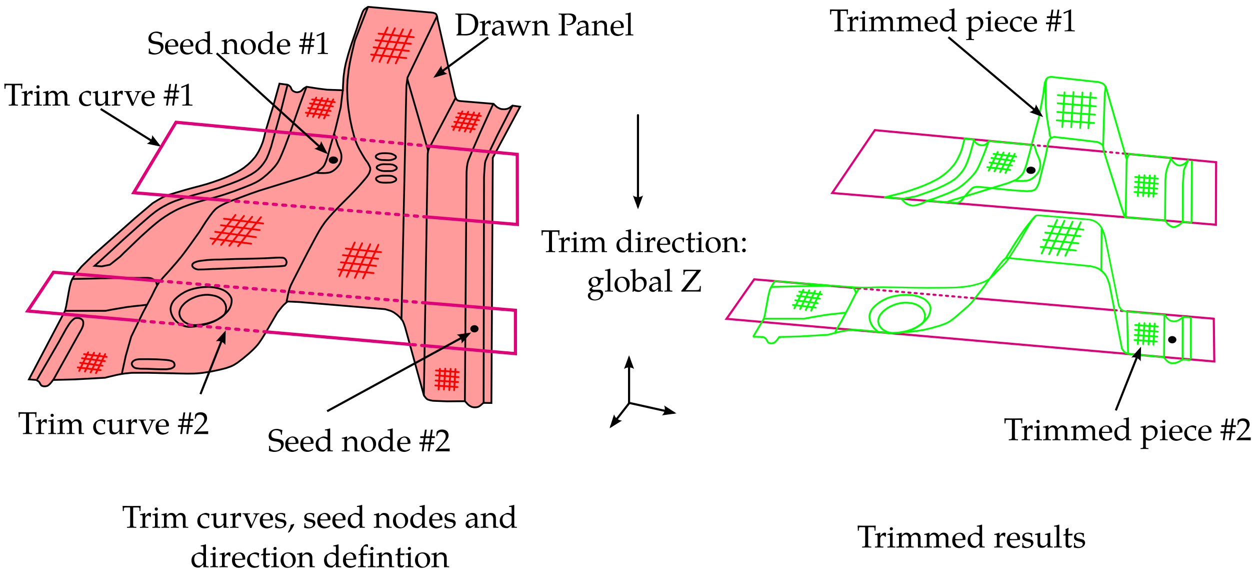


Figure 15-29. An example of 2D trimming of sandwiched part (laminates) from
2005 NUMISHEET benchmark - cross member.

DEFINE_DEATH_TIMES_*OPTION

Available options include:

NODES

SET

RIGID

Purpose: To dynamically define the death times for *BOUNDARY_PRESCRIBED_MOTION based on the locations of nodes and rigid bodies. Once a node or rigid body moves past a plane or a geometric entity, the death time is set to the current time. The input in this section continues until the next '*' card is detected.

Card 1	1	2	3	4	5	6	7	8
Variable	GEO	N1	N2	N3				
Type	I	I	I	I				
Default		0	0	0				

Card 2	1	2	3	4	5	6	7	8
Variable	X_T	Y_T	Z_T	X_H	Y_H	Z_H	R	FLAG
Type	F	F	F	F	F	F	F	
Default								1

ID Cards. Set the list of nodes and rigid bodies affected by this keyword. This input terminates at the next keyword ("*") card.

Card 3	1	2	3	4	5	6	7	8
Variable	NSID1	NSID2	NSID3	NSID4	NSID5	NSID6	NSID7	NSID8
Type	I	I	I	I	I	I	I	I

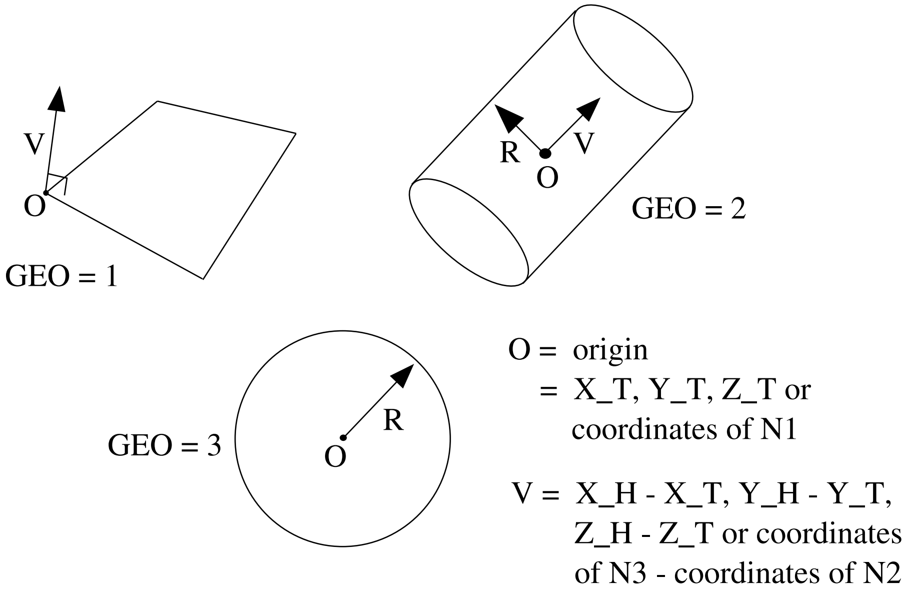


Figure 15-30. Geometry types.

VARIABLE	DESCRIPTION
GEO	Geometric entity type. = 1 plane, = 2 infinite cylinder, = 3 sphere
N1	Node defining the origin of the geometric entity (optional).
N2	Node defining the tail of the orientation vector (optional).
N3	Node defining the head of the orientation vector (optional).
X_T	X coordinate of the origin of the geometric entity and the tail of the orientation vector.
Y_T	Y coordinate of the origin of the geometric entity and the tail of the orientation vector.
Z_T	Z coordinate of the origin of the geometric entity and the tail of the orientation vector.
X_H	X coordinate of the head of the orientation vector.
Y_H	Y coordinate of the head of the orientation vector.
Z_H	Z coordinate of the head of the orientation vector.
R	Radius of cylinder or sphere.

VARIABLE	DESCRIPTION
FLAG	+1 for killing motion when the node is outside of the geometric entity or on the positive side of the plane as defined by the normal direction, or -1 for the inside.
NSIDi	i-th node, node set, or rigid body

Remarks:

1. Either N1 or X_T, Y_T, and Z_T should be specified, but not both.

2. Either N2 and N3 or X_H, Y_H, and Z_H should be specified, but not both. If N2 and N3. Specifying N2 and N3 is equivalent of setting the head of the vector equal to the tail of the vector (X_T, Y_T, and Z_T) plus the vector from N2 to N3.

*DEFINE_DE_ACTIVE_REGION

Purpose: To define an interested region for Discrete Elements (DE) for high efficiency collision pair searching. Any DE leaving this domain will not be considered in the future DE searching and also disabled in the contact algorithm.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	TYPE	Xm	Ym	Zm			
Type	I	I	F	F	F			
Default	none	0	0.	0.	0.			

VARIABLE	DESCRIPTION
ID	Set ID/Box ID
TYPE	EQ.0: Part set ID EQ.1: Box ID

Xm, Ym, Zm

Factor for region's margin on each direction based on region length.

The static coordinates limits are determined either by part set or box option. To extended those limits to provide a buffer zone, these factors can be used. The margin in each direction is calculated in the following way:

Let X_{max} and X_{min} be the limits in the x direction. Then,

$$\Delta X = X_{\mathrm{max}} - X_{\mathrm{min}}$$

Then the margin is computed from the input as,

$$X_{\mathrm{margin}} = \mathrm{Xm} \times \Delta X$$

Then the corresponding limits for the active region are,

$$X'_{\mathrm{max}} = X_{\mathrm{max}} + X_{\mathrm{margin}}$$

$$X'_{\mathrm{min}} = X_{\mathrm{min}} - X_{\mathrm{margin}}$$

*DEFINE_DE_BOND

Purpose: To define a bond model for discrete element sphere (DES).

Card 1	1	2	3	4	5	6	7	8
Variable	SID	STYPE	BDFORM					
Type	I	I	I					
Default	none	0	1					

VARIABLE	DESCRIPTION
SID	DES nodes
STYPE	EQ.0: DES node set EQ.1: DES node EQ.2: DES part set EQ.3: DES part
BDFORM	Bond formulation: EQ.1: Linear bond formulation.

Card 2 for BDFORM = 1.

Card 2	1	2	3	4	5	6	7	8
Variable	PBN	PBS	PBN_S	PBS_S	SFA	ALPHA		MAXGAP
Type	F	F	F	F	F	F		F
Default	none	none	none	none	1.0	0.0		1.E-4

VARIABLE	DESCRIPTION
PBN	Parallel-bond modulus [Pa]. See Remark 1

VARIABLE **DESCRIPTION**

PBS Parallel-bond stiffness ratio. shear stiffness/normal stiffness. See Remark 2

PBN_S Parallel-bond maximum normal stress. A zero value defines an infinite maximum normal stress.

PBS_S Parallel-bond maximum shear stress. A zero value defines an infinite maximum shear stress.

SFA Bond radius multiplier

ALPHA Numerical damping

MAXGAP Maximum gap between two bonded spheres

GT.0.0: defines the ratio of the smaller radius of two bonded spheres as the maximum gap, i.e. MAXGAP × min(r1,r2)

LT.0.0: absolute value is used as the maximum gap.

Remarks:

1.The normal force is calculated as

$$\Delta f_{\mathrm{n}} = \frac{PBN}{(r_0 + r_1)} \times \mathrm{A} \times \Delta u_n$$

where

$$A = \pi r_{eff}^2$$

$$r_{eff} = min(r_0, r_1) \times SFA$$

2.The shear force is calculated as

$$\Delta f_{\mathrm{s}} = \mathrm{PBS} \times \frac{PBN}{(r_0 + r_1)} \times \mathrm{A} \times \Delta u_s$$

*DEFINE_DE_BY_PART

Purpose: To define control parameters for discrete element sphere by part ID. This card overrides the values set in *CONTROL_DISCRETE_ELEMENT.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	NDAMP	TDAMP	FRICS	FRICR	NORMK	SHEARK	
Type	I	F	F	F	F	F	F	
Default	none	0	0	0	0	0.01	2/7	

Card 2	1	2	3	4	5	6	7	8
Variable	GAMMA	VOL	ANG					
Type	F	F	F					
Default	0	0	0					

Card 3 is optional

Card 3	1	2	3	4	5	6	7	8
Variable	LNORM	LSHEAR		FRICD	DC			
Type	I	I		F	F			
Default	0	0		FRICS	0			

VARIABLE	DESCRIPTION
PID	Part ID of DES nodes
NDAMP	Normal damping coefficient
TDAMP	Tangential damping coefficient

VARIABLE	DESCRIPTION
FRICS	Static coefficient of friction. The frictional coefficient is assumed to be dependent on the relative velocity $v_{rel}$of the two DEM in contact $\mu_c = FRICD + (FRICS - FRICD)e^{-DC\bullet\lvert v_{rel}\rvert}$. EQ.0: 3 DOF NE.0: 6 DOF (consider rotational DOF)
FRICR	Rolling friction coefficient
NORMK	Optional: scale factor of normal spring constant (Default = 0.01)
SHEARK	Optional: ratio between ShearK/NormK (Default = 2/7)
GAMMA	Liquid surface tension
VOL	Volume fraction
ANG	Contact angle
LNORM	Load curve ID of a curve that defines function for normal stiffness with respect to norm penetration ratio
LSHEAR	Load curve ID of a curve that defines function for shear stiffness with respect to norm penetration ratio
FRICD	Dynamic coefficient of friction. By default, FRICD = FRICS. The frictional coefficient is assumed to be dependent on the relative velocity $v_{rel}$of the two DEM in contact $\mu_c = FRICD + (FRICS - FRICD)e^{-DC\bullet\lvert v_{rel}\rvert}$.
DC	Exponential decay coefficient. The frictional coefficient is assumed to be dependent on the relative velocity $v_{rel}$of the two DEM in contact $\mu_c = FRICD + (FRICS - FRICD)e^{-DC\bullet\lvert v_{rel}\rvert}$.

See also *CONTROL_DISCRETE_ELEMENT.

*DEFINE_DE_HBOND

Purpose: To define a heterogeneous bond model for discrete element sphere (DES).

Card 1	1	2	3	4	5	6	7	8
Variable	SID	STYPE	HBDFM	IDIM				
Type	I	I	I	I				
Default	none	0	1	3				

VARIABLE	DESCRIPTION
SID	DES nodes
STYPE	EQ.0: DES node set EQ.1: DES node EQ.2: DES part set EQ.3: DES part
HBDFM	Bond formulation: EQ.1: (Reserved) EQ.2: Nonlinear heterogeneous bond formulation for fracture analysis based on the general material models defined in the material cards. DES elements with different material models can be defined within one bond.
IDIM	Space dimension for DES bonds: EQ.2: for 2D plane strain problems EQ.3: for 3D problems.

Card 2 for HBDFM = 2.

Card	1	2	3	4	5	6	7	8
Variable	PBK_SF	PBS_SF	FRGK	FRGS	BONDR	ALPHA	DMG	FRMDL
Type	F	F	F	F	F	F	F	I
Default	1.0	1.0	none	none	none	0.0	1.0	1

VARIABLE **DESCRIPTION**

PBK_SF Scale factor for volumetric stiffness of the bond.

PBS_SF Scale factor for shear stiffness of the bond.

FRGK Critical fracture energy release rate for volumetric deformation due to the hydrostatic pressure.

Special Cases:

EQ.0: A zero value specifies an infinite energy release rate for unbreakable bonds.

LT.0: A negative value defines the energy release rate under volumetric compression (i.e. positive pressure) and FRGS defined below is used under volumetric expansion (i.e. negative pressure).

FRGS Critical fracture energy release rate for shear deformation.

Special Cases:

EQ.0: A zero value specifies an infinite energy release rate for unbreakable bonds.

FRGK.LT.0: See description for FRGK

BONDR Influence radius of the DES nodes.

ALPHA Numerical damping

VARIABLE **DESCRIPTION**

DMG

Continuous parameter for damage model.

EQ.1.0: The bond breaks if the fracture energy in the bond reaches the critical value. Microdamage is not calculated.

∈ (0.5,1): Microdamage effects being once the fracture energy

reaches DMG × FMG[K,S]. Upon the onset of microdamage, the computed damage ratio will increase (monotonically) as the fracture energy grows. Bond weakening from microdamage is modeled by reducing the bond stiffness in proportion to the damage ratio.

FRMDL

Fracture model:

EQ.1: Fracture energy of shear deformation is calculated based on deviatoric stresses.

EQ.2: Fracture energy of shear deformation is calculated based on deviatoric stresses, *excluding the axial component (along the bond).*

EQ.3,4: Same as 1&2, respectively, but FRGK and FRGS are read as the total failure energy density and will be converted to the corresponding critical fracture energy release rate. The total failure energy density is calculated as the total area under uniaxial tension stress-strain curve.

EQ.5,6: Same as 3&4, respectively, as FRGK and FRGS are read as the total failure energy density but will not be converted. Instead, the failure energy within the bond will be calculated.

Models 1&2 are more suitable for brittle materials, and Models 5&6 are easier for ductile materials. Models 3&4 can be used for moderately ductile fracture accordingly.

This is the default fracture model and applied to all DES parts, even if they have different material models. More fracture models can be defined for different materials by specifying an interface ID (ITFID) in the optional card.

Pre-crack Card. This card is optional.

Optional	1	2	3	4	5	6	7	8
Variable	PRECRK	CKTYPE		ITFID				
Type	I	I		I				
Default	none	0		0				

VARIABLE	DESCRIPTION
PRECRK	Shell set, define 3D surfaces of the pre-crack
CKTYPE	EQ.0: Part set EQ.1: Part
ITFID	ID of the interface *INTERFACE_DE_HBOND, which defines different failure models for the heterogeneous bonds within each part and between two parts respectively.

DEFINE_DE_INJECTION_*{OPTION}

Available options include:

<BLANK>

ELLIPSE

Purpose: This keyword injects discrete element spheres (DES) from specified a region at a flow rate given by a user defined curve. When the option is blank the region from which the DES emanate is assumed rectangular. The elliptical option indicates that the region is to be elliptical.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	SID	XC	YC	ZC	XL	YL	CID
Type	I	I	F	F	F	F	F	I
Default	none	none	0.0	0.0	0.0	0.0	0.0	0

Card 2	1	2	3	4	5	6	7	8
Variable	RMASS	RMIN	RMAX_S	VX	VY	VZ	TBEG	TEND
Type	F	F	F	F	F	F	F	F
Default	none	none	RMIN	0.0	0.0	0.0	0.0	10^{20}

Optional card.

Card 3	1	2	3	4	5	6	7	8
Variable	IFUNC							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
PID	Part ID of new generated DES nodes
SID	Node set ID. Nodes and DES properties are generated automatically during input phase based on the user input and assigned to this SID.
XC, YC, ZC	x, y, z coordinate of the center of injection plane
XL	For rectangular planes XL specifies the planar length along the x-axis in the coordinate system specified by CID. For elliptical planes XL specifies the length of the major axis.
YL	For rectangular planes YL specifies the planar length along the y-axis in the coordinate system specified by CID. For elliptical planes YL specifies the length of the minor axis.
CID	Optional local coordinate system ID, see *DEFINE_COORDINATE_SYSTEM
RMASS	Mass flow rate LT.0: Curve ID
RMIN	Minimum DES radius
RMAX	Maximum DES radius
VX, VY, VZ	Vector components defining the initial velocity of injected DES specified relative the coordinate system defined by CID.
TBEG	Birth time
TEND	Death time
IFUNC	EQ.0: Uniform distribution(Default) EQ.1: Gaussian distribution (Remark 1)

Remarks:

The distribution of particle radius follows Gaussian distribution

$$f(r|r_0,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(r-r_0)^2}{2\sigma^2}}$$

Where the mean radius is given by

$$r_0 = \frac{1}{2}(r_{\max} + r_{\min}),$$

and the standard deviation is

$$\sigma = \frac{1}{2}(r_{\max} - r_{\min})$$

*DEFINE_DE_MASSFLOW_PLANE

Purpose: To measure DES mass flow rate across a defined plane. See also the accompanying keyword *DATABASE_DEMASSFLOW which controls the output frequency.

Card 1	1	2	3	4	5	6	7	8
Variable	SLAVE	MASTER	STYPE	MTYPE				
Type	I	I	I	I				
Default	0	0	0	0				

VARIABLE	DESCRIPTION
SLAVE	Node set ID, node ID, part set ID or part ID defining DES on slave side. STYPE below indicates the ID type specified by SLAVE.
MASTER	Part set ID or part ID defining master surface. MTYPE below indicates the ID type specified by MASTER.
STYPE	Slave type. EQ.0: Slave node set EQ.1: Slave node EQ.2: Slave part set EQ.3: Slave part
MTYPE	Master type. EQ.0: Part set EQ.1: Part

*DEFINE_DE_TO_BEAM_COUPLING

Purpose: To define coupling interface between discrete element sphere (DES) and beam.

Card 1	1	2	3	4	5	6	7	8
Variable	SLAVE	MASTER	STYPE	MTYPE				
Type	I	I	I	I				
Default	0	0	0	0				

Card 2	1	2	3	4	5	6	7	8
Variable	FricS	FricD	DAMP	BSORT				
Type	F	F	F	I				
Default	0	0	0	100				

VARIABLE	DESCRIPTION
SLAVE	DES nodes
MASTER	Shell set
STYPE	EQ.0: Slave node set EQ.1: Slave node EQ.2: Slave part set EQ.3: Slave part
MTYPE	EQ.0: Part set EQ.1: Part
FricS	Friction coefficient
FricD	Rolling friction coefficient

VARIABLE	DESCRIPTION
DAMP	Damping coefficient
BSORT	Number of cycle between bucket sortings. (Default = 100)

*DEFINE_DE_TO_SURFACE_COUPLING

Purpose: To define a non-tied coupling interface between discrete element spheres (DES) and a surface defined by shell part(s) or solid part(s). This coupling is currently not implemented for tshell part(s).

Card 1	1	2	3	4	5	6	7	8
Variable	SLAVE	MASTER	STYPE	MTYPE				
Type	I	I	I	I				
Default	0	0	0	0				

Card 2	1	2	3	4	5	6	7	8
Variable	FricS	FricD	DAMP	BSORT	LCVx	LCVy	LCVz	WEARC
Type	F	F	F	I	I	I	I	F
Default	0	0	0	100	0	0	0	0.

User Defined Wear Parameter Cards. Additional Card for WEARC.LT.0.

Card 3	1	2	3	4	5	6	7	8
Variable	W1	W2	W3	W4	W5	W6	W7	W8
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

Card 4 is optional.

Card 4	1	2	3	4	5	6	7	8
Variable	SFP	SFT				CID_RCF	BT	DT
Type	F	F				I	F	F
Default	1.0	1.0				0	0	1.E20

VARIABLE	DESCRIPTION
SLAVE	Node set ID, node ID, part set ID or part ID defining DES on slave side. STYPE below indicates the ID type specified by SLAVE.
MASTER	Part set ID or part ID defining master surface. MTYPE below indicates the ID type specified by MASTER.
STYPE	EQ.0: Slave node set EQ.1: Slave node EQ.2: Slave part set EQ.3: Slave part
MTYPE	EQ.0: Part set EQ.1: Part
FricS	Friction coefficient
FricD	Rolling friction coefficient
DAMP	Damping coefficient (unitless). If a discrete element sphere impacts a rigid surface with a velocity v_{impact}, the rebound velocity is $v_{\mathrm{rebound}} = (1 - \mathrm{DAMP}) v_{\mathrm{impact}}$

VARIABLE	DESCRIPTION
BSORT	Number of cycle between bucket sortings; Default value is 100. For blast simulation with very high DEM particles velocity, it is suggested to set BSORT = 20 or smaller. LT.0: ABS(BSORT) is the minimum number of cycle between bucket sort. This value can be increased during runtime by tracking the velocity of potential coupling pair. This feature only works with MPP currently.
LVCx	Load curve defines surface velocity in *x* direction
LVCy	Load curve defines surface velocity in *y* direction
LVCz	Load curve defines surface velocity in *z* direction
WEARC	WEARC is the wear coefficient. GT.0: Archard's Wear Law, See Remark 1. EQ.-1: Finnie Wear Law, additional card is required EQ.-100: User defined wear model, additional card is required
W1-W8	WEARC = -1, W1 is yield stress of target material WEARC = -100, user defined wear parameters
SFP	Scale factor on contact stiffness. By default, SFP = 1.0. The contact stiffness is calculated as $K_n = SFP \bullet \begin{cases} K \bullet r \bullet NormK & if\ NormK > 0 \\ \lvert NormK \rvert & if\ NormK < 0 \end{cases}$ where K is bulk modulus, r is discrete element radius, $NormK$ is scale factor of the spring constant defined in *CONTROL_DISCRETE_ELEMENT.
SFT	Scale factor for surface thickness (scales true thickness). This option applies only to contact with shell elements. True thickness is the element thickness of the shell elements.
CID_RCF	Coordinate system ID to output *demrcf* force resultants in a local system.
BT	Birth time.
DT	Death time.

Remarks:

1. **Archard's Wear Law.** If WEARC > 0 then wear on the shell surface is calculated using Archard's wear law

$$\dot{h} = \frac{\mathrm{WEARC} \times f_n \times v_t}{A}$$

where,

h = wear depth

f_n = normal contact force from DE

v_t = tangential sliding velocity of the DE on shell

A = area of contact segment

The wear depth is output to the interface force file.

2. **Finnie's Wear Law.** If WEARC=-1 then wear on the shell surface is calculated

using Finnie's wear law. The model of Finnie relates the rate of wear to the rate of kinetic energy of particle impact on a surface as:

$$\mathrm{Q} = \begin{cases} \dfrac{mv^2}{8p}(\sin 2\alpha - 3\sin^2\alpha) & \mathit{if}\ \ \tan\alpha < \dfrac{1}{3} \\ \dfrac{mv^2}{24p}\cos^2\alpha & \mathit{if}\ \ \tan\alpha > \dfrac{1}{3} \end{cases}$$

where, Q is the volume of the material removed from surface, m is particle mass, α is impact angle and p is the yield stress of the target material, p is read from additional user defined wear parameter card as p = w1.. The wear depth is output to the interface force file.

3. *DATABASE_BINARY_DEMFOR controls the output interval of the coupling forces to the DEM interface force file. This interface force file is activated by the command line option "dem=", for example,

```
lsdyna i=inputfilename.k … dem=interfaceforce_filename
```

The DEM interface force file can be read into LS-PrePost for plotting of coupling pressure and forces on the master segments.

4. *DATABASE_RCFORC controls the output interval of the coupling forces to the ASCII demrcf file. This output file is analogous to the rcforc file for *CONTACT.

*DEFINE_DE_TO_SURFACE_TIED

Purpose: To define a tied-with-failure coupling interface between discrete element spheres (DES) and a surface defined by shell part(s) or solid part(s). This coupling is currently not implemented for tshell part(s).

Card 1	1	2	3	4	5	6	7	8
Variable	SLAVE	MASTER	STYPE	MTYPE				
Type	I	I	I	I				
Default	0	0	0	0				

Card 2	1	2	3	4	5	6	7	8
Variable	NFLF	SFLF	NEN	MES	LCID	NSORT		
Type	F	F	F	F	I	I		
Default	Required	Required	2.	2.	0	100		

VARIABLE	DESCRIPTION
SLAVE	Node set ID, node ID, part set ID or part ID defining DES on slave side. STYPE below indicates the ID type specified by SLAVE.
MASTER	Part set ID or part ID defining master surface. MTYPE below indicates the ID type specified by MASTER.
STYPE	EQ.0: Slave node set EQ.1: Slave node EQ.2: Slave part set EQ.3: Slave part
MTYPE	EQ.0: Part set EQ.1: Part

VARIABLE **DESCRIPTION**

NFLF

Normal failure force. Only tensile failure, i.e., tensile normal forces, will be considered in the failure criterion

SFLF

Shear failure force

NEN

Exponent for normal force

MES

Exponent for shear force. Failure criterion:

$$\left(\frac{|f_n|}{\mathrm{NFLF}}\right)^{\mathrm{NEN}} + \left(\frac{|f_s|}{\mathrm{SFLF}}\right)^{\mathrm{MES}} \geq 1.$$

Failure is assumed if the left side is larger than 1. f_n and f_s are the normal and shear interface force.

LCID

Load curve ID define the time dependency of failure criterion

NSORT

Number of cycle between bucket sort

Remarks:

Both NFLF and SFLF must be defined. If failure in only tension or shear is required then set the other failure force to a large value (10^{10}).

DEFINE_ELEMENT_DEATH_*OPTION

Available options include:

SOLID

SOLID_SET

BEAM

BEAM_SET

SHELL

SHELL_SET

THICK_SHELL

THICK_SHELL_SET

Purpose: To define a discrete time or box to delete an element or element set during the simulation. This keyword is only for deformable elements, not rigid body elements.

Card 1	1	2	3	4	5	6	7	8
Variable	EID/SID	TIME	BOXID	INOUT	IDGRP	CID		
Type	I	I	I	I	I	I		
Default		0	0	0	0	0		

VARIABLE	DESCRIPTION
EID/SID	Element ID or element set ID.
TIME	Deletion time for elimination of the element or element set. If BOXID is nonzero, a TIME value of zero is reset to 1.0E+16.
BOXID	Element inside or outside of defined box are deleted depending on the value of INOUT.
INOUT	Location of deleted element: EQ.0: Elements inside box are deleted EQ.1: Element outside of box are deleted

VARIABLE	DESCRIPTION
IDGRP	Group ID. All elements sharing the same positive value of IDGRP are considered to be in the same group. All elements in a group will be simultaneously deleted one cycle after any single element in the group fails. There is no requirement that each *DEFINE_ELEMENT_DEATH command have a unique IDGRP. In other words, elements in a single group can come from multiple *DEFINE_ELEMENT_DEATH commands. Elements in which IDGRP = 0 are not assigned to a group and thus deletion of one element does not enforce deletion of the other elements.
CID	Coordinate ID for transforming box BOXID. If CID is not specified, the box is in the global coordinate system. The box rotates and translates with the coordinate system only if the coordinate system is flagged for an update every time step.

*DEFINE_ELEMENT_GENERALIZED_SHELL

Purpose: Define a general 3D shell formulation to be used in combination with *ELEMENT_GENERALIZED_SHELL. The objective of this feature is to allow the rapid prototyping of new shell element formulations by adding them through the keyword input file.

All necessary information, like the values of the shape functions and their derivatives at various locations (at the integration points and at the nodal points) have to be defined via this keyword. An example for a 9-noded generalized shell element with 4 integration points in the plane is given in Figure 15-31 to illustrate the procedure. The element formulation ID (called ELFORM) used in this keyword needs to be greater or equal than 1000 and will be referenced through *SECTION_SHELL (see Figure 15-3 in *ELEMENT_GENERALIZED_SHELL).

Card 1	1	2	3	4	5	6	7	8
Variable	ELFORM	NIPP	NMNP	IMASS	FORM			
Type	I	I	I	I	I			
Default	none	none	none	none	none			

Weights and Shape Function Values/Derivatives at Gauss Points:

These cards are read according to the following pseudo code:

```
for i  = 1 to NIPP {
        read cardA1(i)
        for k = 1 to NMNP {
                read cardA2(i,k)
        }
} // comment: Read in NIPP × (1 + NMNP) cards
```

Weight Cards. Provide weight for integration point *i*.

(Card A1)$_i$	1	2	3	4	5	6	7	8
Variable	WI							
Type	F							

Integration Point Shape Function Value/Derivatives Cards. Provide the value of the k^{th} shape function and its derivative at the i^{th} integration point.

(Card A2)$_{ik}$	1	2	3	4	5	6	7	8
Variable	NKI		DNKIDR		DNKIDS			
Type	F		F		F			

For FORM = 0 or FORM = 1, Shape Function Derivatives at Nodes:

These cards are read according to the following pseudo code:

```
for l  = 1 to NMNP {
        for k = 1 to NMNP {
                read cardB(l,k)
        }
} // comment: Read in NMNP × NMNP cards
```

Nodal Shape Function Derivative Cards. The value of the k^{th} shape function's derivative at the l^{th} nodal point.

(Card B)$_{lk}$	1	2	3	4	5	6	7	8
Variable	DNKLDR		DNKLDS					
Type	F		F					

For FORM = 2 or FORM = 3, Shape Function 2nd derivative at Gauss Points:

> NOTE: For FORM = 2 and FORM = 3 it is assumed that the shape functions are at least C1 continuous (having a continuous derivative).

The cards for this method are read according to the following pseudo code:

```
for i  = 1 to NIPP {
        for k = 1 to NMNP {
                read cardB(I,k)
        }
} // comment: Read in NGP × NMNP cards
```

Nodal Shape Function Second Derivative Cards. The value of the k^{th} shape function's second derivative at the i^{th} integration point.

(Card B)$_{ik}$	1	2	3	4	5	6	7	8
Variable	D2NKIDR2		D2NKIDRDS		D2NKIDS2			
Type	F		F		F			

VARIABLE	DESCRIPTION
ELFORM	Element Formulation ID referenced via *SECTION_SHELL to connect *ELEMENT_GENERALIZED_SHELL with the appropriate shell formulation. The chosen number needs to be greater or equal than 1000.
NIPP	Number of in-plane integration points.
NMNP	Number of nodes for this element formulation.
IMASS	Option for lumping of mass matrix: EQ.0: row sum EQ.1: diagonal weighting.
FORM	Shell formulation to be used EQ.0: shear deformable shell theory with rotational DOFs (shell normal evaluated at the nodes) EQ.1: shear deformable shell theory without rotational DOFs (shell normal evaluated at the nodes) EQ.2: thin shell theory without rotational DOFs (shell normal evaluated at the integration points) EQ.3: thin shell theory with rotational DOFs (shell normal evaluated at the integration points)
WI	Integration weight at integration point *i*.

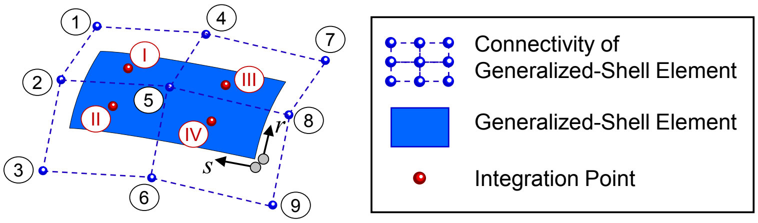
VARIABLE	DESCRIPTION
NKI	Value of the shape function N_k evaluated at integration point *i*.
DNKIDR	Value of the derivative of the shape function N_k with respect to the local coordinate r at the integration point i $\left(\frac{\partial N_k^i}{\partial r}\right)$.
DNKIDS	Value of the derivative of the shape function N_k with respect to the local coordinate s at the integration point i $\left(\frac{\partial N_k^i}{\partial s}\right)$.
DNKLDR	Value of the derivative of the shape function N_k with respect to the local coordinate r at the nodal point l $\left(\frac{\partial N_k^l}{\partial r}\right)$.
DNKLDS	Value of the derivative of the shape function N_k with respect to the local coordinate s at the nodal point l $\left(\frac{\partial N_k^l}{\partial s}\right)$.
D2NKIDR2	Value of the second derivative of the shape function N_k with respect to the local coordinate r at the integration point i $\left(\frac{\partial^2 N_k^i}{\partial r^2}\right)$.
D2NKIDRDS	Value of the second derivative of the shape function N_k with respect to the local coordinates r and s at the integration point i $\left(\frac{\partial^2 N_k^i}{\partial r \partial s}\right)$.
D2NKIDS2	Value of the second derivative of the shape function N_k with respect to the local coordinate s at the integration point i $\left(\frac{\partial^2 N_k^i}{\partial s^2}\right)$.

Remarks:

1. For post-processing and the treatment of contact boundary conditions, the use of interpolation shell elements (see *ELEMENT_INTERPOLATION_SHELL and *CONSTRAINED_NODE_INTERPOLATION) is necessary.

2. The order of how to put in the data for the NMNP nodal points has to be in correlation with the definition of the connectivity of the element in *ELEMENT_GENERALIZED_SHELL.

Example:


1
4
7
2
5
8
3
6
9
I
II
III
IV
r
s
Connectivity of Generalized-Shell Element
Generalized-Shell Element
Integration Point

```
                *DEFINE_ELEMENT_GENERALIZED_SHELL
                $#   elform       nipp       nmnp      imass       form
                      1001           4          9          0          1
                $# integration point 1 (i=1)
                $#                 wi
W1               1.3778659577546E-04
                $#                nki              dnkidr              dnkids
k=1              1.7098997698601E-01 3.3723996630918E+00 2.4666694616947E+00
k=2-9           ...
                $# integration point 2 (i=2)                                    Block A
W2               2.2045855324077E-04
NMNP             5.4296436772101E-02 1.9003752917745E+00 7.8327025592051E+00
Lines           ...
1 (W3)+         $# integration point 3 (i=3)
NMNP Lines      ...
1 (W4)+         $# integration point 4 (i=4)
NMNP Lines      ...
                $# node 1 (l=1)
                $#             dnkldr              dnklds
k=1              4.8275862102259E+00 3.5310344763662E+01
k=2-9           ...
                $# node 2 (l=2)                                                 Block B
NMNP             2.4137931051130E+00 8.8275861909156E+00
Lines           ...
                [...]
NMNP            $# node 9 (l=9)
Lines           ...
```

Figure 15-31. Example of a generalized shell formulation with *DEFINE_ELEMENT_GENERALIZED SHELL.

*DEFINE_ELEMENT_GENERALIZED_SOLID

Purpose: Define a general 3D solid formulation to be used in combination with *ELEMENT_GENERALIZED_SOLID. The objective of this feature is to allow the rapid prototyping of new solid element formulations by adding them through the keyword input file.

All necessary information, like the values of the shape functions and their derivatives at all integration points have to be defined via this keyword. An example for a 18-noded generalized solid element with 8 integration points is given in Figure 15-31 to illustrate the procedure. The element formulation ID (called ELFORM) used in this keyword needs to be greater or equal than 1000 and will be referenced through *SECTION_SOLID (see Figure 17-30 in *ELEMENT_GENERALIZED_SOLID).

Card 1	1	2	3	4	5	6	7	8
Variable	ELFORM	NIP	NMNP	IMASS				
Type	I	I	I	I				
Default	none	none	none	none				

These cards are read according to the following pseudo code:

```
for i  = 1 to NIP {
        read cardA1(i)
        for k = 1 to NMNP {
                read cardA2(i,k)
        }
} // comment: Read in NIP × (1 + NMNP) cards
```

Weight Cards. Provide weight for integration point *i*.

(Card A1)$_i$	1	2	3	4	5	6	7	8
Variable	WI							
Type	F							
Default	none							

Integration Point Shape Function Value/Derivatives Cards. Provide the value of the k^{th} shape function and its derivative at the i^{th} integration point.

(Card A2)$_{ik}$	1	2	3	4	5	6	7	8
Variable	NKI		DNKIDR		DNKIDS		DNKIDT	
Type	F		F		F		F	
Default	none		none		none		none	

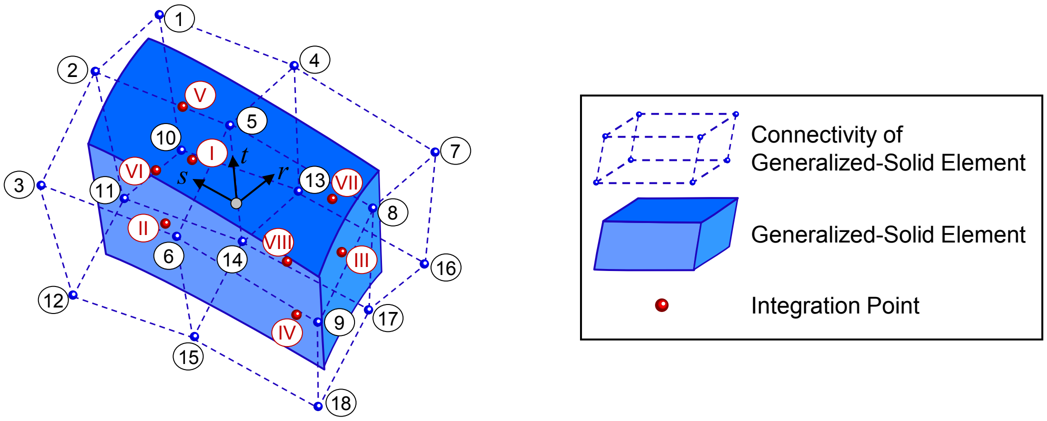
VARIABLE	DESCRIPTION
ELFORM	Element Formulation ID referenced via *SECTION_SOLID to connect *ELEMENT_GENERALIZED_SOLID with the appropriate solid formulation. The chosen number needs to be greater or equal than 1000.
NIP	Number of integration points.
NMNP	Number of nodes for this element formulation.
IMASS	Option for lumping of mass matrix: EQ.0: row sum EQ.1: diagonal weighting.
WI	Integration weight at integration point *i*.
NKI	Value of the shape function N_k evaluated at integration point *i*.
DNKIDR	Value of the derivative of the shape function N_k with respect to the local coordinate r at the integration point $i \left(\frac{\partial N_k^i}{\partial r}\right)$.
DNKIDS	Value of the derivative of the shape function N_k with respect to the local coordinate s at the integration point $i \left(\frac{\partial N_k^i}{\partial s}\right)$.
DNKIDT	Value of the derivative of the shape function N_k with respect to the local coordinate t at the integration point $i \left(\frac{\partial N_k^i}{\partial t}\right)$.

Remarks:

1. For post-processing the use of interpolation solid elements (see *ELEMENT_INTERPOLATION_SOLID and *CONSTRAINED_NODE_INTERPOLATION) is necessary.

2. The order of how to put in the data for the NMNP nodal points has to be in correlation with the definition of the connectivity of the element in *ELEMENT_GENERALIZED_SOLID.

Example:


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
I
II
III
IV
V
VI
VII
VIII
r
s
t
Connectivity of Generalized-Solid Element
Generalized-Solid Element
Integration Point

```
              *DEFINE_ELEMENT_GENERALIZED_SOLID
              $#   elform         nip       nmnp      imass
                     1001           8         18          0
              $# integration point 1  (i=1)
              $#                 wi
W1             1.3778659577546E-04
              $#                nki                dnkidr               dnkids               dnkidt
k=1            1.7098997698601E-01 3.3723996630918E+00 2.4666694616947E+00 1.5327451653258E+00
k=2,18        ...
              $# integration point 2  (i=2)
W2             2.2045855324077E-04
NMNP           5.4296436772101E-02 1.9003752917745E+00 7.8327025592051E+00 3.258715871621E+00
Lines         ...
              [...]                                                                  Block A
              $# integration point 8  (i=8)
W8             3.8574962585875E-04
NMNP           2.6578426581235E-01 1.6258741125438E+00 2.9876495873627E+00 5.403982758392E+00
Lines          ...
```

Figure 15-32. Example of a generalized solid formulation with *DEFINE_ELEMENT_GENERALIZED_SOLID

*DEFINE_FABRIC_ASSEMBLIES

Purpose: Define lists of part sets to properly treat fabric bending between parts.

Define as many cards as needed for the assemblies, using at most 8 part sets per card. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	SPID1	SPID2	SPID3	SPID4	SPID5	SPID6	SPID7	SPID8
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

VARIABLE	DESCRIPTION
SPID*n*	Part set ID that comprises an assembly.

Remarks:

The materials *MAT_FABRIC and *MAT_FABRIC_MAP are equipped with an optional coating feature to model the fabric's bending resistance. See the related parameters ECOAT, SCOAT and TCOAT on these material model manual entries.

The default behavior for these coatings, *which this keyword changes*, excludes T-interesections, and, furthermore requires that all fabric elements must have a consistently oriented normal vector. In Figure 15-33, the left connection of fabric elements is permitted by the default functionality while the right one is not. However, with using this keyword the proper bending treatment for the right connectivity can be activated by adding the following input to the deck

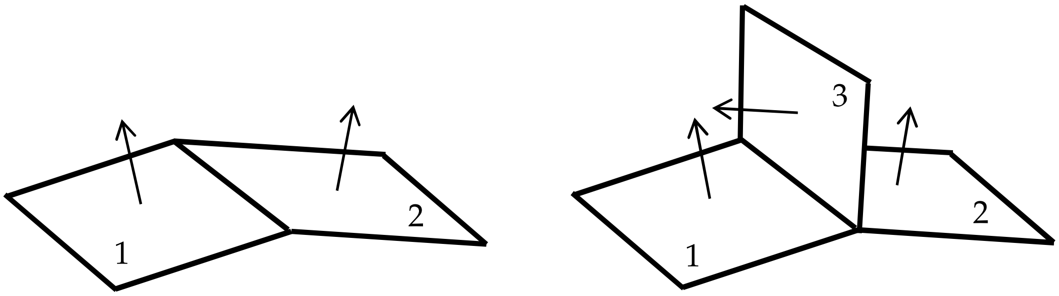


Figure 15-33. Bending in fabric, intended use of *DEFINE_FABRIC_ASSEMBLIES. The numbers indicate the parts the elements belong to.

```
*SET_PART_LIST
1
1,2
*SET_PART_LIST
2
3
*DEFINE_FABRIC_ASSEMBLIES
1,2
```

which decouples part 3 from the other two parts in terms of bending, thus creating a moment free hinge along the edge between part sets 1 and 2. Bending between parts 1 and 2 is unaffected since these are contained in the same fabric assembly.

For several instances of this keyword in an input deck, the list of assemblies is appended. If assemblies are defined and there happens to be fabric parts that do not belong to any of the specified assemblies, then these parts are collected in a separate unlisted assembly. The restriction on consistent normal vectors and on having no T-intersections applies to all elements within an assembly.

*DEFINE_FILTER

Purpose: Define a general purpose filter, currently used by this option:

SENSOR_SWITCH

The input in this section consists of two cards:

Card 1	1	2	3	4	5	6	7	8
Variable	ID	Title						
Type	I	A70						

Card 2	1	2	3	4	5	6	7	8
Type	Type	Data1	Data2	Data3	Data4	Data5	Data6	Data7
Type	A10							

VARIABLE	DESCRIPTION
ID	Identification number.
Title	Title for this filter.
Type	One of the 3 currently defined filter types: DISCRETE, CONTINUOUS, or CHAIN
Data1-7	Filter type specific data, which determines what the filter does.

Filter Types:

FILTER	DESCRIPTION
DISCRETE	The discrete filter operates on a fixed number of values of the input data. The first data field is an A10 character field, which gives the type of operation the filter performs: MIN, MAX, and AVG are the available options. The second data field is an I10 field, giving the number of input values over which the minimum, maximum, or average is computed.

FILTER	DESCRIPTION
CONTINUOUS	Similar to the DISCRETE filter, except that it operates over a fixed time interval. The first data field is exactly the same as for the DISCRETE option. The second data field is an F10 field, indicating the duration of the filter. For example, if AVG is given, and the duration is set to 0.1, a running timestep weighted average is computed over the last 0.1 time of the simulation.
CHAIN	Here, data fields 1-7 are all I10 fields, and give the IDs of a list of other filters (including other CHAIN filters, if desired), each of which will be applied in order. So the raw data is fed to the filter indicated by Data1. The output of that is fed to the next filter, and so on, with up to 7 filters in the chain. List only as many filters as you need.

*DEFINE_FORMING_BLANKMESH

Purpose: This keyword, together with keyword *ELEMENT_BLANKING, enable mesh generation for a sheet metal blank. This keyword is renamed from the previous keyword *CONTROL_FORMING_BLANKMESH. The keyword *DEFINE_CURVE_-TRIM_NEW can be coupled with this keyword to define a blank with a complex periphery and a number of inner hole cutouts.

Card 1	1	2	3	4	5	6	7	8
Variable	IDMSH	ELENG	XLENG	YLENG	ANGLEX	NPLANE	CID	
Type	I	F	F	F	F	I	I	
Default	none	0.0	0.0	0.0	0.0	1	0	

Card 2	1	2	3	4	5	6	7	8
Variable	PIDBK	NID	EID	XCENT	YCENT	ZCENT	XSHIFT	YSHIFT
Type	I	I	I	F	F	F	F	F
Default	1	1	1	0.0	0.0	0.0	0.0	0.0

VARIABLE	DESCRIPTION
IDMSH	ID of the blankmesh (not the blank PID); must be unique.
ELENG	Element edge length.
XLENG	Length of the rectangular blank along X-axis in the coordinate system (CID) defined.
YLENG	Length of the rectangular blank along Y-axis in the coordinate system (CID) defined.
ANGLEX	An angle defined about Z-axis of the CID specified, starting from the X-axis as the zero degree, to rotate the blank and the orientation of the mesh to be generated. The sign of the rotation angle follows the right hand rule. See **Remark 3**.

VARIABLE	DESCRIPTION
NPLANE	Plane in which a flat blank to be generated, in reference to the coordinate system defined (CID): EQ.0 or 1: XY-plane (default) EQ.2: XZ-plane EQ.3: YZ-plane
CID	ID of the local coordinate system, defined by *DEFINE_COORDINATE_SYSTEM. Default is 0 representing global coordinate system.
PIDBK	Part ID of the blank, as defined by *PART.
NID	Starting node ID of the blank to be generated.
EID	Starting element ID of the blank to be generated.
XCENT	X-coordinate of the center of the blank.
YCENT	Y-coordinate of the center of the blank.
ZCENT	Z-coordinate of the center of the blank.
XSHIFT	Blank shifting distance in X-axis in coordinate system defined (CID).
YSHIFT	Blank shifting distance in Y-axis in coordinate system defined (CID).

About the keyword:

A rectangular blank is defined and meshed, which can be trimmed with IGES curves to a desired periphery and inner cutouts. This keyword is used in conjunction with keyword *ELEMENT_BLANKING. The blank outlines and inner holes can be defined using keyword *DEFINE_CURVE_TRIM_NEW.

Application example:

A partial keyword example of generating a flat blank with PID 1 is provided blow. Referring to Figure 15-34, the blank mesh is to be generated in XY plane in a global coordinate system, with an average element edge length of 12 mm and a blank dimension of 1100.0 x 1050.0 mm, with node and element ID starting at 8000, and with the center of the blank in the global origin. The blank is to be trimmed out with an inner cut-out hole, given by the IGES file *innerholes.iges*. Blank outer line is defined with an IGES file *outerlines.iges*. Both IGES files are used to trim the rectangular blank using keyword *DEFINE_CURVE_TRIM_NEW, where the variable TFLG is used to indicate whether it is an inside or outside trim. The blank generated for example is shown in Figure 15-35.

```
*KEYWORD
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*CONTROL_TERMINATION
$#  endtim
     0.000
*CONTROL_FORMING_BLANKMESH
$    IDMSH     ELENG     XLENG     YLENG    ANGLEX    NPLANE       CID
         3     12.00   1100.00     895.0       0.0         0         0
$    PIDBK       NID       EID     XCENT     YCENT     ZCENT    XSHIFT    YSHIFT
         1      8000      8000
*ELEMENT_BLANKING
$#    psid
         1
*DEFINE_CURVE_TRIM_NEW
$#    tcid    tctype      TFLG      TDIR     TCTOL      TOLN    NSEED1    NSEED2
     11111         2         1         0  0.250000  1.000000
innerholes.iges
*DEFINE_CURVE_TRIM_NEW
$#    tcid    tctype      TFLG      TDIR     TCTOL      TOLN    NSEED1    NSEED2
     11112         2        -1         0  0.250000  1.000000
outerlines.iges
*CONTROL_SHELL
......
*CONTROL_SOLUTION
......
*DATABASE_BINARY_D3PLOT
......
*DATABASE_EXTENT_BINARY
......
*SET_PART_list
1
1
*PART
Blank
$#     pid     secid       mid
         1         1         1
*SECTION_SHELL
$#   secid    elform      shrf       nip     propt   qr/irid     icomp     setyp
         1        16  0.833000         7         1         0         0         0
$#      t1        t2        t3        t4      nloc     marea      idof    edgset
  1.500000  1.500000  1.500000  1.500000     0.000     0.000     0.000         0
*MAT_037
$#     mid        ro         e        pr      sigy      etan         r     hlcid
         1 7.9000E-9 2.0700E+5  0.300000 253.25900     0.000  1.408000     90903
*DEFINE_CURVE
     90903
253.2590027
......
           0.9898300         616.7999878
*INTERFACE_SPRINGBACK_LSDYNA
$#    psid      nshv
         1      1000
*END
```

The blank and mesh orientation can be rotated about Z-axis defined. Following the right hand rule, the blank in this case is rotated about Z-axis for a positive 30°, as shown in Figure 15-35, with the angle of 0° aligned with X-axis.

Inner hole and outer periphery can also be trimmed using the NSEEDs variables in keyword *DEFINE_CURVE_TRIM_NEW.

Revision information:

This feature is available in LS-DYNA Revision 59165 or later releases. The keyword name change from *CONTROL… to *DEFINE… started in Revision 69074. The variable NPLANE is implemented in Revision 69128 and later releases.

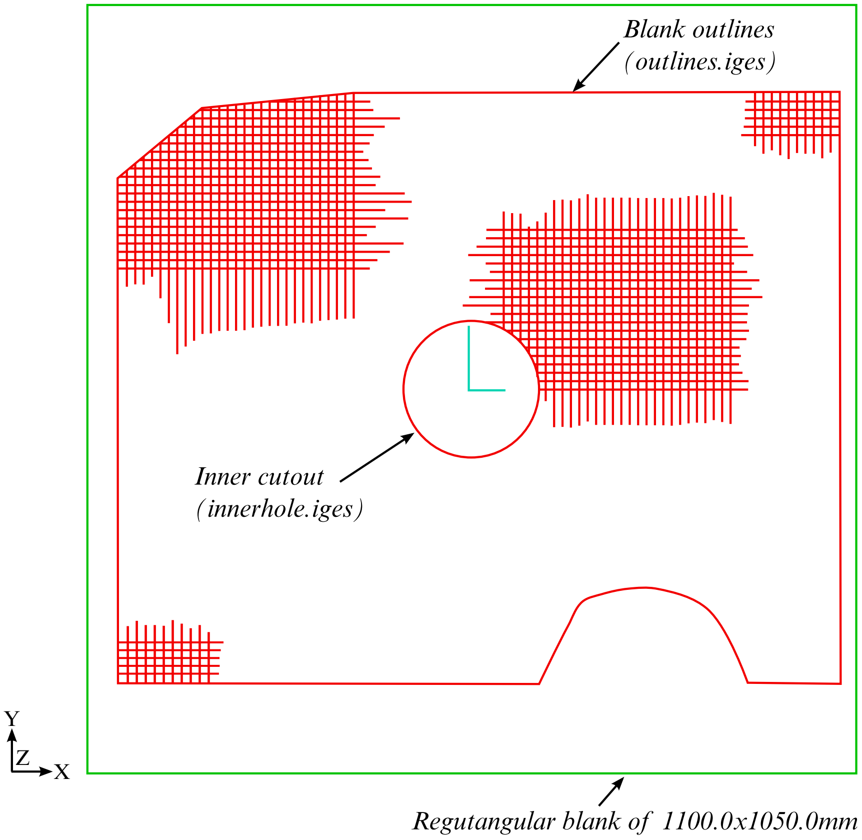


Figure 15-34. Initial input for a blank meshing.

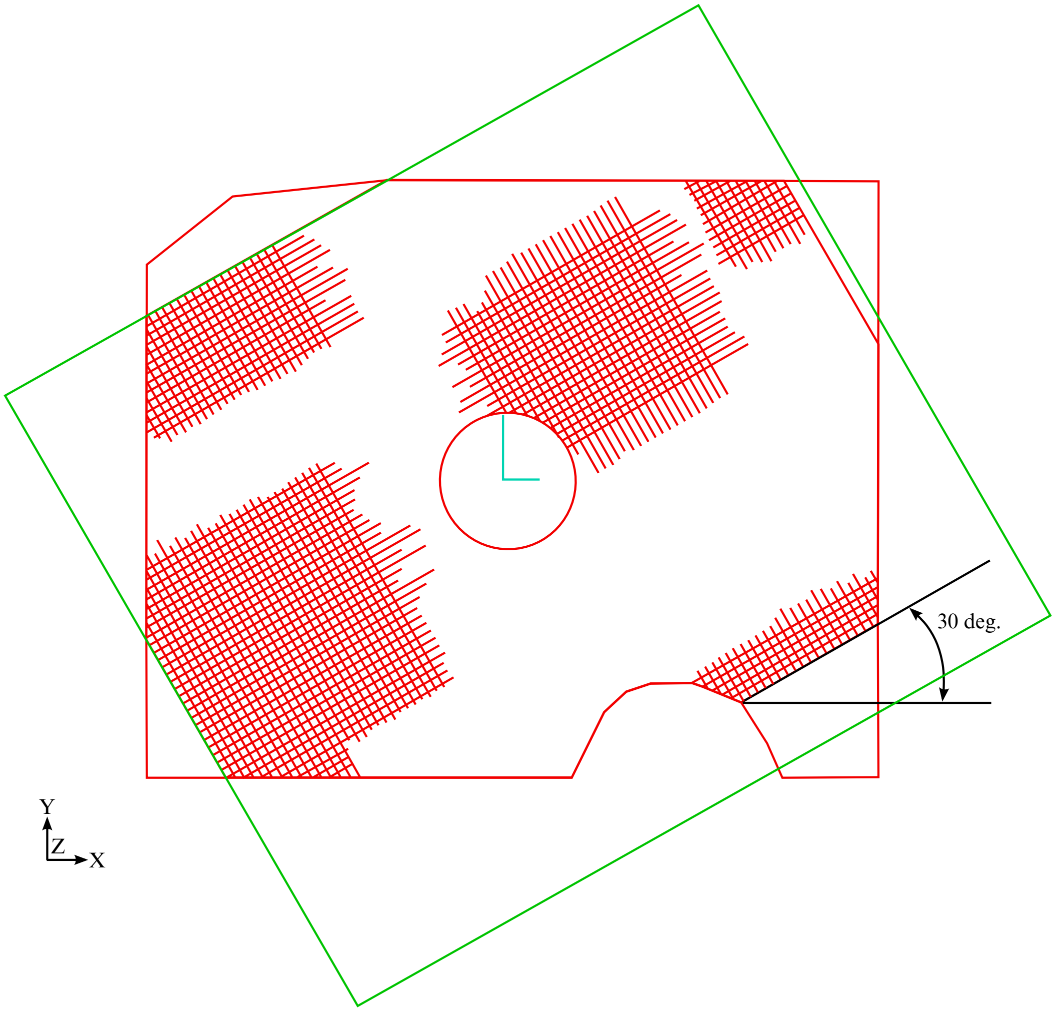


Figure 15-35. Resulting blank mesh.

*DEFINE_FORMING_CLAMP

Purpose: This keyword simplifies the process definition during a clamping simulation, and works as a macro serving as a placeholder for the combination of cards needed to model a clamping process such as those that are commonly used in sheet metal forming. A related keyword includes *DEFINE_FORMING_CONTACT.

Define Clamp Card. Define one card for each clamp set. Include as many cards in the following format as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	CLP1	CLP2	VID	GAP	AT	DT		
Type	I	I	I	F	F	F		
Default	none	none	none	0.0	0.0	0.0		

VARIABLE	DESCRIPTION
CLP1	Part ID of a moving rigid body clamp, defined by *PART and *MAT_020 (*MAT_RIGID).
CLP2	Part ID of a fixed rigid body clamp, defined by *PART and *MAT_020. This is sometimes called "net pad".
VID	Define CLP1 moving direction: GT.0: Vector ID from *DEFINE_VECTOR, specifying the moving direction of CLP1 LT.0: Absolute value is a node ID, whose normal vector will be used to define the moving direction of CLP1.
GAP	Final desired distance between CLP1 and CLP2 at the end of clamping.
AT	Begin time for CLP1's move.
DT	*Duration* of CLP1's move.

About the keyword:

One typical application of this keyword is to estimate springback during the clamping of a formed panel on a checking fixture. Net pads (lower fixed regular or square pads) of a few millimeters thick are placed according to GD&T (*Geometry Dimensioning and Tolerancing*) requirements on a support platform, typically taking shape of the nominal product. Each net pad (CLP2, see Figure 15-36) has a corresponding moving clamp (CLP1).

The movable clamp (CLP1) is initially open so that the formed panel can be loaded onto the net pads. Four-way and two-way position gaging pins are used to initially locate and load the panel in the fixture, before CLP1 is moved to close with the net pad (CLP2). A white light scan is then performed on the panel and scan data is processed to ascertain the degree of panel conformance to the required nominal shape. Even with the clamps fully closed, some severely distorted panels will significantly deviate from the nominal shape when the residual stresses from the forming process are too great.

Although, unrelated to this keyword another common method to determine the panel springback amount is the *free state* check which involves a white light scan on the panel secured on a platform but with no additional forces (clamps – CLP1s) applied to deform the panel (to the net pads CLP2s, for example). LS-DYNA can model both methods and job setups can be easily done by selecting *Implicit Static Flanging* process in *LS-PrePost 4.2's Metal Forming Application / eZ-Setup* (http://ftp.lstc.com/anonymous/outgoing/lsprepost/4.2/win64/). Furthermore, once springback has been determined die compensation (*INTERFACE_COMPENSATION_NEW) can then be performed to minimize or eliminate the springback; the resulting compensated die shapes can be surfaced, re-machined to produce panels that are within dimensional tolerance.

Since the clamp is, typically, modelled with a much coarser mesh than that on a blank, *CONTACT_FORMING_SURFACE_TO_SURFACE should be used. Rotating-type of clamps are currently not supported.

Application example:

A partial keyword example of using the feature is listed below. Referring to Figure 15-36, the drawn and trimmed blank is positioned between the clamps CLP1 and CLP2. The implicit termination "time" is set at 1.0, with a stepping size of 0.25, for a total of four steps – two steps each for the two CLP1. With the original blank thickness of 1.0 mm, the CLP1s are set to close with the lower CLP2s at "time" of 1.0, leaving a total GAP of 1.02 mm. Note the VIDs are defined as "-46980", indicating that the moving clamps (CLP1) will move in the normal direction defined by Node #46980. The contact definition between the panel and the clamps are defined using *DEFINE_FORMING_CONTACT.

```
*KEYWORD
*INCLUDE
./trimmed.dynain
./nets.k
*CONTROL_TERMINATION
1.0
*CONTROL_IMPLICIT_forming
1
*control_implicit_general
1,0.25
*CONTROL_SHELL
⋮
*DATABASE_EXTEND_BINARY
⋮
*PART
Blank
$      PID     SECID       MID
         1         1         1
Clamp1
2,2,2
Clamp2
3,2,3
Clamp3
4,2,2
Clamp4
5,2,3
*MAT_TRANSVERSELY_ANISOTROPIC_ELASTIC_PLASTIC
$      MID        RO         E        PR      SIGY      ETAN         R     HLCID
         1 2.700E-09 12.00E+04      0.28       0.0       0.0     0.672         2
*MAT_RIGID
$#     mid        ro         e        pr         n    couple         m     alias
         2 7.8500E-9 2.1000E+5  0.300000
$#     cmo      con1      con2
  1.000000         7         7
$# lco or a1      a2        a3        v1        v2        v3
     0.000     0.000     0.000     0.000     0.000     0.000

*MAT_RIGID
$#     mid        ro         e        pr         n    couple         m     alias
         3 7.8500E-9 2.1000E+5  0.300000
$#     cmo      con1      con2
  1.000000         4         7
$# lco or a1      a2        a3        v1        v2        v3

*SETION_SHELL
1,16,,7
1.0,1.0,1.0,1.0
*LOAD_BODY_Z
      9997       1.0
*DEFINE_CURVE
      9997
              0.0000            9810.0000
              1.0000            9810.0000
*DEFINE_FORMING_CLAMP
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$     CLP1      CLP2       VID       GAP        AT        DT
         3         2    -46980      1.02       0.0       0.5
         5         4    -46980      1.02       0.5       0.5
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*DEFINE_FORMING_CONTACT
$      IPS       IPM        FS    ONEWAY
         1         2     0.125         1
         1         3     0.125         1
         1         4     0.125         1
         1         5     0.125         1
*END
```

Note with this keyword, formed panel needs to be pre-positioned properly with respect to the clamps by users, and auto-position (*CONTROL_FORMING_AUTOPOSITION) cannot be used. Furthermore, prescribed motions and clamp motion curves do not need to be defined for the clamps.

Revision information:

This feature is available starting in double precision LS-DYNA Revision 99007 for implicit static only.

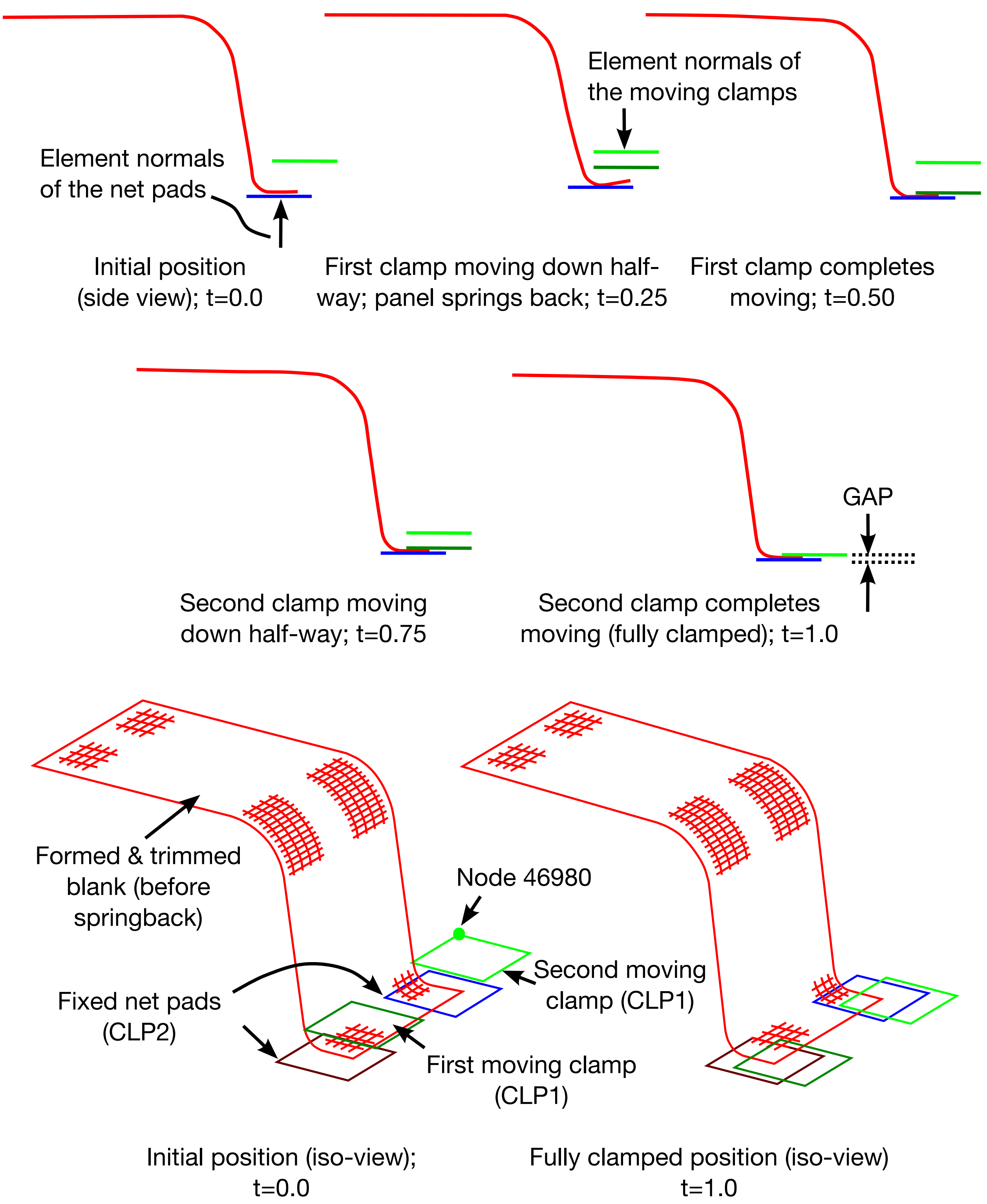

Element normals of the net pads
Initial position (side view); t=0.0
Element normals of the moving clamps
First clamp moving down half-way; panel springs back; t=0.25
First clamp completes moving; t=0.50
Second clamp moving down half-way; t=0.75
GAP
Second clamp completes moving (fully clamped); t=1.0
Formed & trimmed blank (before springback)
Node 46980
Second moving clamp (CLP1)
Fixed net pads (CLP2)
First moving clamp (CLP1)
Initial position (iso-view); t=0.0
Fully clamped position (iso-view) t=1.0

Figure 15-36. Variable definitions and an example of using the negative VID.

*DEFINE_FORMING_CONTACT

Purpose: This keyword works as macro for the FORMING_(ONE_WAY)_SURFACE_TO_SURFACE keyword. It adds one contact definition to the model per data card. Each data card consists of a reduced set of FIELDS compared with the full *CONTACT keyword. The omitted fields take their default values. A related keyword includes *DEFINE_FORMING_CLAMP.

Define Contact Card. Define one card for each contact interface. Define as many cards in the following format as desired. The input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	IPS	IPM	FS	ONEWAY				
Type	I	I	F	I				
Default	none	none	none	0				

VARIABLE	DESCRIPTION
IPS	Part ID of a slave sliding member, typically a deformable sheet metal blank.
IPM	Part ID of a master sliding member, typically a tool or die defined as a rigid body.

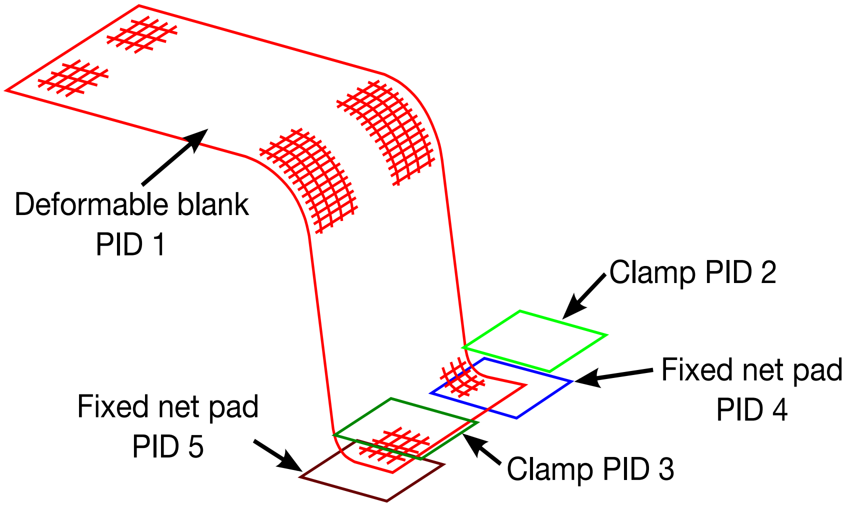


Figure 15-37. Define contact interfaces.

VARIABLE	DESCRIPTION
FS	Coulomb friction coefficient.
ONEWAY	Define FORMING contact type: EQ.0: The contact is FORMING_ONE_WAY_SURFACE_TO_-SURFACE. EQ.1: The contact is FORMING_SURFACE_TO_SURFACE.

Application example:

A partial keyword example of defining contact between a deformable part and two pairs of clamps is given below. In Figure 15-37, a blank PID of 1 is defined to have FORMING_SURFACE_TO_SURFACE contact with rigid body clamps of PID of 2, 3, 4, and 5, with coefficient of frictions for each interface as 0.125, 0.100, 0.125, and 0.100, respectively. Only a total of four lines are needed to define four contact interfaces, as opposed to at least three cards for each interface.

```
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*DEFINE_FORMING_CONTACT
$      IPS       IPM        FS    ONEWAY
         1         2     0.125         1
         1         3     0.100         1
         1         4     0.125         1
         1         5     0.100         1
```

Revision information:

This feature is available starting in LS-DYNA Revision 98988.

*DEFINE_FRICTION

Purpose: Define friction coefficients between parts for use in the contact options:

SINGLE_SURFACE,

AIRBAG_SINGLE_SURFACE,

AUTOMATIC_GENERAL,

AUTOMATIC_SINGLE_SURFACE,

AUTOMATIC_SINGLE_SURFACE_MORTAR,

AUTOMATIC_NODES_TO_SURFACE,

AUTOMATIC_SURFACE_TO_SURFACE,

AUTOMATIC_SURFACE_TO_SURFACE_MORTAR,

AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE,

ERODING_SINGLE_SURFACE.

The input in this section continues until then next "*" card is encountered. Default friction values are used for any part ID pair that is not defined.

The coefficient tables specified by the following cards are activated when FS (see second card of *CONTACT) is set to -2.0. This feature overrides the coefficients defined in *PART_CONTACT (which are turned on only when FS is set to -1.0).

When only *one* friction table is defined, it is used for *all* contacts having FS set to -2. Otherwise, for each contact with FS equal to -2, the keyword reader assigns a table to each *CONTACT by matching the value of FD from *CONTACT with an ID from Card 1 below. Failure to match FD to an ID causes error termination.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	FS_D	FD_D	DC_D	VC_D			
Type	I	F	F	F	F			
Default	0	0.0	0.0	0.0	0.0			

Friction *ij* card. Sets the friction coefficients between parts *i* and *j*. Add as many of these cards to the deck as necessary. The next keyword ("*") card terminates the friction definition.

Card 2…	1	2	3	4	5	6	7	8
Variable	PID*i*	PID*j*	FS*ij*	FD*ij*	DC*ij*	VC*ij*	PTYPE*i*	PTYPE*j*
Type	I	I	F	F	F	F	A	A
Default			0.0	0.0	0.0	0.0		

VARIABLE **DESCRIPTION**

ID

Identification number. Only one table is allowed.

FS_D

Default value of the static coefficient of friction. The frictional coefficient is assumed to dependend on the relative velocity v_{rel} of the surfaces in contact,

$$\mu_c = \mathrm{FD} + (\mathrm{FS} - \mathrm{FD})e^{-\mathrm{DC}|v_{\mathrm{rel}}|}.$$

Default values are used when part pair are undefined. For mortar contact μ_c = FS, i.e., dynamic effects are ignored.

FD_D

Default value of the dynamic coefficient of friction. The frictional coefficient is assumed to depended on the relative velocity v_{rel} of the surfaces in contact,

$$\mu_c = \mathrm{FD} + (\mathrm{FS} - \mathrm{FD})e^{-\mathrm{DC}|v_{\mathrm{rel}}|}.$$

Default values are used when part pair are undefined. For mortar contact μ_c = FS, i.e., dynamic effects are ignored.

DC_D

Default value of the exponential decay coefficient. The frictional coefficient is assumed to be depend on the relative velocity v_{rel} of the surfaces in contact,

$$\mu_c = \mathrm{FD} + (\mathrm{FS} - \mathrm{FD})e^{-\mathrm{DC}|v_{\mathrm{rel}}|}.$$

Default values are used when part pair are undefined. For mortar contact μ_c = FS, i.e., dynamic effects are ignored.

VARIABLE	DESCRIPTION
VC_D	Default value of the coefficient for viscous friction. This is necessary to limit the friction force to a maximum. A limiting force is computed $F_{\text{lim}} = \text{VC} \times A_{\text{cont}},$ where A_{cont} is the area of the segment contacted by the node in contact. The suggested value for VC is to use the yield stress in shear $\text{VC} = \frac{\sigma_o}{\sqrt{3}}$ where σ_0 is the yield stress of the contacted material. Default values are used when part pair are undefined.
PID*i*	Part, or part set, ID *i*.
PID*j*	Part, or part set, ID *j*.
FS*ij*	Static coefficient of friction between parts *i* and *j*.
FD*ij*	Dynamic coefficient of friction between parts *i* and *j*.
DC*ij*	Exponential decay coefficient between parts *i* and *j*.
VC*ij*	Viscous friction between parts *i* and *j*.
PTYPE*i*, PTYPE*j*	EQ.“PSET”: when PTYPE*i* or PTYPE*j* refers to a *SET_PART.

*DEFINE_FRICTION_ORIENTATION

Purpose: This keyword allows for definition of different coefficients of friction (COF) in specific directions, specified using a vector and angles in degree. In addition, COF can be scaled according to the amount of pressure generated in the contact interface. This feature is intended for use with FORMING_ONE_WAY type of contacts. This feature is developed jointly with the Ford Motor Company.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	LCID	LCIDP	V1	V2	V3		
Type	I	I	I	F	F	F		
Default	none	0	0	0.0	0.0	0.0		

VARIABLE	DESCRIPTION
PID	Part ID to which directional and pressure-sensitive COF is to be applied. See *PART.
LCID	ID of the load curve defining COF vs. orientation in degree.
LCIDP	ID of the load curve defining COF scale factor vs. pressure.
V1	Vector components of vector V defining zero-degree (rolling) direction.
V2	Vector components of vector V defining zero-degree (rolling) direction.
V3	Vector components of vector V defining zero-degree (rolling) direction.

The assumption:

Load curves LCID and LCIDP are not extrapolated beyond what are defined. It is recommended that the definition is specified for the complete range of angle and pressure expected. One edge of all elements on the sheet metal blank must align initially with the vector defined by V1, V2, and V3.

Application example:

The following is a partial keyword input of using this feature to define directional frictions and pressure-sensitive COF scale factor.

```
*DEFINE_FRICTION_ORIENTATION
$      PID      LCID     LCIDP        V1        V2        V3
         1        15        16       1.0        0.        0.
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$ define COF vs.  orientation angle
*DEFINE_CURVE
15
0.0, 0.3
90.0, 0.0
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$ define COF scale factor vs.  pressure
*DEFINE_CURVE
16
0.1, 0.3
0.2, 0.3
0.3, 0.3
0.4, 0.2
0.5, 0.5
0.6, 0.4
```

Referring to Figure 15-38, a deformable blank is clamped with 1000N of force between two rigid plates and is pulled along the direction of X-axis for 90 mm using displacement control. Initial and final positions of the blank are shown in Figure 15-39. The normal force is recovered from RCFORC file, as shown in Figure 15-40, which agrees with what is applied. Frictional force (pulling force) in X-direction is plotted as 89N, shown in Figure 15-41. A hand calculation from the input verifies this result:

$$\underbrace{[\text{clamping force}]}_{1000N} \times \underbrace{[x\text{-dir coefficient}]}_{0.3} \times \underbrace{[\text{coefficient scale factor at 0.27 pressure}]}_{0.3} = 90N$$

The interface pressure can be output from an LS-DYNA run when ‘S = filename’ is included in the command line. The binary output can be viewed from LS-PrePost4.0.

The element directions are automatically aligned with the vector V. The left side of Figure 15-42 shows the element directions of the incoming sheet blank. The keyword will re-orient the element directions based on the vector V specified, which has the component of [1.0, 0.0, 0.0] in this case. The re-oriented element directions for the blank are shown on the right side of the Figure.

Following the numeric directions provided in Figure 15-43, LS-PrePost4.0 can be used to check the element directions of a sheet blank.

This feature is generally developed for use with FORMING_ONE_WAY type of contact in SMP. However, this keyword can be used in combination with *CONTACT_FORMING_ONE_WAY_SURFACE TO_SURFACE_ORTHO_FRICTION, and in fact, it can only be used in this manner if running MPP. In this combination, the variables LCID and LCIDP are overridden by friction factor input in ORTHO_FRICTION, while the vector [V1, V2, V3] defines the first orthogonal direction. It furthermore allows the convenience of SSID and MSID in *CONTACT being input as part set IDs (when SSTYP/MSTYP = 2), in which case the segments sets necessary for ORTHO_FRICTION are generated automatically with orientation according to the vectors defined by [V1, V2, V3]. The part set ID input option is typically used by metal forming users. A detailed keyword example is shown in Figure 15-44.

Revision information:

This feature is available in LS-DYNA Revision 60275 and later releases for SMP. It works with MPP with one way forming type of contact with ORTHO_FRICTION starting from Rev 73226. In addition, it works with SMOOTH contact option starting from Revision 69631.

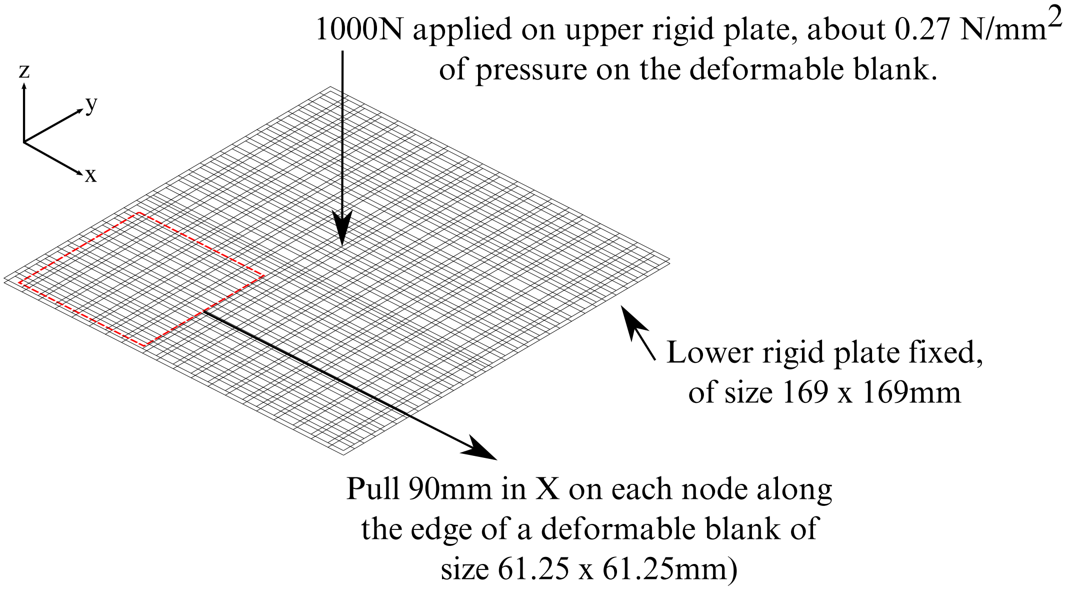

1000N applied on upper rigid plate, about 0.27 N/mm2
of pressure on the deformable blank.
z
y
x
Lower rigid plate fixed,
of size 169 x 169mm
Pull 90mm in X on each node along
the edge of a deformable blank of
size 61.25 x 61.25mm)

Figure [15-38]. Boundary and loading conditions of a small test model.

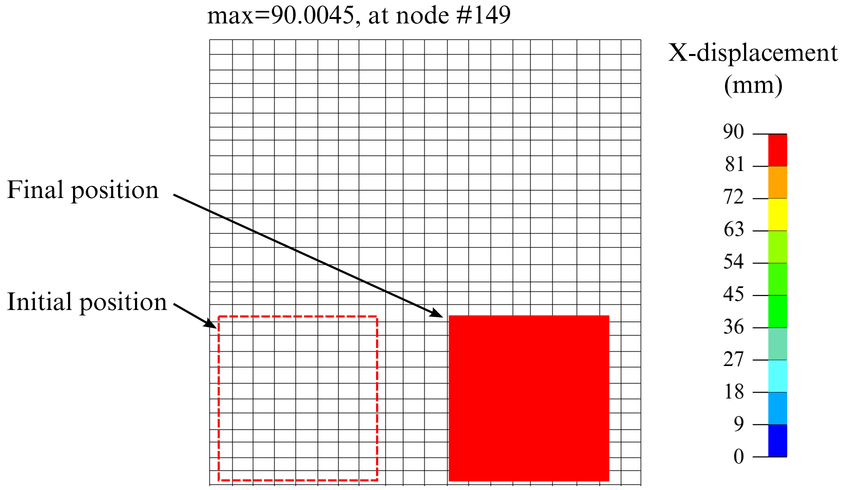


Figure [15-39]. Initial and final position of the blank.

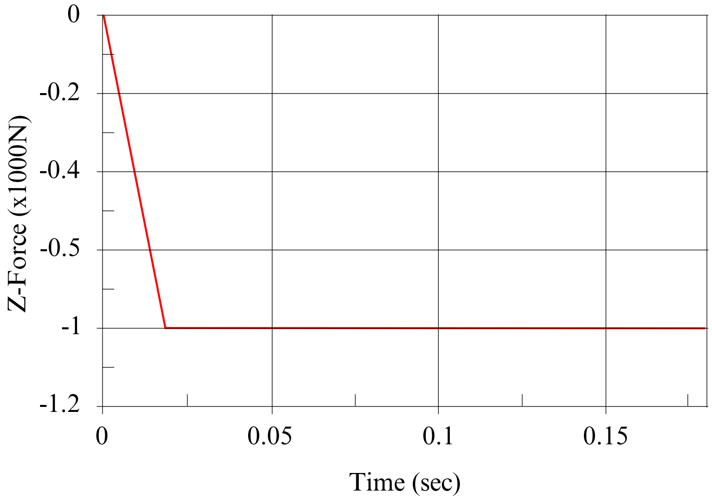


Figure [15-40]. Normal force from RCFORC file.

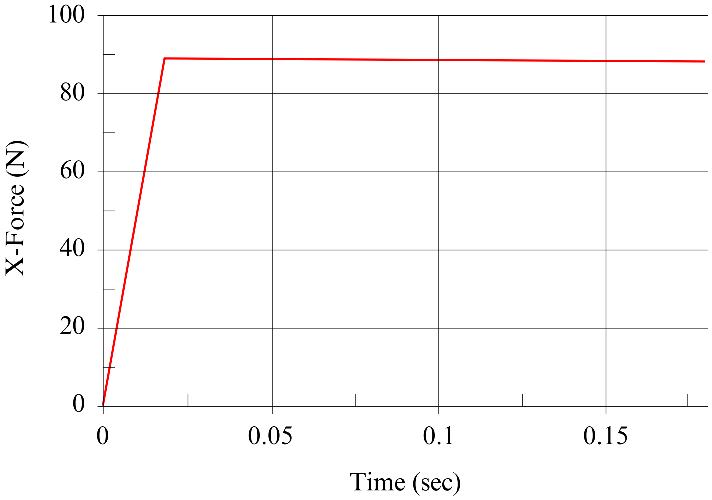


Figure [15-41]. Pulling force (frictional force) from RCFORC file.

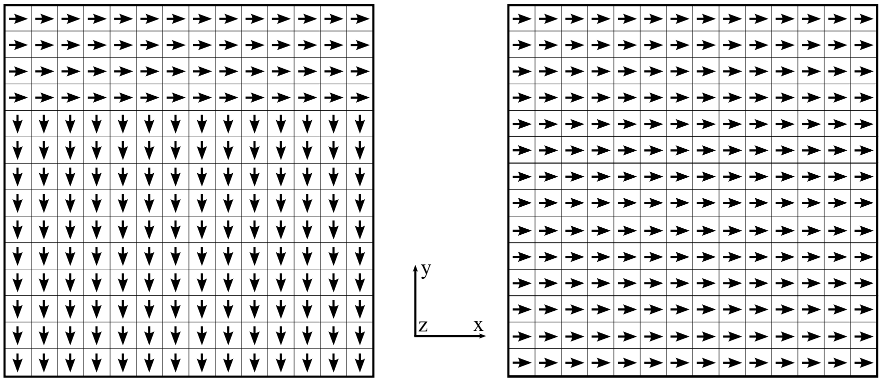


Figure [15-42]. Element directions (N1-N2) of an incoming sheet blank (left) and directions after re-orientation.

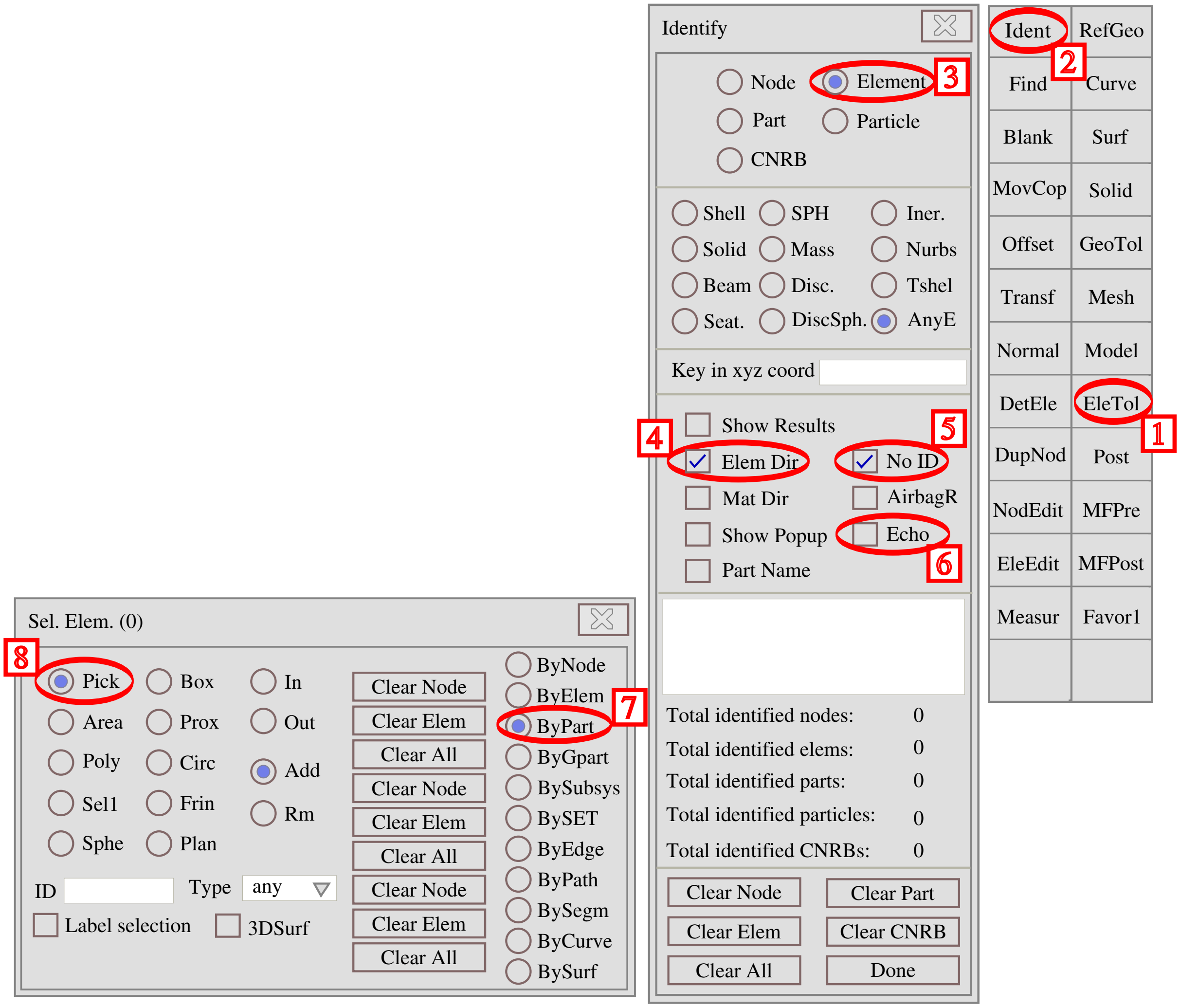
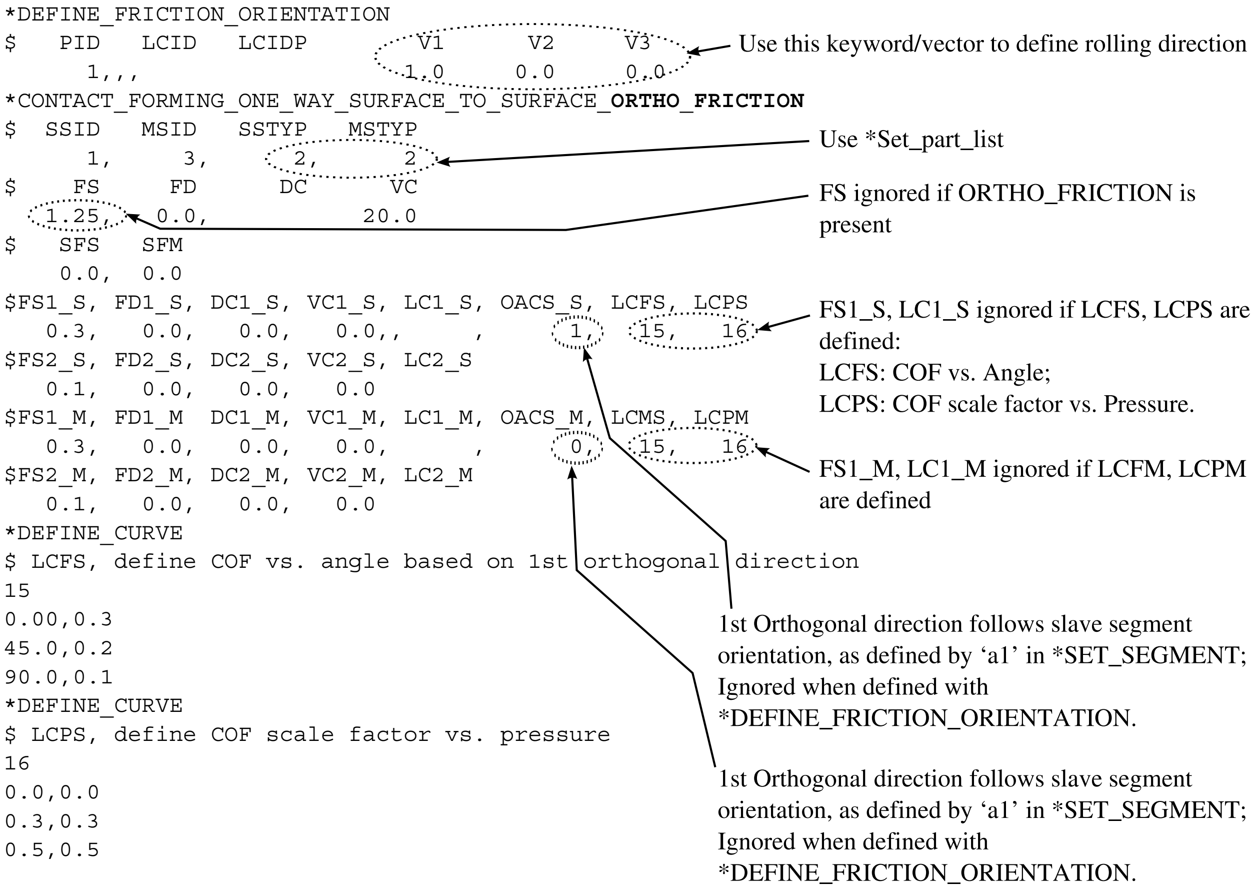


Figure [15-43]. Checking element directions (N1-N2) by part using LS-PrePost4.0.



```
*DEFINE_FRICTION_ORIENTATION
$   PID   LCID   LCIDP        V1      V2      V3
      1,,,                   1.0     0.0     0.0
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ORTHO_FRICTION
$  SSID   MSID   SSTYP   MSTYP
      1,     3,     2,       2
$    FS     FD      DC     VC
   1.25,     0.0,           20.0
$   SFS   SFM
    0.0,  0.0
$FS1_S, FD1_S, DC1_S, VC1_S, LC1_S, OACS_S, LCFS, LCPS
   0.3,   0.0,   0.0,   0.0,,      ,      1,  15,   16
$FS2_S, FD2_S, DC2_S, VC2_S, LC2_S
   0.1,   0.0,   0.0,   0.0
$FS1_M, FD1_M  DC1_M, VC1_M, LC1_M, OACS_M, LCMS, LCPM
   0.3,   0.0,   0.0,   0.0,       ,      0,  15,   16
$FS2_M, FD2_M, DC2_M, VC2_M, LC2_M
   0.1,   0.0,   0.0,   0.0
*DEFINE_CURVE
$ LCFS, define COF vs. angle based on 1st orthogonal direction
15
0.00,0.3
45.0,0.2
90.0,0.1
*DEFINE_CURVE
$ LCPS, define COF scale factor vs. pressure
16
0.0,0.0
0.3,0.3
0.5,0.5
```

Use this keyword/vector to define rolling direction

Use *Set_part_list

FS ignored if ORTHO_FRICTION is present

FS1_S, LC1_S ignored if LCFS, LCPS are defined:
LCFS: COF vs. Angle;
LCPS: COF scale factor vs. Pressure.

FS1_M, LC1_M ignored if LCFM, LCPM are defined

1st Orthogonal direction follows slave segment orientation, as defined by ‘a1’ in *SET_SEGMENT; Ignored when defined with *DEFINE_FRICTION_ORIENTATION.

1st Orthogonal direction follows slave segment orientation, as defined by ‘a1’ in *SET_SEGMENT; Ignored when defined with *DEFINE_FRICTION_ORIENTATION.

Figure [15-44]. Use of this keyword with _ORTHO_FRICTION for MPP.

*DEFINE_FUNCTION

Purpose: Define a function that can be referenced by a limited number of keyword options. The function arguments are different for each keyword that references *DEFINE_FUNCTION. Unless stated otherwise, all the listed argument(s) in their correct order must be included in the argument list. Some usages of *DEFINE_FUNCTION allow random ordering of arguments and argument dropouts. See the individual keywords for the correct format. Some examples are shown below.

The TITLE option is not allowed with *DEFINE_FUNCTION.

Card 1	1	2	3	4	5	6	7	8
Variable	FID	HEADING						
Type	I	A70						

Function Cards. Insert as many cards as needed. These cards are combined to form a single line of input. The next keyword ("*") card terminates this input.

Card	1	2	3	4	5	6	7	8
Variable	FUNCTION							
Type	A80							

VARIABLE	DESCRIPTION
FID	Function ID. Functions, tables (see *DEFINE_TABLE), and load curves may not share common ID's. A unique number has to be defined.
HEADING	An optional descriptive heading.
FUNCTION	Arithmetic expression involving a combination of independent variables and other functions, i.e., "f(a,b,c)=a*2+b*c+sqrt(a*c)" where a, b, and c are the independent variables. The function name, "f(a,b,c)", must be unique since other functions can then use and reference this function. For example, "g(a,b,c,d)=f(a,b,c)**2+d". In this example, two *DEFINE_FUNCTION definitions are needed to define functions f and g.

Remarks:

The following examples serve only as an illustration of syntax.

Unlike *DEFINE_CURVE and *DEFINE_CURVE_FUNCTION, *DEFINE_FUNCTION is always active in dynamic relaxation phase.

Example 1:

Prescribe sinusoidal x-velocity and z-velocity for some nodes.

```
*BOUNDARY_PRESCRIBED_MOTION_SET
$#    nsid       dof       vad      lcid        sf
         1         1         0         1
         1         3         0         2
*DEFINE_FUNCTION
1,x-velo
x(t)=1000*sin(100*t)
*DEFINE_FUNCTION
2,z-velo
a(t)=x(t)+200
```

Example 2:

Ramp up a hydrostatic pressure on a submerged surface.

```
*comment
units: mks

Apply a hydrostatic pressure ramped up over a finite time = trise.

pressure on segment = rho * grav * depth of water
where depth of water is refy - y-coordinate of segment
and refy is the y-coordinate of the water surface

*DEFINE_FUNCTION
10
float hpres(float t, float x, float y, float z, float x0, float y0,
      float z0)
{
  float  fac, trise, refy, rho, grav;
  trise = 0.1; refy = 0.5; rho = 1000.; grav = 9.81;
  fac = 1.0;
  if(t<=trise) fac = t/trise;
  return fac*rho*grav*(refy-y);
}
*LOAD_SEGMENT_SET
1,10
```

Example 2 illustrates that a programming language resembling C can be used in defining a function. Before a variable or function is used, its type must be declared; that is the purpose of "float" (i.e., a real variable rather than integer type) appearing before those entities. The braces indicate the beginning and end of the function being programmed. Semicolons must appear after each statement but several statements may appear on a single line. Please refer to a C programming guide for more detailed information.

*DEFINE_FUNCTION_TABULATED

Purpose: Define a function of one variable using two columns of input data (in the manner of *DEFINE_CURVE) that can be referenced by a limited number of keyword options or by other functions defined via *DEFINE_FUNCTION.

The TITLE option is not allowed with *DEFINE_FUNCTION_TABULATED.

Card 1	1	2	3	4	5	6	7	8
Variable	FID	HEADING						
Type	I	A70						

Card 2	1	2	3	4	5	6	7	8
Variable	FUNCTION							
Type	A80							

Point Cards. Put one pair of points per card (2E20.0). Add as many cards as necessary. Input is terminated when a keyword ("*") card is found.

Cards 3	1	2	3	4	5	6	7	8
Variable	A1		O1					
Type	F		F					
Default	0.0		0.0					

VARIABLE	DESCRIPTION
FID	Function ID. Functions, tables (see *DEFINE_TABLE), and load curves may not share common ID's. A unique number has to be defined.
HEADING	An optional descriptive heading.
FUNCTION	Function name.

VARIABLE	DESCRIPTION
A1, A2, …	Abscissa values.
O1, O2, …	Ordinate (function) values.

Example:

```
*BOUNDARY_PRESCRIBED_MOTION_SET
$  function 300 prescribes z-acceleration of node set 1000
1000,3,1,300
*DEFINE_FUNCTION_TABULATED
201
tabfunc
0., 200
0.03, 2000.
1.0, 2000.
*DEFINE_FUNCTION
300
a(t)=tabfunc(t)*t
$$  following function is equivalent to one above for t < 0.03
$ a(t)=(200.  + 60000.*t)*t
```

*DEFINE_GROUND_MOTION

Purpose: Define an earthquake ground motion history using ground motion records provided as load curves, for use in conjunction with *LOAD_SEISMIC_SSI for dynamic earthquake analysis including nonlinear soil-structure interaction.

Card 1	1	2	3	4	5	6	7	8
Variable	GMID	ALCID	VLCID					
Type	I	I	I					
Default	none	none	0					

VARIABLE	DESCRIPTION
GMID	Ground motion ID. A unique number has to be defined.
ALCID	Load curve ID of ground acceleration history.
VLCID	Load curve ID of ground velocity history.

Remarks:

1. Earthquake ground motion data is typically available either only as ground accelerations, or as a triple of ground accelerations, velocities and displacements. Usually, the velocities and the displacements are computed from the accelerations using specialized filtering and baseline correction techniques, e.g. see peer.berkeley.edu/smcat/process.html. Either input is accepted, with each quantity specified as a load curve. Only the acceleration and the velocity is required in the latter case; LS-DYNA does not require the ground displacement.

2. If only the ground acceleration data is provided for a particular ground motion, LS-DYNA generates a corresponding load curve for the velocity by integrating the acceleration numerically. The generated load curves are printed out to the D3HSP file. It is up to the user to ensure that these generated load curves are satisfactory for the analysis.

*DEFINE_HAZ_PROPERTIES

Purpose: To model the heat affect zone in a welded structure, the yield stress and failure strain are scaled in shell models as a function of their distance from spot welds and the nodes specified in *DEFINE_HAZ_TAILOR_WELDED_BLANK.

Card 1	1	2	3	4	5	6	7	8
Variable	ID_HAZ	IOP	PID	PID_TYP				
Type	I	I	I	I				
Default	0	0	0	0				

Card 2	1	2	3	4	5	6	7	8
Variable	ISS	IFS	ISB	IFB	ISC	IFC	ISW	IFW
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

VARIABLE	DESCRIPTION
ID_HAZ	Property set ID. A unique ID number must be used.
IOP	Activity flag. If IOP = 0, then the scaling is not applied, and if IOP = 1, the scaling is active.
PID	Part or part set ID.
PID_TYP	PID type. PID_TYP = 0 indicates that PID is a *PART ID, and PID_TYP = 1, a part set.
ISS	Curve ID for scaling the yield stress based on the distance to the closest solid element spot weld. Use a negative ID for curves normalized by the spot weld diameter as described in the Remarks below.

VARIABLE	DESCRIPTION
IFS	Curve ID for scaling the failure strain based on the distance to the closest solid element spot weld. Use a negative ID for curves normalized by the spot weld diameter as described in the Remarks below.
ISB	Curve ID for scaling the yield stress based on the distance to the closest beam element spot weld. Use a negative ID for curves normalized by the spot weld diameter as described in the Remarks below.
IFB	Curve ID for scaling the failure strain based on the distance to the closest beam element spot weld. Use a negative ID for curves normalized by the spot weld diameter as described in the Remarks below.
ISC	Curve ID for scaling the yield stress based on the distance to the closest constrained spot weld. Use a negative ID for curves normalized by the spot weld diameter as described in the Remarks below.
IFC	Curve ID for scaling the failure strain based on the distance to the closest constrained spot weld. Use a negative ID for curves normalized by the spot weld diameter as described in the Remarks below.
ISW	Curve ID for scaling the yield stress based on the distance to the closest tailor welded blank node. Use a negative ID for curves normalized by the spot weld diameter as described in the Remarks below.
IFW	Curve ID for scaling the failure strain based on the distance to the tailor welded blank node. Use a negative ID for curves normalized by the spot weld diameter as described in the Remarks below.

Remarks:

The yield stress and failure strain are assumed to vary radially as a function of the distance of a point to its neighboring spot welds. Since larger spot welds may have a larger radius of influence, the smallest scale factor for the yield stress from all the neighboring spot welds is chosen to scale the yield stress at a particular point. The failure strain uses the scaling curve for the same weld.

Curve IDs may be input as negative values to indicate that they are normalized by the diameter of the spot weld to compensate for the effects of the spot weld size. When this option is used, the scale factor is calculated based on the distance divided by the spot weld diameter for the spot weld that is closest to the element.

The distance from a spot weld (or node for the blank) is measured along the surface of the parts in the part set. This prevents the heat softening effects of a weld from jumping across empty space.

The HAZ capability only works with parts with materials using the STOCHASTIC option. It may optionally be simultaneously used with *DEFINE_STOCHASTIC_VARIATION to also account for the spatial variations in the material properties. See *DEFINE_STOCHASTIC_VARIATION for more details.

*DEFINE_HAZ_TAILOR_WELDED_BLANK

Purpose: Specify nodes of a line weld such as in a Tailor Welded Blank. The yield stress and failure strain of the shell elements in the heat affected zone (HAZ) of this weld are scaled according to *DEFINE_HAZ_PROPERTIES.

Card 1	1	2	3	4	5	6	7	8
Variable	IDTWB	IDNS						
Type	I	I						
Default	0	0						

VARIABLE	DESCRIPTION
IDTWB	Tailor Welded Blank ID
IDNS	Node Set ID defining the location of the line weld.

DEFINE_HEX_SPOTWELD_ASSEMBLY_*{OPTION}

Purpose: Define a list of hexahedral elements that make up a single spot weld for computing the force and moment resultants that are written into the swforc output file. A maximum of 16 elements may be used to define an assembly representing a single spot weld. See Figure 15-45. This table of element IDs is generated automatically when beam elements are converted to solid elements. See the input parameter RPBHX associated with the keyword *CONTROL_SPOTWELD_BEAM.

Available options for this command are:

<BLANK>

N

For the <BLANK> option, all solid elements specified on Card 2 make up the spot weld and no additional card is read. For the N option, N is an integer representing the total number of solid elements making up the spot weld. If N is greater than 8, the additional card beyond Card 2 is read. N may not exceed 16.

Card 1	1	2	3	4	5	6	7	8
Variable	ID_SW							
Type	I							
Default	0							

Card 2	1	2	3	4	5	6	7	8
Variable	EID1	EID2	EID3	EID4	EID5	EID6	EID7	EID8
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

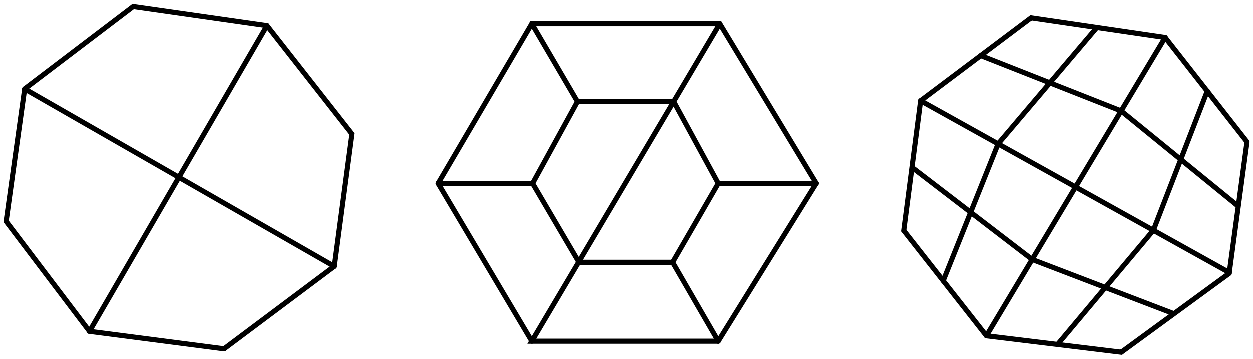


Figure 15-45. Illustration of four, eight, and sixteen element assemblies of solid hexahedron elements forming a single spot weld.

Additional card for N > 8.

Optional	1	2	3	4	5	6	7	8
Variable	EID9	EID10	EID11	EID12	EID13	EID14	EID15	EID16
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

VARIABLE	DESCRIPTION
ID_SW	Spot weld ID. A unique ID number must be used.
EID*n*	Element ID *n* for up to 16 solid hexahedral elements.

Remarks:

The elements comprising a spot weld assembly may share a part ID (PID) with elements in other spot weld assemblies defined using *DEFINE_HEX_SPOTWELD_ASSEMBLY but may not share a PID or even a material ID (MID) with elements that are not included in a *DEFINE_HEX_SPOTWELD_ASSEMBLY.

*DEFINE_LANCE_SEED_POINT_COORDINATES

Purpose: The keyword is to activate the trimming in lancing. It is used in conjunction with *ELEMENT_LANCING to define a seed point which would be on the remaining part after lancing and trim.

Card 1	1	2	3	4	5	6	7	8
Variable	NSEED	X1	Y1	Z1	X2	Y2	Z2	
Type	I	F	F	F	F	F	F	
Default	none	0	0	0	0	0.0	0.0	

VARIABLE	DESCRIPTION
NSEED	Number of seed points. Maximum value of “2” is allowed.
X1, Y1, Z1	Location coordinates of seed point #1.
X2, Y2, Z2	Location coordinates of seed point #2.

Remarks:

1. This keyword will remove all scraps during or after lancing, dependent on how the parameter AT is defined in *ELEMENT_LANCING. Lancing curves must form a closed loop, meaning first and last point coordinates must be coincident. Scraps are the portions that are exclusive of the portions whose seed points are defined by this keyword.

2. The following input defines two sets of seed point coordinates, where a double-attached part may be lanced and trimmed:

```
*DEFINE_LANCE_SEED_POINT_COORDINATES
$    NSEED        X1        Y1        Z1        X2        Y2        Z2
         2    -289.4     98.13  2354.679    -889.4     91.13   255.679
```

3. Refer to manual pages in *ELEMENT_LANCING for more details.

Revision Information

This feature is available in LS-DYNA Revision 107262 and later releases.

:

*DEFINE_MATERIAL_HISTORIES

Purpose: To control the content of the history variables in the d3plot database. This feature is supported for solid, beam and shell elements.

Define as many cards as needed to define the extra history variables. This input ends at the next keyword "*" card.

Card 1	1	2	3	4	5
Variable	LABEL	A1	A2	A3	A4
Type	A40	F	F	F	F
Default	none	0.0	0.0	0.0	0.0

VARIABLE	DESCRIPTION
LABEL	String identifying history variable type.
A*n*	Attributes, see discussion below.

Remarks:

Material models in LS-DYNA use history variables that are specific to the constitutive model being used. For most materials, 6 are reserved for the Cauchy stress components and 1 for the effective plastic strain, but many models have more than that. The history variables may include interesting physical quantities like material damage, material phase compositions, strain energy density and strain rate, but, in addition, they may also include nonphysical quantities like material direction cosines, scale factors and parameters that are used for the numerical algorithms but hard to interpret when post-processed.

By using NEIPS, NEIPB and NEIPH on *DATABASE_EXTENT_BINARY, these extra history variables can be exported to the d3plot database *in the order that they are stored*, and in LS-PrePost the variables may then be plotted (Hist button) or fringed (Misc menu). There are a few drawbacks with this approach. The user must, for instance, have knowledge of the storage location of a certain history variable for a given material model and element type. While this information can be retrieved either in the LS-DYNA manual, on LS-DYNA support sites or in LS-PrePost itself, it is not always convenient.

Furthermore, the same physical quantity may be stored in different locations for different materials and different element types, meaning that history variable #1 will correspond to different things in different parts which complicate post-processing of large models. The user may also be interested in a certain material specific quantity that is not necessarily stored as a history variable, this is not retrievable using this approach. Finally, if the history variable of interest happens to be stored in a bad location, i.e., among the last ones in a long list, it would be necessary to set NEIPS, NEIPB and/or NEIPH large enough to access this variable in LS-PrePost. This could result in unnecessarily large binary plot files.

The present keyword is an attempt to organize the extra history variables with respect to uniformity, a goal is to get an output that is reasonably small and easy to interpret. The input is very simple, use the keyword *DEFINE_MATERIAL_HISTORIES in the keyword input deck, followed by lines that specify the history variables of interest using predetermined labels and attributes. NEIPS, NEIPB and NEIPH on *DATABASE_EXTENT_BINARY will then be overridden by the number of history variables, i.e., number of lines, requested on this card. As an example

```
*DEFINE_MATERIAL_HISTORIES
Instability
Damage
```

would mean that two extra history variables are output to the d3plot database, so NEIPS, NEIPB and NEIPH will internally be set to 2 regardless of the user input. History variable #1 will correspond to an instability measure (between 0 and 1) and history variable #2 will correspond to a material damage (between 0 and 1), i.e., the history variables are output in the order they are listed. If there are several instances of this keyword in an input deck, then the order of the history variables will follow the order that the cards are read by the keyword reader.

In the d3hsp file the user may find the complete list by searching for the string "Material History List". For a material that does not store or calculate an "instability" or "damage" history variable, the output will be zero and thus the output will not be cluttered by unwanted data. Note that this keyword does not necessarily require that the history variable be stored, as long as it can be calculated when LS-DYNA outputs a plot state. This opens for the possibility to request quantities that are not available by just using NEIPS, NEIPB and/or NEIPH on *DATABASE_EXTENT_BINARY.

For large models with many different parts and materials, the *Instability* or *Damage* variables should provide a comprehensive overview and understanding of the critical areas in terms of failure that otherwise may not be assessable.

The permitted LABELs (**case-sensitive)** are

Instability A number between 0 and 1 that indicates how close a element or integration point is to failure or to initiate damage, no attributes apply to this label.

Damage A number between 0 and 1 indicating the damage level of the element or integration point, no attributes apply to this label.

Plastic Strain Rate Effective plastic strain rate, calculated generically for "all" plastic materials from the evolution of effective plastic strain, no attributes apply to this label.

History Fetch a history variable at a known place for a given material type, element type and part set. The following attributes apply.

A1 History variable location, mandatory.

A2 Material type, default applies to all materials.

A3 Element type, 0 for all element types, 1 for solids, 2 for shells and 3 for beams.

A4 Part set, history will only be fetched from the parts in the given part set. The default is all parts in the model.

Examples:

The History label is for users who know where history variables of interest are stored and want to use this to compress or simplify the output. An example is

```
*DEFINE_MATERIAL_HISTORIES
History,4,272,1,23
History,1,81
*SET_PART_LIST
23
2,3
```

which will make a list of two history variables in the output. History variable #1 will fetch the 4th history variable, but displayed only for the RHT concrete model (material 272), solid elements and in parts 2 and 3. History variable #2 will fetch the 1st history variable, display it only for the plasticity with damage model (material 81), but for any element and part. Both of these requested variables happen to be the damage in the respective materials, so an alternative to do something similar would be to use

```
*DEFINE_MATERIAL_HISTORIES
Damage
```

for which the damage for all materials will be displayed in history variable #1.

The history variables that may be requested using this keyword are tabulated in the individual material model chapters, see Volume II of the Keyword Users' Manual. At the end of the remarks for a material model, a table such as the one below is present if there are retrievable history variables.

**DEFINE_MATERIAL_HISTORIES Properties*

Label	Attributes				Description
Instability	-	-	-	-	Failure indicator $\varepsilon^p_{\text{eff}}/\varepsilon^p_{\text{fail}}$, see FAIL
Plastic Strain Rate	-	-	-	-	Effective plastic strain rate $\dot{\varepsilon}^p_{\text{eff}}$

Whether mentioned in such table or not, *Plastic Strain Rate* is available for any material model that calculates plastic strain as the 7th standard history variable. *Label* in the table states what the string LABEL on *DEFINE_MATERIAL_HISTORIES must be, *a1* to *a4* will list attributes A1 to A4 if necessary, and *Description* will be a short description of what is output with this option, including possible restrictions. Further development of this keyword will mainly be driven by customer requests submitted to *suggestions@lstc.com*. Currently only solid, beam and shell elements are supported for the binary d3plot format, a goal is to include thick shells and support ascii/binout output in future versions of LS-DYNA.

Note:
The *Labels* are **case-sensitive**.

*DEFINE_MULTI_DRAWBEADS_IGES

Purpose: This keyword is developed to simplify the creation and definition of draw beads, which previously required the use of many keywords.

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	A80							
Default	none							

Card 2	1	2	3	4	5	6	7	8
Variable	DBID	VID	PID	BLKID	NCUR			
Type	I	I	I	I	I			
Default	none	none	1	1	none			

IGES Curve ID cards. For multiple draw bead curves include as many cards as necessary. Input is terminated at the next (“*”) card.

Card 3	1	2	3	4	5	6	7	8
Variable	CRVID	BFORCE						
Type	I	F						
Default	none	0.0						

VARIABLE	DESCRIPTION
DBID	Draw bead set ID, which may consists many draw bead segments.
VID	Vector ID, as defined by *DEFINE_VECTOR. This vector is used to project the supplied curves to the rigid tool, defined by the parameter PID below.

VARIABLE	DESCRIPTION
PID	Part ID of a rigid tool to which the curves are projected and attached.
BLKID	Part ID of the blank.
NCUR	Number of draw bead curve segments (in the IGES file defined by FILENAME) to be defined.
CVRID	IGES curve ID for each segment.
BFORCE	Draw bead force for each segment.

Remarks:

1. This keyword alone can be used to define draw bead forces around a stamping part. The following partial keyword example shows a draw bead set with ID 98, consists of three curves with ID, 12, 23, and 45, each with bead forces of 102.1, 203.3, 142.5 Newton/mm, respectively, are being created for blank with part ID 1. The beads are projected along vector ID 99, and are attached to a rigid tool with part ID 3. The IGES file to define the draw bead curve is "drawbeads3.iges".

```
*DEFINE_MULTI_DRAWBEADS_IGES
drawbead3.iges
$     DBID       VID       PID     BLKID      NCUR
        98        99         3         1         3
$    CRVID    BFORCE
        12     102.1
        23     203.3
        45     142.5
*define_vector
99,0.0,0.0,0.0,0.0,0.0,1.0
```

Revision information:

This feature is available in LS-DYNA R5 Revision 62840 and later releases.

*DEFINE_PBLAST_AIRGEO

Purpose: To define a simple geometry for initial air domain.

Card 1	1	2	3	4	5	6	7	8
Variable	GID	GTYPE1	GTYPE2					
Type	I	I	I					
Default	0	0	0					

Card 2	1	2	3	4	5	6	7	8
Variable	XA	YA	ZA	XB	YB	ZB		
Type	F	F	F	F	F	F		
Default	0.	0.	0.	0.	0.	0.		

Card 3	1	2	3	4	5	6	7	8
Variable	X0	Y0	Z0	G1	G2	G3		
Type	F	F	F	F	F	F		
Default	0.	0.	0.	0.	0.	0.		

Card 4	1	2	3	4	5	6	7	8
Variable	XC	YC	ZC	G4	G5	G6		
Type	F	F	F	F	F	F		
Default	0.	0.	0.	0.	0.	0.		

VARIABLE	DESCRIPTION
GID	ID of a GEOMETRY defining initial air particle domain.
GTYPE1 GTYPE2	Geometry type EQ.1: box EQ.2: sphere EQ.3: cylinder EQ.4: ellipsoid EQ.5: hemisphere (see Remark 1)
XA, YA, ZA	(XA, YA, ZA) defines a vector of the x-axis
XB, YB, ZB	(XB, YB, ZB) defines a vector of the y-axis
X0, Y0, Z0	Center coordinates of air domain
G1	Dimension value depending on GTYPE. GTYPE.EQ.1: length of x edge GTYPE.EQ.2: Radius of sphere GTYPE.EQ.3: Radius of cross section GTYPE.EQ.4: length of x-axes GTYPE.EQ.5: Radius of hemisphere
G2	Dimension value depending on GTYPE. GTYPE.EQ.1: length of y edge GTYPE.EQ.3: length of cylinder GTYPE.EQ.4: length of y-axes
G3	Dimension value depending on GTYPE. GTYPE.EQ.1: length of z edge GTYPE.EQ.4: length of z-axes
XC, YC, ZC	Center coordinates of domain excluded from the air domain
G4, G5, G6	See definition of G1, G2, G3

Remarks:

1. If GTYPE1/GTYPE2 is 5, the hemisphere is defined in negative z direction defined by the cross product of the y and x axis.

*DEFINE_PBLAST_GEOMETRY

Purpose: To define a simple geometry for high explosives domain.

Card 1	1	2	3	4	5	6	7	8
Variable	GID	GTYPE						
Type	I	I						
Default	0	0						

Card 2	1	2	3	4	5	6	7	8
Variable	XA	YA	ZA	XB	YB	ZB		
Type	F	F	F	F	F	F		
Default	0.	0.	0.	0.	0.	0.		

Card 3	1	2	3	4	5	6	7	8
Variable	Xc	Yc	Zc					
Type	F	F	F					
Default	0.	0.	0.					

Card 4	1	2	3	4	5	6	7	8
Variable	G1	G2	G3					
Type	F	F	F					
Default	0.	0.	0.					

VARIABLE	DESCRIPTION
GID	ID of a GEOMETRY defining high explosive particle domain.
GTYPE	Geometry type EQ.1: box EQ.2: sphere EQ.3: cylinder EQ.4: ellipsoid EQ.5: hemisphere (see Remark 1)
XA, YA, ZA	(XA, YA, ZA) defines a vector of the x-axis
XB, YB, ZB	(XB, YB, ZB) defines a vector of the y-axis
XC	X-coordinate of charge center
YC	Y-coordinate of charge center
ZC	Z-coordinate of charge center
G1	GTYPE.EQ.1: length of X edge GTYPE.EQ.2: Radius of sphere GTYPE.EQ.3: Radius of cross section GTYPE.EQ.4: length of X-axes GTYPE.EQ.5: Radius of hemisphere
G2	GTYPE.EQ.1: length of Y edge GTYPE.EQ.3: length of cylinder GTYPE.EQ.4: length of Y-axes
G3	GTYPE.EQ.1: length of Z edge GTYPE.EQ.4: length of Z-axes

Remarks:

1. If GTYPE is 5, the hemisphere is defined in negative Z direction defined by the cross product of the Y and X axis.

*DEFINE_PLANE

Purpose: Define a plane with three non-collinear points. The plane can be used to define a reflection boundary condition for problems like acoustics.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	X1	Y1	Z1	X2	Y2	Z2	CID
Type	I	F	F	F	F	F	F	I
Default	0	0.0	0.0	0.0	0.0	0.0	0.0	0

Card 2	1	2	3	4	5	6	7	8
Variable	X3	Y3	Z3					
Type	F	F	F					
Default	0.0	0.0	0.0					

VARIABLE	DESCRIPTION
PID	Plane ID. A unique number has to be defined.
X1	X-coordinate of point 1.
Y1	Y-coordinate of point 1.
Z1	Z-coordinate of point 1.
X2	X-coordinate of point 2.
Y2	Y-coordinate of point 2.
Z2	Z-coordinate of point 2.
CID	Coordinate system ID applied to the coordinates used to define the current plane. The coordinates X1, Y1, Z1, X2, Y2, Z2, X3, Y3 and Z3 are defined with respect to the coordinate system CID.
X3	X-coordinate of point 3.

VARIABLE	DESCRIPTION
Y3	Y-coordinate of point 3.
Z3	Z-coordinate of point 3.

Remarks:

1. The coordinates of the points must be separated by a reasonable distance and not collinear to avoid numerical inaccuracies.

DEFINE_POROUS_*OPTION

Available options include:

ALE

LAGRANGIAN

Purpose: The *DEFINE_POROUS_ALE card defines the Ergun porous coefficients for ALE elements. It is to be used with *LOAD_BODY_POROUS. This card with the LAGRANGIAN option, *DEFINE_POROUS_LAGRANGIAN, defines the porous coefficients for Lagrangian elements and is to be used with *CONSTRAINED_LAGRANGE_IN_SOLID (slave parts with CTYPE = 11 or 12).

Card 1	1	2	3	4	5	6	7	8
Variable	EIDBEG	EIDEND	LOCAL	VECID1	VECID2	USERDEF		
Type	I	I	I	I	I	I		
Default	none	0	0	0	0	0		

Card 2	1	2	3	4	5	6	7	8
Variable	Axx	Axy	Axz	Bxx	Bxy	Bxz		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	0.0	0.0	0.0		

Card 2	1	2	3	4	5	6	7	8
Variable	Ayx	Ayy	Ayz	Byx	Byy	Byz		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	0.0	0.0	0.0		

Card 2	1	2	3	4	5	6	7	8
Variable	Azx	Azy	Azz	Bzx	Bzy	Bzz		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	0.0	0.0	0.0		

VARIABLE **DESCRIPTION**

EIDBEG, EIDEND

EIDBEG, EIDEND > 0: Range of thick porous element IDs. These are solids in 3D and shells in 2D.

EIDBEG, EIDEND < 0: Range of thin porous element IDs. These are shells in 3D and beams in 2D. The ALE option does *not* support thin porous elements.

EIDBEG > 0, EIDEND $= 0$: EIDBEG is a set of thick porous elements

EIDBEG > 0, EIDEND < 0: EIDBEG is a set of thin porous elements

LOCAL

Flag to activate an element coordinate system:

EQ.0: The forces are applied in the global directions.

EQ.1: The forces are applied in a local system attached to the element. The system is consistent with DIREC = 1 and CTYPE = 12 in *CONSTRAINED_LAGRANGE_IN_SOLID. For CTYPE = 11, LOCAL is always 1 and the x-axis is aligned with the element normal while the y-axis passes through the element center and the first node in the element connectivity (*ELEMENT_BEAM in 2D or *ELEMENT_SHELL in 3D)

VARIABLE	DESCRIPTION
VECID1, VECID2	*DEFINE_VECTOR IDs to define a specific coordinate system. VECID1 and VECID2 give the x- and y-direction respectively. The z-vector is a cross product of VECID1 and VECID2. If this latter is not orthogonal to VECID1, its direction will be corrected with a cross-product of z- and x-vectors. The vectors are stored as isoparametric locations to update their directions if the element deforms or rotates.
USERDEF	Flag to compute Aij and Bij with a user defined routine in the file dyn21.F called lagpor_getab_userdef. The file is part of the general package usermat.
Aij	Viscous matrix for the porous flow Ergun equation. (see Remark 1):
Bij	Inertial matrix for the porous flow Ergun equation. (see Remark 1):

Remarks:

1. **Ergun Equation.** The Ergun equation computing the pressure gradient along each direction $i = x, y, z$ can be written as follows:

$$\frac{dP}{dx_i} = \sum_{j=1}^{3} \left[\mu A_{ij} V_j + \rho B_{ij} \left| V_j \right| V_j \right]$$

Where,

a) V_i is the relative velocity of the flow in the porous media

b) A_{ij} are the viscous coefficients of the Ergun-type porous flow equation in the i^{th} direction.. This matrix is similar to the viscous coefficients used in *LOAD_BODY_POROUS.

c) B_{ij} are the inertial coefficient of the Ergun-type porous flow equation in the i^{th} direction. This matrix is similar to the inertial coefficients used in *LOAD_BODY_POROUS.

If this keyword defines the porous properties of Lagrangian elements in *CONSTRAINED_LAGRANGE_IN_SOLID, the porous coupling forces are computed with the pressure gradient as defined above instead of the equations used for CTYPE = 11 and 12.

*DEFINE_PRESSURE_TUBE

Purpose: Defines a closed gas filled tube for the simulation of interior pressure waves that result from changes in the tube cross section area over time. The tube is defined by tubular beam elements, and the gas volume is determined by beam cross section area and initial element lengths. Area changes are given by contact penetration from surrounding elements (only mortar contacts currently supported). The pressure calculation is not coupled with the deformation of the beam elements and does not use any data from the material card. Pressure and tube area at the beam nodes are output through *DATABASE_PRTUBE.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	WS	PR					
Type	I	F	F					
Default	0	0.0	0.0					

Optional card

Card 2	1	2	3	4	5	6	7	8
Variable	VISC	CFL	DAMP					
Type	F	F	F					
Default	1.0	0.9	0.0					

VARIABLE	DESCRIPTION
PID	Part ID of tube. All connected beam elements in the part will model a closed tube. Only tubular beam elements are allowed, i.e. ELFORM = 1,4,5,11 with CST = 1 on *SECTION_BEAM. Initial tube cross section area is calculated using the beam inner diameter TT1/TT2. If no inner diameter is given, the outer diameter TS1/TS2 is used. The beam elements may not contain junctions and two different parts where *DEFINE_PRESSURE_TUBE is applied may not share nodes. For MPP all elements in the part will be on a single processor.
WS	Speed of sound c_0 in the gas
PR	Initial gas pressure p_0 inside tube
VISC	Artificial viscosity multiplier $\boldsymbol{v}$, see remarks
CFL	CFL-factor $\boldsymbol{k}$, see remarks
DAMP	Linear damping $\boldsymbol{d}$, see remarks

Remarks:

The pressure tube is modeled with an acoustic approximation of the 1D compressible Euler equations for pipes with varying thickness

$$\frac{\partial}{\partial t}(\rho A) + \frac{\partial}{\partial x}(\rho u A) = 0,$$

$$\frac{\partial}{\partial t}(\rho u A) + \frac{\partial}{\partial x}(\rho u^2 A + pA) = p\frac{\partial A}{\partial x},$$

$$\frac{\partial}{\partial t}(EA) + \frac{\partial}{\partial x}(u(E + p)A) = 0,$$

where $A = A(x,t)$ is the cross section area and ρ, p, u, E is density, pressure, velocity, and energy per unit volume, respectively. The above system is closed under the constitutive relations

$$E = \rho e + \frac{\rho u^2}{2}, \qquad p = p(\rho, e),$$

where e is the internal energy per unit mass.

For an isentropic and isothermal flow, the pressure will be proportional to the density, i.e.

$$p = c_0^2 \rho,$$

and the energy equation can be dropped. This is a good approximation of the Euler equations for acoustic flows where the state variables are smooth perturbations around a background state. For such flows no shocks will develop over time but may be present from initial/boundary values or source terms.

Assuming small perturbations, linearization around $(\rho_0, p_0, u_0 = 0)$ gives the acoustic approximation

$$\frac{\partial}{\partial t}(A\rho) + \rho_0 \frac{\partial y}{\partial x} = 0,$$
$$\rho_0 \frac{\partial y}{\partial t} + c_0^2 \frac{\partial}{\partial x}(A\rho) = p\frac{\partial A}{\partial x},$$

where $y = Au$. Expressed in y and p we have

$$\frac{\partial p}{\partial t} + \frac{\partial \ln A}{\partial t} p + \frac{p_0}{A}\frac{\partial y}{\partial x} = 0,$$
$$\frac{\partial y}{\partial t} + A\frac{c_0^2}{p_0}\frac{\partial p}{\partial x} = 0.$$

This linearized system is solved using the standard Galerkin finite element method, using piecewise linear basis functions and artificial viscosity. Linear damping is added to model energy losses from friction between the gas and the tube walls. With artificial viscosity and linear damping, the system can be written as

$$\frac{\partial p}{\partial t} + \frac{\partial \ln A}{\partial t} p + \frac{p_0}{A}\frac{\partial y}{\partial x} = \epsilon\frac{\partial^2 p}{\partial x^2} - \boldsymbol{d}(p - p_0)$$
$$\frac{\partial y}{\partial t} + A\frac{c_0^2}{p_0}\frac{\partial p}{\partial x} = \epsilon\frac{\partial^2 y}{\partial x^2},$$

where the artificial viscosity is proportional to the maximum initial beam element length, i.e.

$$\epsilon = \boldsymbol{v} c_0 \max_i \Delta x_i,$$

Time integration is independent of the mechanical solver and uses a step size less than or equal to the global time step, satisfying a CFL condition

$$\Delta t < \min_i \frac{\boldsymbol{k}\Delta x_i}{\Delta x_i \left|\frac{\partial \ln A}{\partial t}\right| + 3c_0}.$$

*DEFINE_REGION

Purpose: Define a volume of space, optionally in a local coordinate system.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	TITLE						
Type	I	A70						

Card 2	1	2	3	4	5	6	7	8
Variable	TYPE	CID						
Type	I	I						

Card 3 for Rectangular Prism. Use when type = 0.

Card 3	1	2	3	4	5	6	7	8
Variable	XMN	XMX	YMN	YMX	ZMN	ZMX		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	0.0	0.0	0.0		

Card 3 for Sphere. Use when type = 1.

Card 3	1	2	3	4	5	6	7	8
Variable	XC1	YC1	ZC1	RMIN1	RMAX1			
Type	R	R	R	R	R			
Default	0.0	0.0	0.0	0.0	0.0			

Card 3 for Cylinder. Use when type = 2.

Card 3	1	2	3	4	5	6	7	8
Variable	XC2	YC2	ZC2	AX2	AY2	AZ2	RMIN2	RMAX2
Type	F	F	F	F	F	F	F	F
Default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Card 4 for Cylinder. Use when type = 2.

Card 4	1	2	3	4	5	6	7	8
Variable	L2							
Type	F							
Default	0.0							

Card 3 for Ellipsoid. Use when type = 3.

Card 3	1	2	3	4	5	6	7	8
Variable	XC3	YC3	ZC3	AX3	AY3	AZ3	BX3	BY3
Type	F	F	F	F	F	F	F	F
Default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Card 4 for Ellipsoid. Use when type = 3.

Card 4	1	2	3	4	5	6	7	8
Variable	BZ3	RA3	RB3	RC3				
Type	F	F	F	F				
Default	0.0	0.0	0.0	0.0				

VARIABLE	DESCRIPTION
ID	Region ID
TITLE	Title for this region
TYPE	Region type: EQ.0: Box EQ.1: Sphere or spherical shell EQ.2: Cylinder or cylindrical shell, infinite or finite in length EQ.3: Ellipsoid
CID	Optional local coordinate system ID. If given, all the following input parameters will be interpreted in this coordinate system.
XMN	Lower *x* limit of box
YMN	Lower *y* limit of box
ZMN	Lower *z* limit of box
XMX	Upper *x* limit of box
YMX	Upper *y* limit of box
ZMX	Upper *z* limit of box
XC1, YC1, ZC1	Coordinates of the center of the sphere
RMIN1, RMAX1	The inner and outer radii of the spherical shell. Set RMIN1 = 0 for a solid sphere
XC2, YC2, ZC2	A point on the cylindrical axis

VARIABLE	DESCRIPTION
AX2, AY2, AZ2	A vector which defines the direction of the axis of the cylinder
RMIN2, RMAX2	The inner and outer radii of the cylindrical shell. Set RMIN2 = 0 for a solid cylinder
L2	Length of the cylinder. If L2 = 0, and infinite cylinder is defined. Otherwise the cylinder has one end at the point (XC2, YC2, ZC2) and the other at a distance L2 along the axis in the direction of the vector (AX2, AY2, AZ2)
XC3, YC3, ZC3	Coordinates of the center of the ellipsoid
AX3, AY3, AZ3	A vector in the direction of the first axis of the ellipsoid (axis **a**)
BX3, BY3, BZ3	A vector, $\tilde{\mathbf{b}}$, in the plain of the first and second axes of the ellipsoid. The third axis of the ellipsoid (axis **c**) will be in the direction of $\mathbf{a} \times \tilde{\mathbf{b}}$ and finally the second axis $\mathbf{b} = \mathbf{c} \times \mathbf{a}$
RA3, RB3, RC3	The semi-axis lengths of the ellipsoid

*DEFINE_SD_ORIENTATION

Purpose: Define orientation vectors for discrete springs and dampers. These orientation vectors are optional for this element class. Four alternative options are possible. With the first two options, IOP = 0 or 1, the vector is defined by coordinates and is fixed permanently in space. The third and fourth option orients the vector based on the motion of two nodes, so that the direction can change as the line defined by the nodes rotates.

Card	1	2	3	4	5	6	7	8
Variable	VID	IOP	XT	YT	ZT	NID1	NID2	
Type	I	I	F	F	F	I	I	
Default	0	0	0.0	0.0	0.0	0	0	
Remarks	none	1	IOP = 0,1	IOP = 0,1	IOP = 0,1	IOP = 2,3	IOP = 2,3	

VARIABLE **DESCRIPTION**

VID Orientation vector ID. A unique ID number must be used.

IOP Option:

EQ.0: deflections/rotations are measured and forces/moments applied along the following orientation vector.

EQ.1: deflections/rotations are measured and forces/moments applied along the axis between the two spring/damper nodes projected onto the plane normal to the following orientation vector.

EQ.2: deflections/rotations are measured and forces/moments applied along a vector defined by the following two nodes.

EQ.3: deflections/rotations are measured and forces/moments applied along the axis between the two spring/damper nodes projected onto the plane normal to the a vector defined by the following two nodes.

XT x-value of orientation vector. Define if IOP = 0,1.

YT y-value of orientation vector. Define if IOP = 0,1.

VARIABLE	DESCRIPTION
ZT	z-value of orientation vector. Define if IOP = 0,1.
NID1	Node 1 ID. Define if IOP = 2,3.
NID2	Node 2 ID. Define if IOP = 2, 3.

Remarks:

1. The orientation vectors defined by options 0 and 1 are fixed in space for the duration of the simulation. Options 2 and 3 allow the orientation vector to change with the motion of the nodes. Generally, the nodes should be members of rigid bodies, but this is not mandatory. When using nodes of deformable parts to define the orientation vector, care must be taken to ensure that these nodes will not move past each other. If this happens, the direction of the orientation vector will immediately change with the result that initiate severe instabilities can develop.

*DEFINE_SET_ADAPTIVE

Purpose: To control the adaptive refinement level by element or part set.

Card 1	1	2	3	4	5	6	7	8
Variable	SETID	STYPE	ADPLVL	ADPSIZE				
Type	I	I	I	F				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
SETID	Element set ID or part set ID
STYPE	Set type for SETID: 1-element set 2-part set
ADPLVL	Adaptive refinement level for all elements in SETID set.
ADSIZE	Minimum element size to be adapted based on element edge length for all elements in SETID set.

Remarks:

1. This option is for 3D-shell h-adaptivity only at the present time.
2. The order of defining refinement level for any elements is *CONTROL_ADAPTIVE and *DEFINE_BOX_ADAPTIVE.
3. If there are multiple definitions of refinement level or element size for any elements, the latter one will be used.

*DEFINE_SPH_ACTIVE_REGION

Purpose: The purpose of this keyword is to increase the efficiency of the SPH method's neighborhood search algorithm by specifying an *active region*. All SPH elements located outside of the active region are deactivated. This card supports active regions consisting of the volume bounded by two closed surfaces (boxes, centered cylinders, and centered spheres are currently supported). Once the SPH particle is deactivated, it will stay inactive.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	TYPE	STYPE	CYCLE				
Type	I	I	I	I				
Default	none	0	0	1				

VARIABLE	DESCRIPTION
ID	Part Set ID/Part ID
TYPE	EQ.0: Part set EQ.1: Part
STYPE	Type of the region. EQ.0: Rectangular box EQ.1: Cylinder EQ.2: Sphere
CYCLE	Number of cycles between each check

Interior Rectangular Box Card. Card 2 format used for STYLE = 0.

Card 2	1	2	3	4	5	6	7	8
Variable	XIMIN	YIMIN	ZIMIN	XIMAX	YIMAX	ZIMAX		
Type	F	F	F	F	F	F		
Default	none	none	none	none	none	none		

Outer Rectangular Box Card. Card 3 format used for STYPE = 0.

Card 3	1	2	3	4	5	6	7	8
Variable	XOMIN	XOMIN	ZOMIN	XOMAX	YOMAX	ZOMAX		
Type	F	F	F	F	F	F		
Default	none	none	none	none	none	none		

VARIABLE	DESCRIPTION
XIMIN, YIMIN, ZIMIN	Minimum x, y, z coordinate of the inner box
XIMAX, YIMAX, ZIMAX	Maximum x, y, z coordinates of the inner box
XOMIN, YOMIN, ZOMIN	Minimum x, y, z coordinate of the outer box
XOMAX, YOMAX, ZOMAX	Maximum x, y, z coordinates of the outer box

Cylinder Axis Card. Card 2 format used for STYPE = 1. (see Remark 1)

Card 2	1	2	3	4	5	6	7	8
Variable	X0	Y0	Z0	XH	YH	ZH		
Type	F	F	F	F	F	F		
Default	none	none	none	none	none	none		

Cylinder Radii Card. Card 3 format used for STYPE = 1.

Card 3	1	2	3	4	5	6	7	8
Variable	RMIN	ZMIN	RMAX	ZMAX				
Type	F	F	F	F				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
X0, Y0, Z0	Coordinates of the center of the cylinder base. The nested cylinders are sharing the same starting base plane. This point also serves as the tail for the vector specifying the direction of the cylinders' axis. (See Remark 1)
XH, YH, ZH	Coordinates for the head of the cylinders axial direction vector.
RMIN, ZMIN	Radius and length of the interior cylinder.
RMAX, ZMAX	Radius and length of the outer cylinder.

Center of Sphere Card. Card 2 used for STYPE = 2.

Card 2	1	2	3	4	5	6	7	8
Variable	X0	Y0	Z0					
Type	F	F	F					
Default	none	none	none					

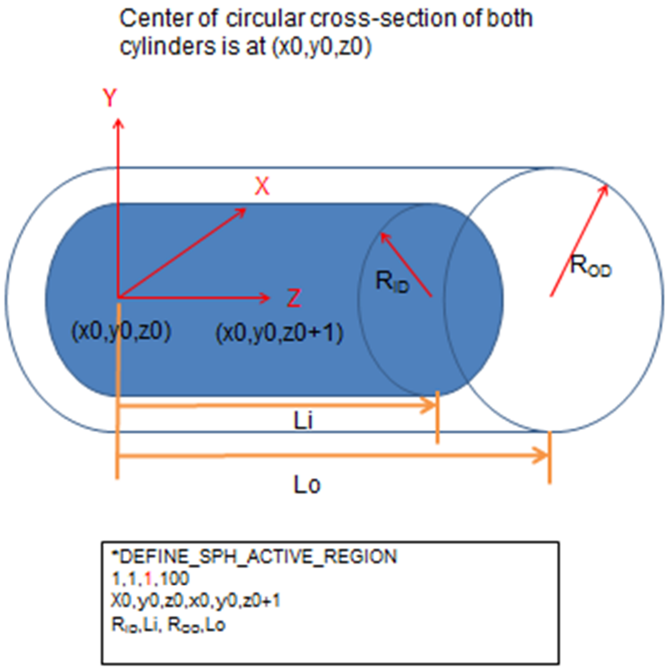
Sphere Radii Card. Card 3 used for STYPE = 2.

Card 3	1	2	3	4	5	6	7	8
Variable	RMIN	RMAX						
Type	F	F						
Default	none	none						

VARIABLE	DESCRIPTION
X0, Y0, Z0	The spheres' center.
RMIN	Radius of the interior sphere
RMAX	Radius of the outer sphere

Remarks:

1.Cylindrical system for SPH active region (see figure 15-46)


Center of circular cross-section of both cylinders is at (x0,y0,z0)
Y
X
Z
R_{ID}
R_{OD}
(x0,y0,z0)
(x0,y0,z0+1)
Li
Lo

```
*DEFINE_SPH_ACTIVE_REGION
1,1,1,100
X0,y0,z0,x0,y0,z0+1
RID,Li, ROD,Lo
```

Figure 15-46. Example DEFINE_SPH_ACTIVE_REGION

DEFINE_SPH_DE_COUPLING_*{OPTION}

Purpose: Define a penalty based contact. This option is to be used for the node to node contacts to couple SPH solver and discrete element sphere (DES) solver.

The available options include:

<BLANK>

ID

ID Card. Additional card for ID keyword option.

Optional	1	2	3	4	5	6	7	8
Variable	DID	HEADING						
Type	I	A80						
Default	none	none						

SPH Part Cards. Provide as many as necessary. Input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	SPHID	DESID	SPHTYP	DESTYP	PFACT	DFACT	SPHBOX	
Type	I	I	I	I	F	F	I	
Default	none	none	none	none	1.0	0.	none	

VARIABLE	DESCRIPTION
DID	Definition ID. This must be a unique number.
HEADING	Definition descriptor. It is suggested that unique descriptions be used.
SPHID	SPH part or part set ID.
DESID	DES part or part set ID.

SPHTYP SPH part type:

EQ.0: Part set ID,

EQ.1: Part ID.

DESTYP DES part type:

EQ.0: Part set ID,

EQ.1: Part ID.

PFACT Penalty scale factor

DFACT Penalty scale factor for contact damping coefficient

SPHBOX BOX ID for SPH parts, See Remark 1.

Remarks:

SPHBOX is used to define the box IDs for the SPH parts. Only the particles that inside the boxes are defined for the node to node contacts.

*DEFINE_SPH_INJECTION

Purpose: This keyword injects SPH elements from user defined grid points.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	NSID	CID	VX	VY	VZ	AREA	
Type	I	I	I	F	F	F	F	
Default	none	none	None	0.0	0.0	0.0	0.0	

Card 2	1	2	3	4	5	6	7	8
Variable	TBEG	TEND						
Type	F	F						
Default	0.0	1.0E20						

VARIABLE	DESCRIPTION
PID	Part ID of newly generated SPH elements.
NSID	Node set ID. Nodes are used for initial injection position for the SPH elements.
CID	Local coordinate system ID, see *DEFINE_COORDINATE_SYSTEM. X and Y coordinates define the injection plane, Z coordinate defines the normal to the injection plane.
VX, VY, VZ	Velocity of the inject elements: $\mathbf{v} = (\mathrm{VX}, \mathrm{VY}, \mathrm{VZ})$
AREA	The area of initial injection surface. The density of injection flow comes from the material models see *MAT definition.
TBEG	Birth time
TEND	Death time

DEFINE_SPH_TO_SPH_COUPLING_*{OPTION}

Purpose: Define a penalty based contact. This option is to be used for the node to node contacts between SPH parts.

The available options include:

<BLANK>

ID

ID Cards. Additional card for ID keyword option.

Optional	1	2	3	4	5	6	7	8
Variable	DID	HEADING						
Type	I	A70						
Default	none	none						

Sets of coupling cards:

Each set consists of a Card 1 and may include an additional Card 2. Unless the card following Card 1 contains an "&" in its first column, the optional card is not read. Provide as many sets as necessary. This input terminates at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	SSID	MSID	SSTYP	MSTYP	IBOX1	IBOX2	PFACT	SRAD
Type	I	I	I	I	I	I	F	F
Default	none	none	none	none	none	none	1.0	1.0

Optional. The keyword reader identifies this card by an "&" in the first column.

Card 2	1	2	3	4	5	6	7	8
Variable	DFACT	ISOFT						
Type	F	I						
Default	0.0	0						

VARIABLE	DESCRIPTION
DID	Definition ID. This must be a unique number.
HEADING	Definition descriptor. It is suggested that unique descriptions be used.
SSID	Slave part or part set ID.
MSID	Master part or part set ID.
SSTYP	Slave part type: EQ.0: Part set ID, EQ.1: Part ID.
MSTYP	Master part type: EQ.0: Part set ID, EQ.1: Part ID.
IBOX1	Box ID for slave parts, See Remark 1.
IBOX2	Box ID for master parts, See Remark 1.
PFACT	Penalty scale factor, See Remark 2.
SRAD	Scale factor for nodes to nodes contact criteria, See Remark 3.
DFACT	Penalty scale factor for contact damping coefficient, See Remark 4.

ISOFT	Soft constraint option: EQ. 0: penalty formulation EQ. 1: soft constraint formulation The soft constraint may be necessary if the material constants of the parts in contact have a wide variation in the elastic bulk moduli. In the soft constraint option, the interface stiffness is based on the nodal mass and the global time step size.

Remarks:

1. IBOX1 and IBOX2 are used to define the box IDs for the slave parts and the master parts respectively. Only the particles that inside the boxes are defined for the node to node contacts.

2. For High Velocity Impact problems, a smaller value (ranges from 0.01 to 1.0e-4) of PFACT variable is recommended. A number ranges from 0.1 to 1 is recommended for low velocity contact between two SPH parts.

3. Contact between two SPH particles from different parts is detected when the distance of two SPH particles is less than SRAD*(sum of smooth lengths from two particles)/2.0.

4. DFACT = 0.0 is the default and is recommended. For DFACT > 0.0, interaction between SPH parts includes a viscous effect, providing some stickiness similar to the particle approximation invoked when CONT = 0 in *CONTROL_SPH. At present, no recommendation can be given for a value of DFACT other than the value should be less than 1.0.

.

DEFINE_SPOTWELD_FAILURE_*{OPTION}

The available options are

<BLANK>

ADD

Purpose: Define spot weld failure data for the failure criterion developed by Lee and Balur (2011). This is OPT = 10 on *MAT_SPOTWELD. It is available for spot welds consisting of beam elements, solid elements, or solid assemblies. Furthermore, *DEFINE_SPOTWELD_FAILURE requires that the weld nodes be tied to shell elements using tied constraint based contact options: For beam element welds, only *CONTACT_SPOTWELD is valid. For solid element welds or solid assembly welds, valid options are the following.

*CONTACT_TIED_SURFACE_TO_SURFACE

*CONTACT_SPOTWELD

*CONTACT_TIED_SHELL_EDGE_TO_SURFACE

Other tied contact types cannot be used.

The ADD keyword option adds materials to a previously defined spot weld failure data set.

Data Card 1. This card contains the data set's ID and the first 7 parameters. When the ADD option is active *leave the 7 parameters blank*.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	TFLAG	DC1	DC2	DC3	DC4	EXN	EXS
Type	I	I	F	F	F	F	F	F
Default	none	0	1.183	0.002963	0.0458	0.1	1.51	1.51

Data Card 2. . This card contains 3 spot weld failure data parameters. *Do not include this card when the ADD option is active*

Card 2	1	2	3	4	5	6	7	8
Variable	NAVG	D_SN	D_SS	R_SULT				
Type	I	F	F	F				
Default	0	none	none	0.0				

Material-Specific Strength Data Cards. Include one card for each material associated with the data set. The next keyword ("*") card terminates the keyword.

Card 3	1	2	3	4	5	6	7	8
Variable	MID	SN	SS					
Type	I	F	F					
Default	none	none	none					

VARIABLE	DESCRIPTION
ID	Identification number of data set, input as FVAL on *MAT_-SPOTWELD
TFLAG	Thickness flag for nominal stress calculation EQ.0: Use minimum thickness EQ.1: Use average thickness
DC1	Dynamic coefficient, c_1
DC2	Dynamic coefficient, c_2
DC3	Dynamic coefficient, c_3
DC4	Dynamic coefficient, c_4
EXN	Exponent on the normal term, n_n
EXS	Exponent on the shear term, n_s

VARIABLE	DESCRIPTION
NAVG	Number of points in the time average of the load rates
D_SN	Default value of the static normal strength, $S_{n,\mathrm{stat}}$
D_SS	Default value of the static shear strength, $S_{s,\mathrm{stat}}$
R_SULT	Reference ultimate strength (see remark below)
MID	Material ID number of welded shell material
SN	Static normal strength of material MID. $S_{n,\mathrm{stat}}$
SS	Static shear strength of material MID, $S_{s,\mathrm{stat}}$

Remarks:

This stress based failure model, which was developed by Lee and Balur (2011), uses nominal stress in the numerator and dynamical strengths in the denominator. The weld fails when the stresses are outside of the failure surface defined as

$$\left(\frac{s_n}{S_{n,\mathrm{dyn}}}\right)^{n_n} + \left(\frac{s_s}{\mathrm{S}_{s,\mathrm{dyn}}}\right)^{n_s} = 1$$

where s_n and s_s are nominal stress in the normal and tangential directions such that

$$s_n = \frac{P_n}{Dt}$$

$$s_s = \frac{P_s}{Dt}.$$

P_n and P_s are the loads carried by the weld in the normal and tangential directions, $D$is the weld diameter, and t is the thickness of the welded sheets. If the sheets have different thicknesses, then TFLAG controls whether the minimum or average thickness is used. The dynamical strength terms in the denominator are load-rate dependent and are derived from static strength:

$$S_{n,\mathrm{dyn}} = S_{n,\mathrm{stat}} \left[c_1 + c_2 \left(\frac{\dot{P}_n}{c_4}\right) + c_3 \log\left(\frac{\dot{P}_n}{c_4}\right)\right]$$

$$S_{s,\mathrm{dyn}} = S_{s,\mathrm{stat}} \left[c_1 + c_2 \left(\frac{\dot{P}_s}{c_4}\right) + c_3 \log\left(\frac{\dot{P}_s}{c_4}\right)\right]$$

where the constants c_1 to c_4 are the input in the fields DC1 to DC4, $\dot{P}_n$ and $\dot{P}_s$ are the load rates, and $S_{n,\mathrm{stat}}$ and $S_{s,\mathrm{stat}}$ are the static strengths of the welded sheet materials which for each material are input using SN and SS.

When two different materials are welded, the material having the smaller normal strength determines the strengths used for the weld. Materials that do not have SN and SS values default to D_SN and D_SS from card 1. The default values for DC1 to DC4, and EXN and EXS are based on the work Chao, Wang, Miller and Zhu (2010). and Wang, Chao, Zhu, and Miller (2010). These parameters are unitless except for DC4 which has units of force per unit time. The default value of 0.1 is for MN/sec.

The load rate, $\dot{P}$, can be time averaged to reduce the effect of high frequency oscillations on the dynamic weld strength. NAVG is the number of terms in the time average.

If R_SULT is defined on Card 2 and the PID keyword option is not used, then D_SN and D_SS are interpreted to be reference values of the normal and shear static strength, and the SN field on Card 3 is interpreted as a material specific ultimate strength. These values are then use to calculate material specific strength values by

$$S_{n,stat} = S_{n,ref}\left(\frac{S_u}{S_{u,ref}}\right)$$

$$S_{s,stat} = S_{s,ref}\left(\frac{S_u}{S_{u,ref}}\right)$$

where $S_{n,ref}$, $S_{s,ref}$, and $S_{u,ref}$, are D_SN, D_SS, and R_SULT on card 2, and S_u is SN on card 3. With this option, the SS values are ignored. If the PID keyword option is used, then R_SULT is ignored and SN and SS are the static strength values.

*DEFINE_SPOTWELD_FAILURE_RESULTANTS

Purpose: Define failure criteria between part pairs for predicting spot weld failure. This table is implemented for *solid* element spot welds, which are used with the tied, constraint based, contact option: *CONTACT_TIED_SURFACE_TO_SURFACE. *Note that other tied contact types cannot be used.* The input in this section continues until then next "*" card is encountered. Default values are used for any part ID pair that is not defined. Only one table can defined. See *MAT_SPOTWELD where this option is used whenever *OPT* = 7.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	DSN	DSS	DLCIDSN	DLCIDSS			
Type	I	F	F	I	I			
Default	0	0.0	0.0	0	0			

Failure Cards. Provide as many as necessary. The next keyword ("*") card terminates the table definition.

Card 2…	1	2	3	4	5	6	7	8
Variable	PID_I	PID_J	SNIJ	SSIJ	LCIDSNIJ	LCIDSSIJ		
Type	I	I	F	F	I	I		
Default	none	none	0.0	0.0	0	0		

VARIABLE	DESCRIPTION
ID	Identification number. Only one table is allowed.
DSN	Default value of the normal static stress at failure.
DSS	Default value of the transverse static stress at failure.
DLCIDSN	Load curve ID defining a scale factor for the normal stress as a function of strain rate. This factor multiplies DSN to obtain the failure value at a given strain rate.

VARIABLE	DESCRIPTION
DLCIDSS	Load curve ID defining a scale factor for static shear stress as a function of strain rate. This factor multiplies DSN to obtain the failure value at a given strain rate.
PID_I	Part ID I.
PID_J	Part ID J.
SNIJ	The maximum axial stress at failure between parts I and J. The axial stress is computed from the solid element stress resultants, which are based on the nodal point forces of the solid element.
DSSIJ	The maximum shear stress at failure between parts I and J. The shear stress is computed from the solid element stress resultants, which are based on the nodal point forces of the solid element.
LCIDSNIJ	Load curve ID defining a scale factor for the normal stress as a function of strain rate. This factor multiplies SNIJ to obtain the failure value at a given strain rate.
LCIDSSIJ	Load curve ID defining a scale factor for static shear stress as a function of strain rate. This factor multiplies SSIJ to obtain the failure value at a given strain rate.

Remarks:

The stress based failure model, which was developed by *Toyota Motor Corporation,* is a function of the peak axial and transverse shear stresses. The entire weld fails if the stresses are outside of the failure surface defined by:

$$\left(\frac{\sigma_{rr}}{\sigma_{rr}^F}\right)^2 + \left(\frac{\tau}{\tau^F}\right)^2 - 1 = 0$$

where σ_{rr}^F and τ^F are specified in the above table by part ID pairs. LS-DYNA automatically identifies the part ID of the attached shell element for each node of the spot weld solid and checks for failure. If failure is detected the solid element is deleted from the calculation.

If the effects of strain rate are considered, then the failure criteria becomes:

$$\left[\frac{\sigma_{rr}}{f_{dsn}(\dot{\varepsilon}^p)\sigma_{rr}^F}\right]^2 + \left[\frac{\tau}{f_{dss}(\dot{\varepsilon}^p)\tau^F}\right]^2 - 1 = 0$$

*DEFINE_SPOTWELD_MULTISCALE

Purpose: Associate beam sets with multi-scale spot weld types for modeling spot weld failure via the multi-scale spot weld method.

Spot Weld/Beam Set Association Cards. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	TYPE	BSET	TYPE	BSET	TYPE	BSET	TYPE	BSET
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
TYPE	MULTISCALE spot weld type to use. See *INCLUDE_-SPOTWELD_MULTISCALE
BSET	Beam set which uses this multi-scale spot weld type for failure modeling.

Remarks:

See *INCLUDE_MULTISCALE_SPOTWELD for a detailed explanation of this capability.

*DEFINE_SPOTWELD_RUPTURE_PARAMETER

Purpose: Define a parameter by part ID for shell elements attached to spot weld *beam* elements using the constrained contact option: *CONTACT_SPOTWELD. *This table will not work with other contact types*. Only one table is permitted in the problem definition. Data, which is defined in this table, is used by the stress based spot weld failure model developed by *Toyota Motor Corporation*. See *MAT_SPOTWELD where this option is activated by setting the parameter *OPT* to a value of 9. This spot weld failure model is a development of *Toyota Motor Corporation*.

Card 1	1	2	3	4	5	6	7	8
Variable	PID							
Type	I							
Default								

Card 2	1	2	3	4	5	6	7	8
Variable	C11	C12	C13	N11	N12	N13		SIG_PF
Type	F	F	F	F	F	F		F
Default								

Card 3	1	2	3	4	5	6	7	8
Variable	C21	C22	C23	N2				SIG_NF
Type	F	F	F	F				
Default								

Card 4	1	2	3	4	5	6	7	8
Variable	LCDPA	LCDPM	LCDPS	LCDNA	LCDNM	LCDNS		NSMT
Type	I	I	I	I	I	I		I
Default	0	0	0	0	0	0		0

VARIABLE	DESCRIPTION
PID	Part ID for the attached shell.
C11-N2	Parameters for model, see Remarks below.
SIG_PF	Nugget pull-out stress, σ_P.
SIG_NF	Nugget fracture stress, σ_F.
LCDPA	Curve ID defining dynamic scale factor of spot weld axial load rate for nugget pull-out mode.
LCDPM	Curve ID defining dynamic scale factor of spot weld moment load rate for nugget pull-out mode.
LCDPS	Curve ID defining dynamic scale factor of spot weld shear load rate for nugget pull-out mode.
LCDNA	Curve ID defining dynamic scale factor of spot weld axial load rate for nugget fracture mode.
LCDNM	Curve ID defining dynamic scale factor of spot weld moment load rate for nugget fracture mode.
LCDNS	Curve ID defining dynamic scale factor of spot weld shear load rate for nugget fracture mode.
NSMT	The number of time steps used for averaging the resultant rates for the dynamic scale factors.

Remarks:

This failure model incorporates two failure functions, one for nugget pull-out and the other for nugget fracture. The nugget pull-out failure function is

$$F_p = \frac{\mathrm{C11}\times\dfrac{A}{D^{\mathrm{N11}}} + \mathrm{C12}\times\dfrac{M}{D^{\mathrm{N12}}} + \mathrm{C13}\times\dfrac{S}{D^{\mathrm{N13}}}}{\sigma_P\left[1+\left(\dfrac{\dot{\varepsilon}^p}{C}\right)^{1/p}\right]}$$

where *A*, *M*, and *S* are the axial force, moment, and shear resultants respectively, *D* is the spot weld diameter, and the Cowper-Symonds coefficients are from the attached shell material model. If the Cowper-Symonds coefficients aren't specified, the term within the square brackets, [ ], is 1.0. The fracture failure function is

$$F_n = \frac{\sqrt{(\mathrm{C21}\times A + \mathrm{C22}\times M)^2 + 3(\mathrm{C23}\times S)^2}}{D^{\mathrm{N2}}\sigma_F\left[1+\left(\dfrac{\dot{\varepsilon}^p}{C}\right)^{1/p}\right]}.$$

When the load curves for the rate effects are specified, the failure criteria are

$$F_p = \frac{\mathrm{C11}\times f_{\mathrm{dpa}}(\dot{A})\times\dfrac{A}{D^{\mathrm{N11}}} + C12\times f_{\mathrm{dpa}}(\dot{M})\times\dfrac{M}{D^{\mathrm{N12}}} + \mathrm{C13}\times f_{\mathrm{dpa}}(\dot{S})\times\dfrac{S}{D^{\mathrm{N13}}}}{\sigma_P}$$

$$F_n = \frac{\sqrt{\left[\mathrm{C21}\times f_{\mathrm{dna}}(\dot{A})\times A + \mathrm{C22}\times f_{\mathrm{dnm}}(\dot{M})\times M\right]^2 + 3\left[\mathrm{C23}\times f_{dns}(\dot{S})\times S\right]^2}}{D^{\mathrm{N2}}\sigma_F}$$

where *f* is the appropriate load curve scale factor. The scale factor for each term is set to 1.0 for when no load curve is specified. No extrapolation is performed if the rates fall outside of the range specified in the load curve to avoid negative scale factors. A negative load curve ID designates that the curve abscissa is the $\log_{10}$ of the resultant rate. This option is recommended when the curve data covers several orders of magnitude in the resultant rate. Note that the load curve dynamic scaling replaces the Cowper-Symonds model for rate effects.

Failure occurs when either of the failure functions is greater than 1.0.

*DEFINE_SPOTWELD_RUPTURE_STRESS

Purpose: Define a static stress rupture table by part ID for shell elements connected to spot weld *beam* elements using the constrained contact option: *CONTACT_SPOTWELD. *This table will not work with other contact types*. Only one table is permitted in the problem definition. Data, which is defined in this table, is used by the stress based spot weld failure model developed by *Toyota Motor Corporation*. See *MAT_SPOTWELD where this option is activated by setting the parameter *OPT* to a value of 6.

Part Cards. Define rupture stresses part by part. The next keyword ("*") card terminates this input.

Card	1	2	3	4	5	6	7	8
Variable	PID	SRSIG	SIGTAU	ALPHA				
Type	I	F	F	F				

VARIABLE	DESCRIPTION
PID	Part ID for the attached shell.
SRSIG	Axial (normal) rupture stress, σ_{rr}^F.
SRTAU	Transverse (shear) rupture stress, τ^F.
ALPHA	Scaling factor for the axial stress as defined by Toyota. The default value is 1.0.

Remarks:

The stress based failure model, which was developed by *Toyota Motor Corporation,* is a function of the peak axial and transverse shear stresses. The entire weld fails if the stresses are outside of the failure surface defined by:

$$\left(\frac{\sigma_{rr}}{\sigma_{rr}^F}\right)^2 + \left(\frac{\tau}{\tau^F}\right)^2 - 1 = 0$$

where σ_{rr}^F and τ^F are specified in the above table by part ID. LS-DYNA automatically identifies the part ID of the attached shell element for each node of the spot weld beam and independently checks each end for failure. If failure is detected in the end attached to the shell with the greatest plastic strain, the beam element is deleted from the calculation.

If the effects of strain rate are considered, then the failure criteria becomes:

$$\left[\frac{\sigma_{rr}}{\sigma_{rr}^F(\dot{\varepsilon}^p)}\right]^2 + \left[\frac{\tau}{\tau^F(\dot{\varepsilon}^p)}\right]^2 - 1 = 0$$

where $\sigma_{rr}^F(\dot{\varepsilon}^p)$ and $\tau^F(\dot{\varepsilon}^p)$ are found by using the Cowper and Symonds model which scales the static failure stresses:

$$\sigma_{rr}^F(\dot{\varepsilon}^p) = \sigma_{rr}^F\left[1 + \left(\frac{\dot{\varepsilon}^p}{C}\right)^{1/p}\right]$$

$$\tau^F(\dot{\varepsilon}^p) = \tau^F\left[1 + \left(\frac{\dot{\varepsilon}^p}{C}\right)^{1/p}\right]$$

where $\dot{\varepsilon}^p$is the average plastic strain rate which is integrated over the domain of the attached shell element, and the constants p and C are uniquely defined at each end of the beam element by the constitutive data of the attached shell. The constitutive model is described in the material section under keyword: *MAT_PIECEWISE_LINEAR_-PLASTICITY.

The peak stresses are calculated from the resultants using simple beam theory.

$$\sigma_{rr} = \frac{N_{rr}}{A} + \frac{\sqrt{M_{rs}^2 + M_{rt}^2}}{\alpha Z} \quad \tau = \frac{M_{rr}}{2Z} + \frac{\sqrt{N_{rs}^2 + N_{rt}^2}}{A}$$

where the area and section modulus are given by:

$$A = \pi\frac{d^2}{4}$$
$$Z = \pi\frac{d^3}{32}$$

and d is the diameter of the spot weld beam.

DEFINE_STAGED_CONSTRUCTION_PART_*{OPTION}

Available options include:

<BLANK>

SET

Purpose: Staged construction. This keyword offers a simple way to define parts that are removed (e.g., during excavation), added (e.g., new construction) and used temporarily (e.g., props) during the analysis. Available for solid, shell, and beam element parts.

Part Cards. Provide as many as necessary. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID/PSID	STGA	STGR					
Type	I	I	I					
Default	none	See Remarks	See Remarks					

VARIABLE	DESCRIPTION
PID	Part ID (or Part Set ID for the_SET option)
STGA	Construction stage at which part is added
STGR	Construction stage at which part is removed

Remarks:

Used with *DEFINE_CONSTRUCTION_STAGES (defines the meaning of stages STGA and STGR) and *CONTROL_STAGED_CONSTRUCTION. If STGA = 0, the part is present at the start of the analysis. If STGR = 0, the part is still present at the end of the analysis. Examples:

1. Soil that is excavated would have STGA = 0 but STGR > 0

2. New construction would have STGA > 0 and STGR = 0

3. Temporary works would have STGA > 0, STGR > STGA.

This is a convenience feature that reduces the amount of input data needed for many typical construction models. Internally, LS-DYNA checks for *LOAD_REMOVE_PART, *LOAD_GRAVITY_PART and *LOAD_STIFFEN_PART referencing the same PID. Generally, these will not be present and LS-DYNA creates the data using STGA and STGR, and default gravity and pre-construction stiffness factor from *CONTROL_-STAGED_CONSTRUCTION. If existing cards are found, STGA and STGR are inserted into the existing data. During the analysis, any load curves entered on those existing cards will override STGA and STGR.

DEFINE_STOCHASTIC_ELEMENT_*OPTION

Options:

SOLID_VARIATON for solid elements.

SHELL_VARIATION for shell elements.

Purpose: Define the stochastic variation in the yield stress, damage/failure models, density, and elastic moduli for solid material models with the STOCHASTIC option, currently materials 10, 15, 24, 81, and 98. This option overrides values assigned by *DEFINE_STOCHASTIC_VARIATION.

Card 1	1	2	3	4	5	6	7	8
Variable	IDE	VARSY	VARF	VARRO	VARE			
Type	I	F	F	F	F			
Default	0	0	0	0	0			

VARIABLE	DESCRIPTION
IDE	Element ID
VARSY	The yield stress and its hardening function are scaled by 1.+VARSY.
VARF	The failure criterion is scaled by 1+VARF
VARRO	The density is scaled by 1+VARRO. This is intended to be used with topology optimization. This option is not available for shell elements.
VARE	The elastic moduli are scaled by 1+VARE. This is intended to be used with topology optimization.

*DEFINE_STOCHASTIC_VARIATION

Purpose: Define the stochastic variation in the yield stress and damage/failure models for material models with the STOCHASTIC option, currently materials 10, 15, 24, 81, and 98 and the shell version of material 123.

Card 1	1	2	3	4	5	6	7	8
Variable	ID_SV	PID	PID_TYP	ICOR	VAR_S	VAR_F	IRNG	
Type	I	I	I	I	I	I	I	
Default	0	0	0	0	0	0	0	

Yield Stress Card for Built-in Distribution. Card 2 for VAR_S set to 0, 1, or 2.

Card 2	1	2	3	4	5	6	7	8
Variable	R1	R2	R3					
Type	F	F	F					
Default								

Yield Stress Card for Load Curve. Card 2 for VAR_S set to 3 or 4.

Card 2	1	2	3	4	5	6	7	8
Variable	LCID							
Type	I							
Default	0							

Failure Strain Card for Built-in Distribution. Card 2 for VAR_F set to 0, 1, or 2.

Card 2	1	2	3	4	5	6	7	8
Variable	R1	R2	R3					
Type	F	F	F					
Default								

Failure Strain Card for Load Curve. Card 2 for VAR_F set to 3 or 4.

Card 2	1	2	3	4	5	6	7	8
Variable	LCID							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
ID_SV	Stochastic variation ID. A unique ID number must be used.
PID	*PART ID or *SET_PART ID.
PID_TYP	Flag for PID type. If PID and PID_TYP are both 0, then the properties defined here apply to all shell and solid parts using materials with the STOCHASTIC option. EQ.0: PID is a *PART ID. EQ.1: PID is a *SET_PART ID
ICOR	Correlation between the yield stress and failure strain scaling. EQ.0: Perfect correlation. EQ.1: No correlation. The yield stress and failure strain are independently scaled.

VARIABLE	DESCRIPTION
VAR_S	Variation type for scaling the yield stress. EQ.0: The scale factor is 1.0 everywhere. EQ.1: The scale factor is random number in the uniform random distribution in the interval defined by R1 and R2. EQ.2: The scale factor is a random number obeying the Gaussian distribution defined by R1, R2, and R3. EQ.3: The scale factor is defined by the probability distribution function defined by curve LCID. EQ.4: The scale factor is defined by the cumulative distribution function defined by curve LCID.
VAR_F	Variation type for scaling failure strain. EQ.0: The scale factor is 1.0 everywhere. EQ.1: The scale factor is random number in the uniform random distribution in the interval defined by R1 and R2. EQ.2: The scale factor is a random number obeying the Gaussian distribution defined by R1, R2, and R3. EQ.3: The scale factor is defined by the probability distribution function defined by curve LCID. EQ.4: The scale factor is defined by the cumulative distribution function defined by curve LCID.
IRNG	Flag for random number generation. EQ.0: Use deterministic (pseudo-) random number generator. The same input always leads to the same distribution. EQ.1: Use non-deterministic (true) random number generator. With the same input, a different distribution is achieved in each run.
R1, R2, R3	Real values to define the stochastic distribution. See below.
LCID	Curve ID defining the stochastic distribution. See below.

Remarks:

Each integration point x_g in the parts specifed by PID is assigned the random scale factors R_S and R_F that are applied to the values calculated by the material model for the yield stress and failure strain.

$$\sigma_y = R_s(x_g)\sigma_y(\bar{\varepsilon}^p, \dots)$$

$$\bar{\varepsilon}^p_{\text{FAIL}} = R_F(x_g)\bar{\varepsilon}^p_{\text{FAIL}}(\dot{\varepsilon}, \bar{\varepsilon}^p, \dots)$$

The scale factors vary spatially over the model according to the chosen statistical distributions defined in this section and are independent of time. The scale factors may be completely correlated or uncorrelated with the default being completely correlated since the failure strain is generally reduced as the yield stress increases. The scale factors R_S and R_F may be stored as extra history variables as follows:

Material Model	History Variable # for R_S	History Variable # for R_F
10	5	6
15	7	8
24	6	7
81	6	7
98	7	8
123 (shells only)	6	7

The user is responsible for defining the distributions so that they are physically meaningful and are restricted to a realistic range. Since neither the yield stress nor the failure strain may be negative, for example, the minimum values of the distributions must always be greater than zero.

The probability that a particular value R will occur defines the *probability distribution function*, $P(R)$. Since a value must be chosen from the distribution, the integral from the minimum to the maximum value of R of the probability distribution function must be 1.0,

$$\int_{R_{\text{MIN}}}^{R_{\text{MAX}}} P(R)dR = 1.$$

Another way to characterize a distribution is the *cumulative distribution function* $C(R)$ which defines the probability that a value will lie between R_{MIN} and R,

$$C(R) = \int_{R_{\text{MIN}}}^{R} P(\hat{R})d\hat{R}.$$

By definition $C(R_{MIN}) = 0$ and $C(R_{MAX}) = 1$. An inverse cumulative probability function D gives the number for a cumulative probability of $C(R)$,

$$D\big(C(R)\big) = R.$$

A random varriable satisfying the probability distribution function *P(R)* can be generated from a sequence of uniformly distributed numbers, $\hat{R}_I$, for $I = 1, N$, using the inverse cumulative distribution function D as

$$R_I = D(\hat{R}_I).$$

The scale factors for the yield stress and the failure strain may be generated using the same value of $\hat{R}_I$ for both (ICOR = 0) or by using independent values each one (ICOR = 1). If the same values are used, there is perfect correlation, and the failure strain scale factor becomes an implicit value of yield stress scale factor.

VAR = 0. No Scaling.

The corresponding yield stress or failure strain is not scaled.

VAR = 1. Scaling from Uniform Distribution

A uniform distribution is specified by setting VAR = 1. The input variable R1 is interpreted as R_{MIN} and R2 as R_{MAX}. If R1 = R2, then the yield stress or failure strain will be scaled by R1.

When using the uniform random distribution, the probability of a particular value is given by

$$P(R) = \frac{1}{R_{\mathrm{MAX}} - R_{\mathrm{MIN}}}$$

and the cumulative probability function is given by

$$C(R) = \frac{R - R_{\mathrm{MIN}}}{R_{\mathrm{MAX}} - R_{\mathrm{MIN}}}.$$

VAR = 2. Gaussian Distribution

The Gaussian distribution, VAR = 2, is smoothly varying with a peak at μ and 63 percent of the values occurring within the interval of one standard deviation σ, $[\mu - \sigma, \mu + \sigma]$. The input parameter R1 is interpreted as the mean, μ,, while R2 is interpreted as the standard deviation, σ. There is a finite probability that the values of R will be outside of the range that are physically meaningful in the scaling process, and R3 which is interpreted as δ restricts the range of R to $[\mu - \delta, \mu + \delta]$ The resulting truncated Gaussian distribution is rescaled such that,

$$D(\mu + \delta) = 1.$$

VAR = 3 or 4. Distribution from a Load Curve

The user may directly specify the probability distribution function or the cumulative probability distribution function with *DEFINE_CURVE by setting VAR = 3 or VAR = 4, respectively, and then specifying the required curve ID on the next data card.

Stochastic variations may be used simultaneously with the heat affected zone (HAZ) options in LS-DYNA (see *DEFINE_HAZ_PROPERTIES). The effect of the scale factors from stochastic variation and HAZ options are multiplied together to scale the yield stress and failure strain,

$$\sigma_y = R_S(x_g) R_S^{\mathrm{HAZ}} \sigma_y(\bar{\varepsilon}^p, \dots)$$

$$\bar{\varepsilon}^p_{\mathrm{FAIL}} = R_F(x_g) R_F^{\mathrm{HAZ}} \bar{\varepsilon}^p_{\mathrm{FAIL}}(\dot{\varepsilon}, \bar{\varepsilon}^p, \dots).$$

*DEFINE_TABLE

Purpose: To interpolate from point data a continuously indexed family of nonintersecting curves. The family of curves, $\mathcal{F}$, consists of x-y curves, $f_s(x)$, indexed by a parameter, s.

$$\mathcal{F} = \{f_s(x) | \forall s \in [s_{\min}, s_{\max}]\}.$$

The interpolation is built up by sampling functions in $\mathcal{F}$ at discrete parameter values, s_i,

$$f_{s_i}(x) \in \mathcal{F}.$$

The points, s_i, are input to LS-DYNA on the data cards for the *DEFINE_TABLE keyword. LS-DYNA requires that they be ordered from least to greatest. The curves, $f_{s_i}(x)$, must be defined as lists of (x, y) pairs in a collection of *DEFINE_CURVE sections that directly follow the *DEFINE_TABLE section. Each *DEFINE_CURVE section is paired to its corresponding s_i value by list position (and not load curve ID, for that see *DEFINE_TABLE_2D).

Card 1	1	2	3	4	5	6	7	8
Variable	TBID	SFA	OFFA					
Type	I	F	F					
Default	none	1.	0.					

Points Cards. Place one point per card. The values must be in ascending order. Input is terminated when a "*DEFINE_CURVE" keyword card is found.

Card 2	1	2	3	4	5	6	7	8
Variable	VALUE							
Type	F							
Default	0.0							

Include one ***DEFINE_CURVE** input section here for each point defined above. The i^{th} *DEFINE_CURVE card contains the curve at the i^{th} *DEFINE_TABLE value.

VARIABLE	DESCRIPTION
TBID	Table ID. Tables and Load curves may not share common ID's. LS-DYNA allows load curve ID's and table ID's to be used interchangeably.
SFA	Scale factor for VALUE.
OFFA	Offset for VALUE, see explanation below.
VALUE	Load curve will be defined corresponding to this value, e.g., this value could be a strain rate, see purpose above.

Motivation:

This capability was implemented with strain-rate dependent stress-strain relations in mind. To define such a function, the first step is to tabulate stress-strain curves at known strain-rate values. Then, the list of strain-rates is written in ascending order to the data cards following *DEFINE_TABLE. Following *DEFINE_TABLE, the tabulated stress-strain curves must be input to LS-DYNA as a set of *DEFINE_CURVE sections ordered so that the i^{th} curve corresponds to the i^{th} strain-rate point. This section is structured as:

***DEFINE_TABLE**
strain-rate point 1
strain-rate point 2
⋮
strain-rate point n
***DEFINE_CURVE**
[stress-strain curve at strain-rate 1]
***DEFINE_CURVE**
[stress-strain curve at strain-rate 2]
⋮
***DEFINE_CURVE**
[stress-strain curve at strain-rate n]

Details, Features and Limitations:

1. All the curves in a table must start from the same abscissa value and end at the same abscissa value. This limitation is necessary to avoid slow indirect addressing in the inner loops used in the constitutive model stress evaluation. Curves must not intersect except at the origin and end points.

2. Each curve may have unique spacing and an arbitrary number of points in its definition.

3. All the curves in a table must share the same value of LCINT.

4. In most applications, curves can only be extrapolated in one direction, that is, to the right of the last data point. An example would be curves representing effective stress vs. effective plastic strain. For cases when extrapolation is only to the right, the curves comprising a table are allowed to intersect *only* at their starting point but the curves *and their extrapolations* must not intersect elsewhere.

   For other applications in which the curves are extrapolated in both directions, the curves and their extrapolations are not allowed to intersect except at the origin (0,0). An example would be curves representing force vs. change in gage length where negative values are compressive and positive values are tensile.

5. Load curve IDs defined for the table may be referenced elsewhere in the input.

6. No keyword commands may come between *DEFINE_TABLE and the *DEFINE_CURVE commands that feed the table. The set of *DEFINE_CURVE commands must not be interrupted by any other keyword. This coupling between *DEFINE_TABLE and subsequent *DEFINE_CURVE commands is an exception to the general order-independence of the keyword format.

7. VALUE is scaled in the same manner as in *DEFINE_CURVE, i.e.,

$$\text{Scaled value} = \text{SFA}\times(\text{Defined value} + \text{OFFA}).$$

8. Unless stated otherwise in the description of a keyword command that references a table, there is no extrapolation beyond the range of VALUEs defined for the table. For example, if the table VALUE represents strain rate and the calculated strain rate exceeds the last/highest VALUE given by the table, the stress-strain curve corresponding to the last/highest table VALUE will be used.

*DEFINE_TABLE_2D

Purpose: Define a table. Unlike the *DEFINE_TABLE keyword above, a curve ID is specified for each value defined in the table. This allows the same curve ID to be referenced by multiple tables, and the curves may be defined anywhere in the input file. Other than these differences from *DEFINE_TABLE, the general rules given in the remarks of *DEFINE_TABLE still apply.

Card 1	1	2	3	4	5	6	7	8
Variable	TBID	SFA	OFFA					
Type	I	F	F					
Default	none	1.	0.					

Points Cards. Place one point per card. The values must be in ascending order. Input is terminated when a "*DEFINE_CURVE" keyword card is found.

Card 2	1	2	3	4
Variable	VALUE	CURVE ID		
Type	F	I		
Default	0.0	none		

VARIABLE **DESCRIPTION**

TBID

Table ID. Tables and Load curves may not share common ID's. LS-DYNA allows load curve ID's and table ID's to be used interchangeably.

SFA

Scale factor for VALUE.

OFFA

Offset for VALUE, see explanation below.

VALUE

Load curve will be defined corresponding to this value. The value could be, for example, a strain rate.

CURVEID

Load curve ID. See Remark 1.

Remarks:

1. Though generally of no concern to the user, curve CURVEID is automatically duplicated during initialization and the duplicate curve is automatically assigned a unique curve ID. The generated curve IDs used by the table are revealed in d3hsp. It is generally only necessary to know the generated curve IDs when interpreting warning messages about those curves.

*DEFINE_TABLE_3D

Purpose: Define a three dimensional table. For each value defined below, a table ID is specified. For example, in a thermally dependent material model, the value given below could correspond to temperature for a table ID defining effective stress versus strain curves for a set of strain rate values. Each table ID can be referenced by multiple three dimensional tables, and the tables may be defined anywhere in the input.

Card 1	1	2	3	4	5	6	7	8
Variable	TBID	SFA	OFFA					
Type	I	F	F					
Default	none	1.	0.					

Points Cards. Place one point per card. The values must be in ascending order. Input is terminated when a "*DEFINE_CURVE" keyword card is found.

Card 2	1	2	3	4
Variable	VALUE	TABLE ID		
Type	F	I		
Default	0.0	none		

VARIABLE	DESCRIPTION
TBID	Table ID. Tables and Load curves may not share common ID's. LS-DYNA allows load curve ID's and table ID's to be used interchangeably.
SFA	Scale factor for VALUE.
OFFA	Offset for VALUE, see explanation below.
VALUE	Load curve will be defined corresponding to this value, e.g., this value could be a strain rate for example.
TABLEID	Table ID.

Remarks:

1. VALUE is scaled in the same manner as in *DEFINE_CURVE, i.e.,

$$\text{Scaled value} = \text{SFA} \times (\text{Defined value} + \text{OFFA}).$$

2. Unless stated otherwise in the description of a keyword command that references a table, there is no extrapolation beyond the range of VALUEs defined for the table. For example, if the table VALUE represents strain rate and the calculated strain rate exceeds the last/highest VALUE given by the table, the stress-strain curve corresponding to the last/highest table VALUE will be used

*DEFINE_TABLE_MATRIX

This is an alternative input format for *DEFINE_TABLE that allows for reading data from an unformatted text file containing a matrix with data separated by comma delimiters. The purpose is to use data saved directly from excel sheets without having to convert it to keyword syntax.

Card 1	1	2	3	4	5	6	7	8
Variable	TBID	FILENAME						
Type	I	A70						

Card 2	1	2	3	4	5	6	7	8
Variable	NROW	NCOL	SROW	SCOL	SVAL	OROW	OCOL	OVAL
Type	I	I	F	F	F	F	F	F
Default	None	None	1.	1.	1.	0.	0.	0.

VARIABLE	DESCRIPTION
TBID	Table ID. Tables and Load curves may not share common ID's. LS-DYNA allows load curve ID's and table ID's to be used interchangeably.
FILENAME	Name of file containing table data (stored as a matrix).
NROW	Number of rows in the matrix, same as number of rows in the file FILENAME. A negative value of NROW switches the interpretation of rows and columns in the read matrix, see remarks.
NCOL	Number of columns in the matrix, same as number of data entries per row in the file FILENAME
SROW	Scale factor for row data, see remarks.
SCOL	Scale factor for column data, see remarks.
SVAL	Scale factor for matrix values, see remarks.

VARIABLE	DESCRIPTION
OROW	Offset for row data, see remarks.
OCOL	Offset for column data, see remarks.
OVAL	Offset for matrix values, see remarks.

Remarks:

The use of this keyword allows for inputting a table in form of a matrix from a file, exemplified here by a 4 × 5 matrix.

	C1	C2	C3	C4
R1	V11	V12	V13	V14
R2	V21	V22	V23	V24
R3	V31	V32	V33	V34
⋮	⋮	⋮	⋮	⋮

The unformatted file representing this matrix would contain the following data

```
,C1,C2,C3,C4
R1,V11,V12,V13,V14
R2,V21,V22,V23,V24
R3,V31,V32,V33,V34
```

Note that the first entry in the matrix is a dummy and delimited by an initial comma in the file. The keyword card for this matrix is: (TBID = 1000 and the filename is file.txt)

```
*DEFINE_TABLE_MATRIX
1000,file.txt
4,5
```

This is equivalent to using:

```
*DEFINE_TABLE
1000
C1
C2
C3
C4
*DEFINE_CURVE
1001
R1,V11
R2,V21
R3,V31
*DEFINE_CURVE
1002
R1,V12
```

```
R2,V22
R3,V32
*DEFINE_CURVE
1003
R1,V13
R2,V23
R3,V33
*DEFINE_CURVE
1004
R1,V14
R2,V24
R3,V34
```

All entries in the matrix can be scaled and offset following the convention for other tables and curves:

$$\text{Scaled Value} = \text{S[ROW/COL]} \times (\text{Value} + \text{O[ROW/COL]})$$

Finally, the matrix can be transposed by setting NROW to a negative value. In the example above this would mean that

```
*DEFINE_TABLE_MATRIX
1000,file.txt
-4,5
```

is equivalent to using:

```
*DEFINE_TABLE
1000
R1
R2
R3
*DEFINE_CURVE
1001
C1,V11
C2,V12
C3,V13
C4,V14
*DEFINE_CURVE
1002
C1,V21
C2,V22
C3,V23
C4,V24
*DEFINE_CURVE
1003
C1,V31
C2,V32
C3,V33
C4,V34
```

In this case, any scaling applies to the matrix entries before transposing the data, i.e., for row entries the scaled value is

$$\text{Scaled Value} = \text{SROW} \times (R + \text{OROW}),$$

and for column entries

$$\text{Scaled Value} = \text{SCOL} \times (C + \text{OCOL})$$

regardless the sign of TBID.

*DEFINE_TARGET_BOUNDARY

Purpose: This keyword is used to define the desired boundary of a formed part. This boundary provides the criteria used during blank-size development. The definitions associated with this keyword are used, exclusively, by the *INTERFACE_BLANKSIZE_DEVELOPMENT feature.

Point Cards. Include one card for each point in the curve. These points are interpolated to form a closed curve. This input is terminated with *END.

Card 1	1	2	3	4	5	6	7	8	9	10
Variable	X		Y		Z					
Type	E16.0		E16.0		E16.0					
Default	none		none		none					

VARIABLE	DESCRIPTION
X, Y, Z	Location coordinates of a target node.

Remarks:

1. The keyword file specified on the second data card for the *INTERFACE_BLANKSIZE_DEVELOPMENT keyword must contain a *DEFINE_TARGET_BOUNDARY keyword.

2. A partial keyword input is shown below. Note that the input is in a 3E16.0 FORTRAN format. Also note that the first and last curve points coincide.

```
*KEYWORD
*DEFINE_TARGET_BOUNDARY
    -1.83355e+02    -5.94068e+02    -1.58639e+02
    -1.80736e+02    -5.94071e+02    -1.58196e+02
    -1.78126e+02    -5.94098e+02    -1.57813e+02
    -1.75546e+02    -5.94096e+02    -1.57433e+02
    -1.72888e+02    -5.94117e+02    -1.57026e+02
          ⋮               ⋮               ⋮
    -1.83355e+02    -5.94068e+02    -1.58639e+02
*END
```

Typically, these boundary nodes obtained from the boundary curves for a final (trimmed) piece, or from a draw blank edge at a certain distance outside of the draw beads. LS-PrePost 4.1 can generate the points for this keyword from IGES data. To use IGES data select *Curve* → *Convert*→ *Method (To Keyword)* → Select *DEFINE_TARGET_BOUNDARY; pick the curves then select “To Key”. To output a keyword choose File → Save as → Save Keyword As, and select “Output Version” as “V971_R7”.

3. This feature is available in LS-DYNA R6 Revision 74560 and later releases.

*DEFINE_TRACER_PARTICLES_2D

Purpose: Define tracer particles that follow the deformation of a material. This is useful for visualizing the deformation of a part that is being adapted in a metal forming operation. Nodes used as tracer particles should only be used for visualization and not associated with anything in the model that may alter the response of the model, e.g., they should not be used in any elements except those with null materials.

Card 1	1	2	3	4	5	6	7	8
Variable	NSET	PSET						
Type	I	I						
Default	none	0						

VARIABLE	DESCRIPTION
NSET	The node set ID for the nodes used as tracer particles.
PSET	Optional part set ID. If this part set is specified, only tracer particles in these parts are updated and the others are stationary. If this part set is not specified, all tracer particles are updated.

*DEFINE_TRANSFORMATION

Purpose: Define a transformation for the INCLUDE_TRANSFORM keyword option. The *DEFINE_TRANSFORMATION command must be defined before the *INCLUDE_TRANSFORM command can be used.

Card 1	1	2	3	4	5	6	7	8
Variable	TRANID							
Type	I							
Default	none							

Transformation Cards. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	OPTION	A1	A2	A3	A4	A5	A6	A7
Type	A	F	F	F	F	F	F	F

VARIABLE	DESCRIPTION
TRANID	Transform ID.
OPTION	For the available options see the table below.
A1-A7	Parameters. See Table 15-47 below for the available options.

OPTION	PARAMETERS	FUNCTION
MIRROR	a1, a2, a3, a4, a5, a6, a7	Reflect, about a mirror plane defined to contain the point (a1, a2, a3) having its normal pointing from point (a1, a2, a3) toward (a4, a5, a6). Setting a7 = 1 reflects the coordinate system as well, i.e., the mirrored coordinate system uses the left-hand-rule to determine the local *z*-axis.
SCALE	a1, a2, a3	Scale the global x, y, and z coordinates of a point by a1, a2, and a3, respectively. If zero, a default of unity is set.
ROTATE	a1, a2, a3, a4, a5, a6, a7	Rotate through an angle (deg), a7, about a line with direction cosines a1, a2, and a3 passing through the point with coordinates a4, a5, and a6. If a4 through a7 are zero, then a1 and a2 are the ID's of two POINTs and a3 defines the rotation angle. The axis of rotation is defined by a vector going from point with ID a1 to point with ID a2.
TRANSL	a1, a2, a3	Translate the x, y, and z coordinates of a point by a1, a2, and a3, respectively.
POINT	a1,a2,a3,a4	Define a point with ID, a1, with the initial coordinates a2, a3, and a4.
POS6P	a1, a2, a3, a4, a5, a6	Positioning by 6 points. Affine transformation (rotation and translation, no scaling) given by three start points a1, a2, and a3 and three target points a4, a5, and a6. The six POINTs must be defined before they are referenced. Only 1 POS6P option is permitted within a *DEFINE_TRANSFORMATION definition.
POS6N	a1, a2, a3, a4, a5, a6	Positioning by 6 nodes. Affine transformation (rotation and translation, no scaling) given by three start nodes a1, a2, and a3 and three target nodes a4, a5, and a6. The six nodes must be defined before they are referenced. Only 1 POS6N option is permitted within a

OPTION	PARAMETERS	FUNCTION
		*DEFINE_TRANSFORMATION definition.

Table 15-47. List of allowed transformations.

Each option represents a transformation matrix. When more than one option is used, the transformation matrix defined by MIRROR, SCALE, ROTATE or TRANSL is applied to the previously defined existing matrix to form the new global transformation matrix. Therefore the ordering of the SCALE, ROTATE, and TRANSL commands is important. It is generally recommended to first scale, then rotate, and finally translate the model. POS6P and POS6N differ from other options by not applying their transformation matrix to the existing global matrix. Instead the transformation matrix defined by POS6P or POS6N replaces the existing matrix and becomes the new global transformation matrix.

The POINT option in ROTATE provides a means of defining rotations about axes defined by the previous transformations. The coordinates of the two POINTs are transformed by all the transformations up to the transformation where they are referenced. The POINTs must be defined before they are referenced, and their identification numbers are local to each *DEFINE_TRANSFORMATION. The coordinates of a POINT are transformed using all the transformations before it is referenced, not just the transformations between its definition and its reference. To put it another way, while the ordering of the transformations is important, the ordering between the POINTs and the transformations is not important.

NOTE: When *DEFINE_TRANSFORMATION is called from within the target of an *INCLUDE_TRANSFORM keyword, the result will involve stacked transformations.

In the following example, the *DEFINE_TRANSFORMATION command is used 3 times to input the same dummy model and position it as follows:

4. Transformation id 1000 imports the dummy model (dummy.k) and rotates it 45 degrees about *z*-axis at the point (0.0,0.0,0.0). Transformation id 1001 performs the same transformation using the POINT option.

5. Transformation id 2000 imports the same dummy model (dummy.k) and translates 1000 units in the *x* direction.

6. Transformation id 3000 imports the same dummy model (dummy.k) and translates 2000 units in the x direction. For each *DEFINE_TRANSFORMATION, the commands TRANSL, SCALE, and ROTATE are available. The trans-

formations are applied in the order in which they are defined in the file, e.g., transformation id 1000 in this example would translate, scale and then rotate the model. *INCLUDE_TRANSFORM uses a transformation id defined by a *DEFINE_TRANSFORMATION command to import a model and perform the associated transformations. It also allows the user upon importing the model to apply offsets to the various entity ids and perform unit conversion of the imported model.

```
*KEYWORD
*DEFINE_TRANSFORMATION
      1000
$ option &       dx&       dy&       dz&
TRANSL        0000.0       0.0       0.0
$ option &       dx&       dy&       dz&
SCALE           1.00       1.0       1.0
$ option &       dx&       dy&       dz&       px&       py&       pz&
angle&
ROTATE          0.00       0.0       1.0      0.00      0.00       0.0
45.00
*DEFINE_TRANSFORMATION
      1001
POINT              1       0.0       0.0       0.0
POINT              2       0.0       0.0       1.0
ROTATE             1         2      45.0
*DEFINE_TRANSFORMATION
      2000
$ option &       dx&       dy&       dz&
TRANSL        1000.0       0.0       0.0
*DEFINE_TRANSFORMATION
$ tranid &
     3000
$ option &       dx&       dy&       dz&

TRANSL        2000.0       0.0       0.0
*INCLUDE_TRANSFORM
dummy.k
$idnoff  &   ideoff&   idpoff& idmoff  &  idsoff &  iddoff&  iddoff &
         0         0         0         0         0        0         0
$  idroff&   ilctmf&
         0         0
$  fctmas&   fcttim&   fctlen&  fcttem &   incout&
    1.0000    1.0000      1.00       1.0         1
$ tranid &
      1000
*INCLUDE_TRANSFORM
dummy.k
$idnoff  &   ideoff&   idpoff& idmoff  &  idsoff &  iddoff&  iddoff &
   1000000   1000000   1000000   1000000   1000000  1000000   1000000
$  idroff&   ilctmf&
   1000000   1000000
$  fctmas&   fcttim&   fctlen&  fcttem &   incout&
    1.0000    1.0000      1.00       1.0         1
$ tranid &
      2000
*INCLUDE_TRANSFORM
dummy.k
$idnoff  &   ideoff&   idpoff& idmoff  &  idsoff &  iddoff&  iddoff &
   2000000   2000000   2000000   2000000   2000000  2000000   2000000
$  idroff&   ilctmf&
   2000000   2000000
$  fctmas&   fcttim&   fctlen&  fcttem &   incout&
    1.0000    1.0000      1.00       1.0         1
$ tranid &
```

```
      3000
*END
```

*DEFINE_TRIM_SEED_POINT_COORDINATES

Purpose: The keyword is developed to facilitate blank trimming in a stamping line die simulation. It allows for the trimming process and inputs to be defined independent of the previous process simulation results, and is applicable to shell, solid and laminate.

Card 1	1	2	3	4	5	6	7	8
Variable	NSEED	X1	Y1	Z1	X2	Y2	Z2	
Type	I	F	F	F	F	F	F	
Default	none	0	0	0	0	0.0	0.0	

VARIABLE	DESCRIPTION
NSEED	Number of seed points. Maximum value of “2” is allowed.
X1, Y1, Z1	Location coordinates of seed point #1.
X2, Y2, Z2	Location coordinates of seed point #2.

Remarks:

1. This keyword is used in conjunction with keywords *ELEMENT_TRIM and *DEFINE_CURVE_TRIM, where variables NSEED1 and NSEED2 should be left as blank. For detailed usage, refer to **Seed Node Definition** section in *DEFINE_CURVE_TRIM.

2. Variable NSEED is set to the number of seed points desired. For example, in a double attached drawn panel trimming, NSEED would equal to 2.

3. A partial keyword inputs for a single drawn panel trimming is listed below.

```
*INCLUDE_TRIM
drawn.dynain
*ELEMENT_TRIM
         1
*DEFINE_CURVE_TRIM_NEW
$#    TCID    TCTYPE      TFLG      TDIR     TCTOL      TOLN     NSEED
         1         2                  11     0.250
trimlines.iges
*DEFINE_TRIM_SEED_POINT_COORDINATES
$    NSEED        X1       Y1          Z1       X2        Y2        Z2
         1    -271.4    89.13   1125.679
*DEFINE_VECTOR
11,0.0,0.0,0.0,0.0,0.0,10.0
```

Typically, seed point coordinates can be selected from the stationary post in punch home position.

4. This feature is available in LS-DYNA R4 Revision 53048 and later releases.

*DEFINE_VECTOR

Purpose: Define a vector by defining the coordinates of two points.

Card	1	2	3	4	5	6	7	8
Variable	VID	XT	YT	ZT	XH	YH	ZH	CID
Type	I	F	F	F	F	F	F	I
Default	0	0.0	0.0	0.0	0.0	0.0	0.0	0
Remarks								

VARIABLE	DESCRIPTION
VID	Vector ID
XT	X-coordinate of tail of vector
YT	Y-coordinate of tail of vector
ZT	Z-coordinate of tail of vector
XH	X-coordinate of head of vector
YH	Y-coordinate of head of vector
ZH	Z-coordinate of head of vector
CID	Coordinate system ID to define vector in local coordinate system. All coordinates, XT, YT, ZT, XH, YH, and ZH are in respect to CID. EQ.0: global (default).

Remarks:

1. The coordinates should differ by a certain margin to avoid numerical inaccuracies.

*DEFINE_VECTOR_NODES

Purpose: Define a vector with two nodal points.

Card	1	2	3	4	5	6	7	8
Variable	VID	NODET	NODEH					
Type	I	I	I					
Default	0	0	0					

VARIABLE	DESCRIPTION
VID	Vector ID
NODET	Nodal point to define tail of vector
NODEH	Nodal point to define head of vector

EXAMPLES

The following examples demonstrate the input for these options:

*DEFINE_BOX

*DEFINE_COORDINATE_NODES,

*DEFINE_COORDINATE_SYSTEM,

*DEFINE_COORDINATE_VECTOR

*DEFINE_CURVE

*DEFINE_SD_ORIENTATION

*DEFINE_VECTOR commands.

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$   *DEFINE_BOX
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Define box number eight which encloses a volume defined by two corner
$  points: (-20.0, -39.0, 0.0) and (20.0, 39.0, 51.0).  As an example, this
$  box can be used as an input for the *INITIAL_VELOCITY keyword in which
$  all nodes within this box are given a specific initial velocity.
$
*DEFINE_BOX
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$    boxid       xmm       xmx       ymn       ymx       zmn       zmx
         8     -20.0      20.0     -39.0      39.0       0.0      51.0
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$   *DEFINE_COORDINATE_NODES
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Define local coordinate system number 5 using three nodes: 10, 11 and 20.
$  Nodes 10 and 11 define the local x-direction.  Nodes 10 and 20 define
$  the local x-y plane.
$
$  For example, this coordinate system (or any coordinate system defined using
$  a *DEFINE_COORDINATE_option keyword) can be used to define the local
$  coordinate system of a joint, which is required in order to define joint
$  stiffness using the *CONSTRAINED_JOINT_STIFFNESS_GENERALIZED keyword.
$
*DEFINE_COORDINATE_NODES
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      cid        n1        n2        n3
         5        10        11        20
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$   *DEFINE_COORDINATE_SYSTEM
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Define local coordinate system number 3 using three points.  The origin of
$  local coordinate system is at (35.0, 0.0, 0.0).  The x-direction is defined
$  from the local origin to (35.0, 5.0, 0.0).  The x-y plane is defined using
$  the vector from the local origin to (20.0, 0.0, 20.0) along with the local
$  x-direction definition.
$
*DEFINE_COORDINATE_SYSTEM
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      cid        Xo        Yo        Zo        Xl        Yl        Zl
         3      35.0       0.0       0.0      35.0       5.0       0.0
$
$       Xp        Yp        Zp
      20.0       0.0      20.0
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$  *DEFINE_COORDINATE_VECTOR
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Define local coordinate system number 4 using two vectors.
$    Vector 1 is defined from (0.0, 0.0, 0.0) to (1.0, 1.0, 0.0)
$    Vector 2 is defined from (0.0, 0.0, 0.0) to (1.0, 1.0, 1.0)
$  See the corresponding keyword command for a description.
$
*DEFINE_COORDINATE_VECTOR
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      cid        Xx        Yx        Zx        Xv        Yv        Zv
         4       1.0       1.0       0.0       1.0       1.0       1.0
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$  *DEFINE_CURVE
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Define curve number 517.  This particular curve is used to define the
$  force-deflection properties of a spring defined by a *MAT_SPRING_INELASTIC
$  keyword.  The abscissa value is offset 25.0 as a means of modeling a gap
$  at the front of the spring.  This type of spring would be a compression
$  only spring.
$
*DEFINE_CURVE
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$     lcid      sidr      scla      sclo      offa      offo
       517                                    25.0
$
$           abscissa            ordinate
                 0.0                 0.0
                80.0                58.0
                95.0                35.0
               150.0                44.5
               350.0                45.5
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$  *DEFINE_SD_ORIENTATION
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  A discrete spring is defined with two nodes in 3-D space.  However, it is
$  desired to have the force of that spring to act only in the z-direction.
$  The following definition makes this happen.  Additionally, vid = 7
$  must be specified in the *ELEMENT_DISCRETE keyword for this spring.
$
*DEFINE_SD_ORIENTATION
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      vid       iop        xt        yt        zt      nid1      nid2
         7         0       0.0       0.0       1.0
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$   *DEFINE_VECTOR
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Define vector number 5 from (0,0,0) to (0,1,1).  As an example, this vector
$  can be used to define the direction of the prescribed velocity of a node
$  using the *BOUNDARY_PRESCRIBED_MOTION_NODE keyword.
$
*DEFINE_VECTOR
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      vid        xt        yt        zt        xh        yh        zh
         3       0.0       0.0       0.0       0.0       1.0       1.0
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

*DEFORMABLE_TO_RIGID

The cards in this section are defined in alphabetical order and are as follows:

*DEFORMABLE_TO_RIGID

*DEFORMABLE_TO_RIGID_AUTOMATIC

*DEFORMABLE_TO_RIGID_INERTIA

If one of these cards is defined, then any deformable part defined in the model may be switched to rigid during the calculation. Parts that are defined as rigid (*MAT_RIGID) in the input are permanently rigid and cannot be changed to deformable.

Deformable parts may be switched to rigid at the start of the calculation by specifying them on the *DEFORMABLE_TO_RIGID card.

Part switching may be specified on a restart (see RESTART section of this manual) or it may be performed automatically by use of the *DEFORMABLE_TO_RIGID_AUTOMATIC cards.

The *DEFORMABLE_TO_RIGID_INERTIA cards allow inertial properties to be defined for deformable parts that are to be swapped to rigid at a later stage.

It is not possible to perform part material switching on a restart if it was not flagged in the initial analysis. The reason for this is that extra memory needs to be set up internally to allow the switching to take place. If part switching is to take place on a restart, but no parts are to be switched at the start of the calculation, no inertia properties for switching and no automatic switching sets are to be defined, then just define one *DEFORMABLE_TO_RIGID card without further input.

*DEFORMABLE_TO_RIGID

Purpose: Define materials to be switched to rigid at the start of the calculation.

Card	1	2	3	4	5	6	7	8
Variable	PID	MRB	PTYPE					
Type	I	I	A					
Default	none	0						

VARIABLE	DESCRIPTION
PID	Part ID of the part which is switched to a rigid material, also see *PART.
MRB	Part ID of the master rigid body to which the part is merged. If zero, the part becomes either an independent or master rigid body.
PTYPE	Type of PID: EQ.“PART”: PID is a part ID. EQ.“PSET”: PID is a part set ID. All parts included in part set PID will be switched to rigid at the start of the calculation.

*DEFORMABLE_TO_RIGID_AUTOMATIC

Purpose: Define a set of parts to be switched to rigid or to deformable at some stage in the calculation This keyword's data cards are enumerated below:

- 2 parameter cards: see "Card 1" and "Card 2" below
- D2R (see 5th field of Card 2) instances of "Card 3"
- R2D (see 6th field of Card 2) instances of "Card 4"
- Total number of cards = 2 + D2R + R2D

Card 1	1	2	3	4	5	6	7	8
Variable	SWSET	CODE	TIME 1	TIME 2	TIME 3	ENTNO	RELSW	PAIRED
Type	I	I	F	F	F	I	I	I
Default	none	0	0.	10^{20}	0.	0.	0	0
Remark		1				1, 2		3

Card 2	1	2	3	4	5	6	7	8
Variable	NRBF	NCSF	RWF	DTMAX	D2R	R2D	OFFSET	
Type	I	I	I	F	I	I	F	
Default	0	0	0	0.	0	0	0	

VARIABLE	DESCRIPTION

SWSET
Set number for this automatic switch set. Must be unique.

CODE
Activation switch code. Defines the test to activate the automatic material switch of the part:

EQ.0: switch takes place at time 1,

EQ.1: switch takes place between time 1 and time 2 if rigid wall force (specified below) is zero,

EQ.2: switch takes place between time 1 and time 2 if contact

VARIABLE	DESCRIPTION
	surface force (specified below) is zero, EQ.3: switch takes place between time 1 and time 2 if rigid wall force (specified below) is non-zero, EQ.4: switch takes place between time 1 and time 2 if contact surface force (specified below) is non-zero. EQ.5: switch is controlled by *SENSOR_CONTROL with TYPE = DEF2RIG, see *SENSOR_CONTROL. When CODE = 5, inputs of column 3 to column 8, TIME1~PAIRED, are ignored.
TIME1	Switch will not take place before this time.
TIME2	Switch will not take place after this time: EQ.0: Time 2 set to 10^{20}
TIME3	Delay period. After this part switch has taken place, another automatic switch will not take place for the duration of the delay period. If set to zero a part switch may take place immediately after this switch.
ENTNO	Rigid wall/contact surface number for switch codes 1, 2, 3, 4.
RELSW	Related switch set. The related switch set is another automatic switch set that must be activated before this part switch can take place: EQ.0: no related switch set.
PAIRED	Define a pair of related switches. EQ.0: not paired EQ.1: paired with switch set RELSW and is the Master switch. EQ.-1: paired with switch set RELSW and is the Slave switch.

VARIABLE **DESCRIPTION**

NRBF

Nodal rigid body flag.

For all values of NRBF, nodal rigid bodies defined using *CONSTRAINED_NODAL_RIGID_BODY and *CONSTRAINED_GENERALIZED_WELD_OPTION, which share any nodes with a rigid body created by deformable-to-rigid switching, are merged with the latter to form a single rigid body. Other actions dependent upon the value of NRBF are:

EQ.0: no further action,

EQ.1: delete all remaining nodal rigid bodies, that is, delete those nodal rigid bodies that do not share any nodes with a rigid body created by deformable-to-rigid switching,

EQ.2: Activate nodal rigid bodies.

NCSF

Nodal constraint set flag.

If nodal constraint/spot weld definitions are active in the deformable bodies that are switched to rigid, then the definitions should be deleted to avoid instabilities:

EQ.0: no change,

EQ.1: delete,

EQ.2: activate.

RWF

Flag to delete or activate rigid walls:

EQ.0: no change,

EQ.1: delete,

EQ.2: activate.

DTMAX

Maximum permitted time step size after switch.

D2R

Number of deformable parts to be switched to rigid plus number of rigid parts for which new master/slave rigid body combinations will be defined:

EQ.0: no parts defined.

R2D

Number of rigid parts to be switched to deformable:

EQ.0: no parts defined.

VARIABLE	DESCRIPTION
OFFSET	Optional contact thickness for switch to deformable. For contact, its value should be set to a value greater than the contact thickness offsets to ensure the switching occurs prior to impact. This option applies if and only if CODE is set to 3 or 4. For CODE = 3 all rigid wall options are implemented. For CODE = 4, the implementation works for the contact type CONTACT_AUTOMATIC_when the options: ONE_WAY_SURFACE_TO_SURFACE, NODES_TO_SURFACE, and SUR-FACE_TO_SURFACE are invoked.

Deformable to Rigid Cards. D2R additional cards with one for each part.

Card 3	1	2	3	4	5	6	7	8
Variable	PID	MRB	PTYPE					
Type	I	I	A					
Default	none	0						

VARIABLE	DESCRIPTION
PID	Part ID of the part which is switched to a rigid material. When PID is merged to another rigid body by the MRB field, this part is allowed to be rigid before the switch.
MRB	Part ID of the master rigid body to which part PID is merged. If zero, part PID becomes either an independent or master rigid body.
PTYPE	Type of PID: EQ.“PART”: PID is a part ID. EQ.“PSET”: PID is a part set ID.

Rigid to Deformable Cards. R2D additional cards with one for each part.

Card 4	1	2	3	4	5	6	7	8
Variable	PID	PTYPE						
Type	I	A						
Default	none							

VARIABLE	DESCRIPTION
PID	Part ID of the part which is switched to a deformable material.
PTYPE	Type of PID: EQ.“PART”: PID is a part ID. EQ.“PSET”: PID is a part set ID.

Remarks:

1. **Allowed Contact Types.** Only surface to surface and node to surface contacts can be used to activate an automatic part switch.

2. **Rigid Wall Numbering.** Rigid wall numbers are the order in which they are defined in the deck. The first rigid wall and the first contact surface encountered in the input deck will have an entity number of 1. The contact surface id is that as defined on the *CONTACT_…_ID card.

3. **Paired Switches.** Switch sets may be paired together to allow a pair of switches to be activated more than once. Each pair of switches should use consistent values for CODE, i.e. 1 & 3 or 2 & 4. Within each pair of switches, the related switch, RELSW, should be set to the ID of the other switch in the pair. The Master switch (PAIRED = 1) will be activated before the Slave switch (PAIRED = -1). Pairing allows the multiple switches to take place as for example when contact is made and lost several times during an analysis.

   If the delete switch is activated, ALL corresponding constraints are deactivated regardless of their relationship to a switched part. By default, constraints which are directly associated with a switched part are deactivated/activated as necessary.

```
$  Define a pair or related switches that will be activated by (no)force on
$  Contact 3.  To start with switch set 20 will be activated (PAIRED=1)
swapping
```

```
$  the PARTS to RIGID.  When the contact force is none zero switch set 10 will
be
$  activated swapping the PARTS to DEFORMABLE.  If the contact force returns
to
$  zero switch set 20 will be activated again making the PARTS RIGID.
$
*DEFORMABLE_TO_RIGID_AUTOMATIC
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>...
.8
$    swset      code    time 1    time 2    time 3     entno     relsw
paired
        20         2                                     3        10
1
$     nrbf      ncsf       rwf     dtmax       D2R       R2D
                                                 1
*DEFORMABLE_TO_RIGID_AUTOMATIC
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>...
.8
$    swset      code    time 1    time 2    time 3     entno     relsw
paired
        10         4                                     3        20
-1
$     nrbf      ncsf       rwf     dtmax       D2R       R2D
                                                    1
```

*DEFORMABLE_TO_RIGID_INERTIA

Purpose: Inertial properties can be defined for the new rigid bodies that are created when the deformable parts are switched. These can only be defined in the initial input if they are needed in a later restart. Unless these properties are defined, LS-DYNA will recompute the new rigid body properties from the finite element mesh. The latter requires an accurate mesh description. **When rigid bodies are merged to a master rigid body, the inertial properties defined for the master rigid body apply to all members of the merged set.**

Card 1	1	2	3	4	5	6	7	8
Variable	PID							
Type	I							
Default	none							

Card 2	1	2	3	4	5	6	7	8
Variable	XC	YC	ZC	TM				
Type	F	F	F	F				

Card 3	1	2	3	4	5	6	7	8
Variable	IXX	IXY	IXZ	IYY	IYZ	IZZ		
Type	F	F	F	F	F	F		
Default	none	0.0	0.0	none	0.0	none		

VARIABLE	DESCRIPTION
PID	Part ID, see *PART.
XC	x-coordinate of center of mass
YC	y-coordinate of center of mass

VARIABLE	DESCRIPTION
ZC	z-coordinate of center of mass
TM	Translational mass
IXX	I_{xx} (the xx component of inertia tensor)
IXY	I_{xy}
IXZ	I_{xz}
IYY	I_{yy}
IYZ	I_{yz}
IZZ	I_{zz}

*ELEMENT

The element cards in this section are defined in alphabetical order:

*ELEMENT_BLANKING

*ELEMENT_BEAM_*{OPTION}*_*{OPTION}*

*ELEMENT_BEAM_PULLEY

*ELEMENT_BEAM_SOURCE

*ELEMENT_DIRECT_MATRIX_INPUT

ELEMENT_DISCRETE_*{OPTION}

ELEMENT_DISCRETE_SPHERE_*{OPTION}

*ELEMENT_GENERALIZED_SHELL

*ELEMENT_GENERALIZED_SOLID

ELEMENT_INERTIA_*{OPTION}

*ELEMENT_INTERPOLATION_SHELL

*ELEMENT_INTERPOLATION_SOLID

*ELEMENT_LANCING

ELEMENT_MASS_*{OPTION}

ELEMENT_MASS_MATRIX_*{OPTION}

ELEMENT_MASS_PART_*{OPTION}

*ELEMENT_PLOTEL

*ELEMENT_SEATBELT

*ELEMENT_SEATBELT_ACCELEROMETER

*ELEMENT_SEATBELT_PRETENSIONER

*ELEMENT_SEATBELT_RETRACTOR

*ELEMENT_SEATBELT_SENSOR

*ELEMENT_SEATBELT_SLIPRING

ELEMENT_SHELL_*{OPTION}

*ELEMENT_SHELL_NURBS_PATCH

*ELEMENT_SHELL_SOURCE_SINK

ELEMENT_SOLID_*{OPTION}

*ELEMENT_SOLID_NURBS_PATCH

*ELEMENT_SPH

*ELEMENT_TRIM

ELEMENT_TSHELL_*{OPTION}

The ordering of the element cards in the input file is completely arbitrary. An arbitrary number of element blocks can be defined preceded by a keyword control card.

*ELEMENT_BLANKING

Purpose: This keyword is used to define a part set to be used in keyword *DEFINE_-FORMING_BLANKMESH for a blank mesh generation.

Card 1	1	2	3	4	5	6	7	8
Variable	PSID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
PSID	Part set ID, defined by *SET_PART.

Remarks:

1. This keyword is used in conjunction with *DEFINE_FORMING_BLANKMESH to generate mesh on a sheet blank for metal forming simulation.

2. This feature is available in LS-DYNA R5 Revision 59165 or later releases.

*ELEMENT_BEAM_*{OPTION}*_*{OPTION}*

Available options include:

<BLANK>

THICKNESS, SCALAR, SCALR or SECTION

PID

OFFSET

ORIENTATION

WARPAGE

ELBOW *(beta)*

Purpose: Define two node elements including 3D beams, trusses, 2D axisymmetric shells, and 2D plane strain beam elements. The type of the element and its formulation is specified through the part ID (see *PART) and the section ID (see *SECTION_BEAM).

Two alternative methods are available for defining the cross sectional property data. The THICKNESS and SECTION options are provided for the user to override the *SECTION_BEAM data which is taken as the default if the THICKNESS or SECTION option is not used. . The SECTION option applies only to resultant beams (ELFORM.eq.2 on *SECTION_BEAM). End release conditions are imposed using constraint equations, and caution must be used with this option as discussed in remark 2 below. The SCALAR/SCALR options applies only to material model type 146, *MAT_1DOF_GENERALIZED_SPRING.

The PID option is used by the type 9 spot weld element only and is ignored for all other beam types. When the PID option is active an additional card is read that gives two part ID's that are tied by the spot weld element. If the PID option is inactive for the type 9 element the nodal points of the spot weld are located to the two nearest master segments. In either case, *CONTACT_SPOTWELD must be defined with the spot weld beam part as slave and the shell parts (including parts PID1 and PID2) as master. The surface of each segment should project to the other and in the most typical case the node defining the weld, assuming only one node is used, should lie in the middle; however, this is not a requirement. Note that with the spot weld elements only one node is needed to define the weld, and two nodes are optional.

The options ORIENTATION and OFFSET are not available for discrete beam elements.

The ELBOW option is a 3-node beam element with quadratic interpolation that is tailored for the piping industry. It includes 12 degrees of freedom, including 6 ovalization degrees of freedom for describing the ovalization, per node. That is a total of 36 DOFs for each element. An internal pressure can also be given that tries to stiffen the pipe. The pressure, if activated accordingly, can also contribute to the elongation of the pipe. The control node must be given but it is only used for initially straight elbow elements. For curved elements the curvature center is used as the control node. See *SECTION_BEAM for more information about the physical properties such as pressure and output options.

Card 1	1	2	3	4	5	6	7	8	9	10
Variable	EID	PID	N1	N2	N3	RT1	RR1	RT2	RR2	LOCAL
Type	I	I	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	0	0	0	0	2
Remarks					1	2,3	2,3	2,3	2,3	2,3

Thickness Card. Additional Card for THICKNESS keyword option.

Card	1	2	3	4	5	6	7	8	9	10
Variable	PARM1		PARM2		PARM3		PARM4		PARM5	
Type	F		F		F		F		F	
Remarks	5		5		5		5		5,6	

Section Card. Additional card required for SECTION keyword option.

Card	1	2	3	4	5	6	7	8
Variable	STYPE	D1	D2	D3	D4	D5	D6	
Type	A	F	F	F	F	F	F	
Remarks								

Scalar card. Additional card for SCALAR keyword option.

Card	1	2	3	4	5	6	7	8	9	10
Variable	VOL		INER		CID		DOFN1		DOFN2	
Type	F		F		F		F		F	

Scalar Card (alternative). Additional card for SCALR keyword option.

Card	1	2	3	4	5	6	7	8	9	10
Variable	VOL		INER		CID1		CID2		DOFNS	
Type	F		F		F		F		F	

Spot Weld Part Card. Additional card for PID keyword option.

Card	1	2	3	4	5	6	7	8	9	10
Variable	PID1	PID2								
Type	I	I								
Default	none	none								
Remarks										

Offset Card. Additional card for OFFSET keyword option.

Card	1	2	3	4	5	6	7	8
Variable	WX1	WY1	WZ1	WX2	WY2	WZ2		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	0.0	0.0	0.0		
Remarks	8	8	8	8	8	8		

Orientation Card. Additional card for ORIENTATION keyword option.

Card	1	2	3	4	5	6	7	8
Variable	VX	VY	VZ					
Type	F	F	F					
Default	0.0	0.0	0.0					
Remarks								

Warpage Card. Additional card for WARPAGE keyword option.

Card	1	2	3	4	5	6	7	8
Variable	SN1	SN2						
Type	I	I						
Default	none	none						
Remarks								

Elbow Card. Additional card for ELBOW keyword option.

Card	1	2	3	4	5	6	7	8
Variable	MN							
Type	I							
Default	none							
Remarks								

VARIABLE	DESCRIPTION
EID	Element ID. A unique ID is generally required, i.e., EID must be different from the element ID's also defined under *ELEMENT_DISCRETE and *ELEMENT_SEATBELT. If the parameter, BEAM, is set to 1 on the keyword input for *DATABASE_BINARY_D3PLOT, the null beams used for visualization are not created for the latter two types, and the ID's used for the discrete elements and the seatbelt elements can be identical to those defined here.
PID	Part ID, see *PART.
N1	Nodal point (end) 1.
N2	Nodal point (end) 2. This node is optional for the spot weld, beam type 9, since if it not defined it will be created automatically and given a non-conflicting nodal point ID. Nodes N1 and N2 are automatically positioned for the spot weld beam element. For the zero length discrete beam elements where one end is attached to ground, set N2 = -N1. In this case, a fully constrained nodal point will be created with a unique ID for node N2.
N3	Nodal point 3 for orientation. The third node, N3, is optional for beam types 3, 6, 7, 8, and if the cross-section is circular, beam types 1 and 9. The third node is used for the discrete beam, type 6, if and only if SCOOR is set to 2.0 in the *SECTION_BEAM input, but even in this case it is optional. An orientation vector can be defined directly by using the option, ORIENTATION. In this case N3 can be defined as zero.
RT1, RT2	Release conditions for translations at nodes N1 and N2,

VARIABLE **DESCRIPTION**

respectively:

EQ.0: no translational degrees-of-freedom are released

EQ.1: x-translational degree-of-freedom

EQ.2: y-translational degree-of-freedom

EQ.3: z-translational degree-of-freedom

EQ.4: x and y-translational degrees-of-freedom

EQ.5: y and z-translational degrees-of-freedom

EQ.6: z and x-translational degrees-of-freedom

EQ.7: x, y, and z-translational degrees-of-freedom (3DOF)

This option does not apply to the spot weld, beam type 9.

RR1, RR2

Release conditions for rotations at nodes N1 and N2, respectively:

EQ.0: no rotational degrees-of-freedom are released

EQ.1: x-rotational degree-of-freedom

EQ.2: y-rotational degree-of-freedom

EQ.3: z-rotational degree-of-freedom

EQ.4: x and y-rotational degrees-of-freedom

EQ.5: y and z-rotational degrees-of-freedom

EQ.6: z and x-rotational degrees-of-freedom

EQ.7: x, y, and z-rotational degrees-of-freedom (3DOF)

This option does not apply to the spot weld, beam type 9.

LOCAL

Coordinate system option for release conditions:

EQ.1: global coordinate system

EQ.2: local coordinate system (default)

PARM1

Based on beam type:

Type.EQ.1: beam thickness, s direction at node 1

Type.EQ.2: area

Type.EQ.3: area

Type.EQ.4: beam thickness, s direction at node 1

Type.EQ.5: beam thickness, s direction at node 1

VARIABLE **DESCRIPTION**

Type.EQ.6: volume, see description for VOL below.

Type.EQ.7: beam thickness, s direction at node 1

Type.EQ.8: beam thickness, s direction at node 1

Type.EQ.9: beam thickness, s direction at node 1

PARM2 Based on beam type:

Type.EQ.1: beam thickness, s direction at node 2

Type.EQ.2: Iss

Type.EQ.3: ramp-up time for dynamic relaxation

Type.EQ.4: beam thickness, s direction at node 2

Type.EQ.5: beam thickness, s direction at node 2

Type.EQ.6: Inertia, see description for INER below.

Type.EQ.7: beam thickness, s direction at node 2

Type.EQ.8: beam thickness, s direction at node 2

Type.EQ.9: beam thickness, s direction at node 2

PARM3 Based on beam type:

Type.EQ.1: beam thickness, t direction at node 1

Type.EQ.2: Itt

Type.EQ.3: initial stress for dynamic relaxation

Type.EQ.4: beam thickness, t direction at node 1

Type.EQ.5: beam thickness, t direction at node 1

Type.EQ.6: local coordinate ID

Type.EQ.7: not used.

Type.EQ.8: not used.

Type.EQ.9: beam thickness, t direction at node 1

PARM4 Based on beam type:

Type.EQ.1: beam thickness, t direction at node 2

Type.EQ.2: Irr

Type.EQ.3: not used

Type.EQ.4: beam thickness, t direction at node 2

VARIABLE **DESCRIPTION**

Type.EQ.5: beam thickness, t direction at node 2

Type.EQ.6: area

Type.EQ.7: not used.

Type.EQ.8: not used.

Type.EQ.9: beam thickness, t direction at node 2

PARM5

Based on beam type:

Type.EQ.1: not used

Type.EQ.2: shear area

Type.EQ.3: not used

Type.EQ.4: not used

Type.EQ.5: not used

Type.EQ.6: offset

Type.EQ.7: not used.

Type.EQ.8: not used.

Type.EQ.9: print flag to SWFORC file. The default is taken from the SECTION_BEAM input. To override set PARM5 to 1.0 to suppress printing, and to 2.0 to print.

STYPE

Section type (A format) of resultant beam, see Figure 36-1:

EQ.SECTION_01: I-Shape
EQ.SECTION_02: Channel
EQ.SECTION_03: L-Shape
EQ.SECTION_04: T-Shape
EQ.SECTION_05: Tubular box
EQ.SECTION_06:Z-Sape
EQ.SECTION_07: Trapezoidal
EQ.SECTION_08: Circular
EQ.SECTION_09: Tubular
EQ.SECTION_10: I-Shape 2
EQ.SECTION_11: Solid box

EQ.SECTION_12: Cross
EQ.SECTION_13: H-Shape
EQ.SECTION_14: T-Shape 2
EQ.SECTION_15: I-Shape 3
EQ.SECTION_16: Channel 2
EQ.SECTION_17: Channel 3
EQ.SECTION_18: T-Shape 3
EQ.SECTION_19: Box-Shape 2
EQ.SECTION_20: Hexagon
EQ.SECTION_21: Hat-Shape
EQ.SECTION_22: Hat-Shape 2

D1-D6

Input parameters for section option using STYPE above.

VARIABLE **DESCRIPTION**

PID1

Optional part ID for spot weld element type 9.

PID2

Optional part ID for spot weld element type 9.

VOL

Volume of discrete beam and scalar (MAT_146) beam. If the mass density of the material model for the discrete beam is set to unity, the magnitude of the lumped mass can be defined here instead. This lumped mass is partitioned to the two nodes of the beam element. The translational time step size for the type 6 beam is dependent on the volume, mass density, and the translational stiffness values, so it is important to define this parameter. Defining the volume is also essential for mass scaling if the type 6 beam controls the time step size.

INER

Mass moment of inertia for the six degree of freedom discrete beam and scalar (MAT_146) beam. This lumped inertia is partitioned to the two nodes of the beam element. The rotational time step size for the type 6 beam is dependent on the lumped inertia and the rotational stiffness values, so it is important to define this parameter if the rotational springs are active. Defining the rotational inertia is also essential for mass scaling if the type 6 beam rotational stiffness controls the time step size.

CID

Coordinate system ID for orientation, material type 146, see *DEFINE_COORDINATE_SYSTEM. If CID = 0, a default coordinate system is defined in the global system.

DOFN1

Active degree-of-freedom at node 1, a number between 1 to 6 where 1, 2, and 3 are the x, y, and z-translations and 4, 5, and 6 are the x, y, and z-rotations. This degree-of-freedom acts in the local system given by CID above. This input applies to material model type 146.

DOFN2

Active degree-of-freedom at node 2, a number between 1 to 6. This degree-of-freedom acts in the local system given by CID above. This input applies to material model type 146.

CID1

Coordinate system ID at node 1 for orientation, material type 146, see *DEFINE_COORDINATE_SYSTEM. If CID1 = 0, a default coordinate system is defined in the global system.

CID2

Coordinate system ID at node 2 for orientation, material type 146, see *DEFINE_COORDINATE_SYSTEM. If CID2 = 0, a default coordinate system is defined in the global system.

VARIABLE	DESCRIPTION
DOFNS	Active degrees-of-freedom at node 1 and node 2. A two-digit number, the first for node 1 and the second for node 2, between 11 to 66 is expected where 1, 2, and 3 are the x, y, and z-translations and 4, 5, and 6 are the x, y, and z-rotations. These degrees-of-freedom acts in the local system given by CID1 and CID2 above. This input applies to material model type 146. If DOFNS = 12 the node one has an x-translation and node 2 has a y translation.
WX1-WZ1	Offset vector at nodal point N1. See Remark 8.
WX2-WZ2	Offset vector at nodal point N2. See Remark 8.
VX,VY,VZ	Coordinates of an orientation vector relative to node N1. In this case, the orientation vector points to a virtual third node and so the input variable N3 should be left undefined.
SN1	Scalar nodal point (end) 1. This node is required for the WARPAGE option.
SN2	Scalar nodal point (end) 2. This node is required for the WARPAGE option.

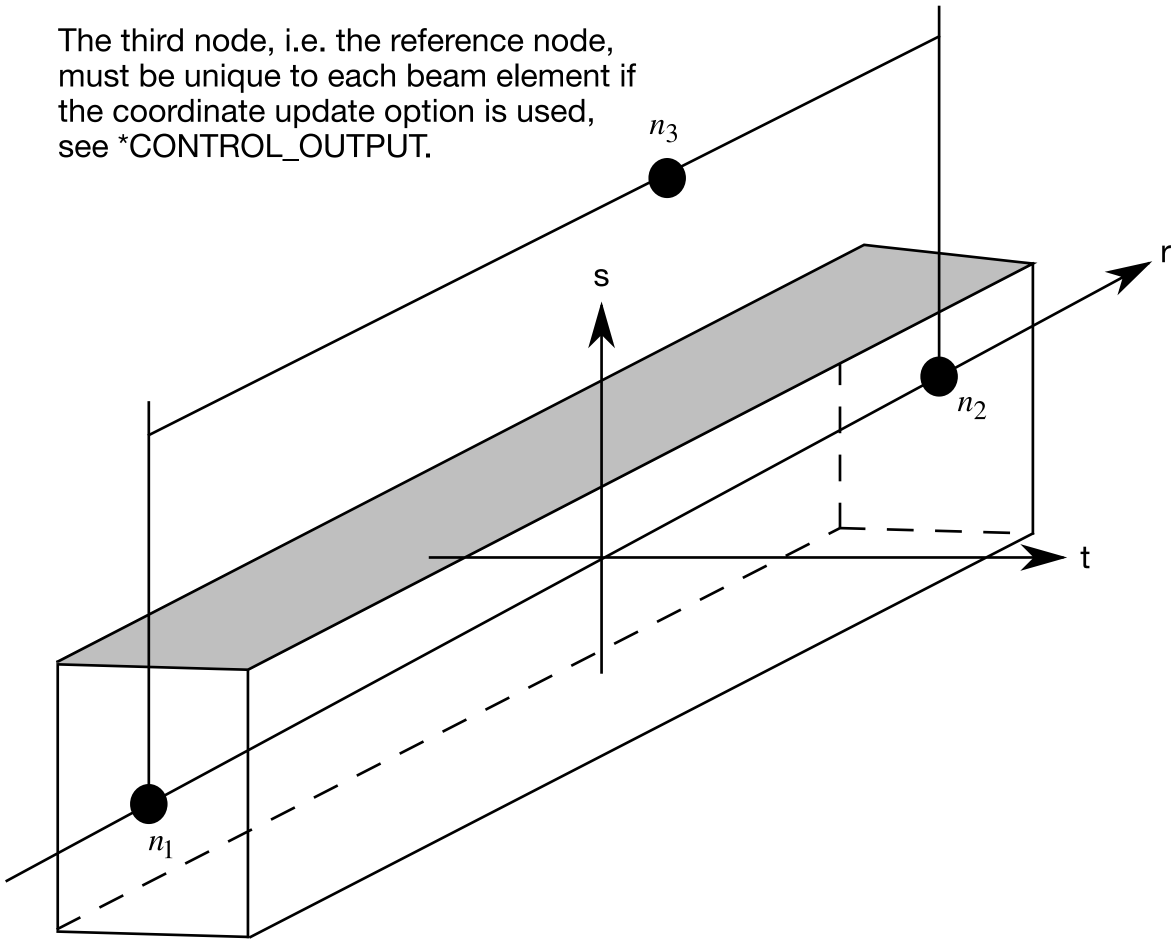

The third node, i.e. the reference node, must be unique to each beam element if the coordinate update option is used, see *CONTROL_OUTPUT.
n_3
s
r
n_2
t
n_1

Figure 17-1. LS-DYNA beam elements. Node n_3 determines the initial orientation of the cross section.

VARIABLE	DESCRIPTION
MN	Middle node for the ELBOW element. See Remark 9.

Remarks:

1. A plane through N_1, N_2, and N_3 defines the orientation of the principal r-s plane of the beam, see Figure 17-1.

2. This option applies to all three-dimensional beam elements. The released degrees-of-freedom can be either global, or local relative to the local beam coordinate system, see Figure 17-1. A local coordinate system is stored for each node of the beam element and the orientation of the local coordinate systems rotates with the node. To properly track the response, the nodal points with a released resultant are automatically replaced with new nodes to accommodate

the added degrees-of-freedom. Then constraint equations are used to join the nodal points together with the proper release conditions imposed.

Nodal points which belong to beam elements which have release conditions applied cannot be subjected to other constraints such as applied displacement/velocity/acceleration boundary conditions, nodal rigid bodies, nodal constraint sets, or any of the constraint type contact definitions.

Force type loading conditions and penalty based contact algorithms may be used with this option.

3. Please note that this option may lead to nonphysical constraints if the translational degrees-of-freedom are released, but this should not be a problem if the displacements are infinitesimal.

4. If the THICKNESS option is not used, or if THICKNESS is used but essential PARMx values are not provided, beam properties are taken from *SECTION_BEAM.

5. In the case of the THICKNESS option for type 6, i.e., discrete beam elements, PARM1 through PARM5 replace the first five parameters on card 2 of *SECTION_BEAM. Cables are a subset of type 6 beams. PARM1 is for non-cable discrete beams and is optional for cables, PARM2 and PARM3 apply only to non-cable discrete beams, and PARM4 and PARM5 apply only to cables.

6. In the THICKNESS option, PARM5 applies only to beam types 2, 6 (cables only), and 9.

7. The stress resultants are output in local coordinate system for the beam. Stress information is optional and is also output in the local system for the beam.

8. Beam offsets are sometimes necessary for correctly modeling beams that act compositely with other elements such as shells or other beams. When the OFFSET option is specified, global X, Y, and Z components of two offset vectors are given, one vector for each of the two beam nodes. The offset vector extends from the beam node (N1 or N2) to the reference axis of the beam. The beam reference axis lies at the origin of the local s and t axes. For beam formulations 1 and 11, this origin is halfway between the outermost surfaces of the beam cross-section. Note that for cross-sections that are not doubly symmetric, e.g, a T-section, the reference axis does not pass through the centroid of the cross-section. For beam formulation 2, the origin is at the centroid of the cross-section.

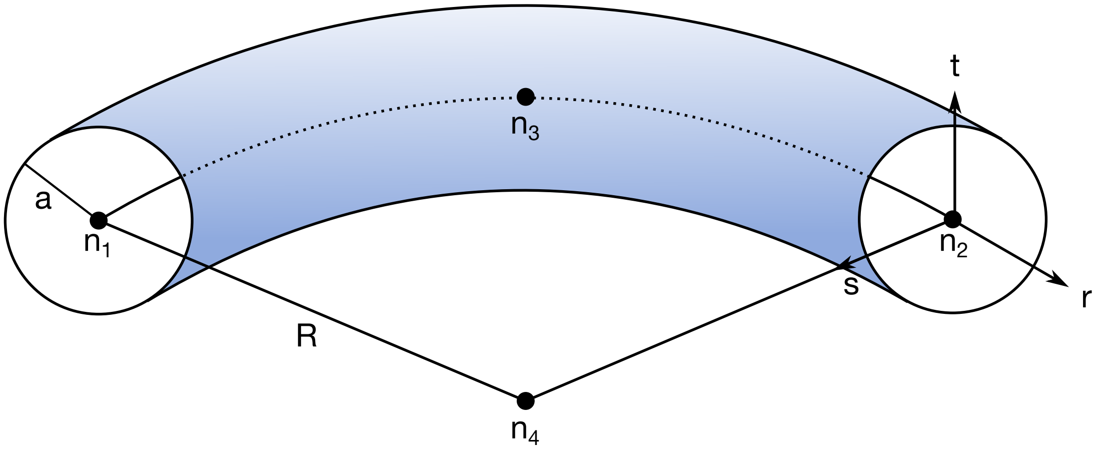


Figure 17-2. LS-DYNA Elbow element. Node n4 is the control node and is given as the beam center of curvature.

9. The Elbow beam is defined with 4 nodes, see Figure 17-2. Node n1, n2 being the end nodes, and node n3 is the middle node. It is custom to set n3 at the midpoint of the beam. Node n4 is an orientation node that should be at the curvature center of the beam. If a straight beam is defined initially, the orientation node must be defined and should be on the convex side of the beam. If a curved beam is defined initially the orientation node is automatically calculated as the center of the beam curvature. However, an orientation node is still required at the input.

   The extra nodes that include the ovalization degree of freedom are written to the messag file during initialization. These extra nodes have 3 dofs each. That means that there are 2 extra nodes for each physical node. For example it can look something like this:

```
ELBOW BEAM:           1
n1-n3-n2:             1     3     2
ovalization nodes:    5     7     6
                      8    10     9
```

   And it means that node 1 have the ovalization extra nodes 5 and 8. The first line of ovalization nodes includes the c_1, c_2 and c_3 parameters, and the second line includes the d_1, d_2 and d_3 parameters. That means that node 5 include c_1, c_2 and c_3, and node 8 include d_1, d_2 and d_3 for beam node 1. All ovalization dofs can be written to the ascii file "elbwov" if the correct print flag is set on *SECTION_BEAM. These extra nodes can be constrained as usual nodes. For example for a cantilever beam that is mounted at node 1, the nodes 1, 5 and 8 should be constrained. The ovalization is approximated with the following trigonometric function:

$$w(r,\theta) = \sum_{k=1}^{3} \sum_{m=1}^{3} h_k(r)\left(c_m^k \cos 2m\theta + d_m^k \sin 2m\theta\right), \qquad -1 \le r \le 1, 0 \le \theta < 2\pi$$

where h_k is the interpolation function at the physical node k.

The Elbow beam only supports tubular cross sections and the pipe outer radius, a, should be smaller than the pipe bend radius, R. That is a/R<<1. Moreover, the ELBOW beams have 4 stresses: axial rr-, shear rs-, shear rt- and loop-stresses. The loop stress is written at each integration point and can be visualized in LS-PrePost with the user fringe plot file "elbwlp.k". NOTE that the loop-stress is not written to d3plot as default! The NEIPB flag on *DATABASE_EXTENT_BINARY must be set to enable d3plot support.

Right now there is only basic support from the material library. The following materials are currently supported for the ELBOW beam (if requested more materials might be added in the future):

*MAT_ELASTIC (MAT_001)

*MAT_PLASTIC_KINEMATIC (MAT_003)

*MAT_ELASTIC_PLASTIC_THERMAL (MAT_004)

*MAT_VISCOELASTIC (MAT_006)

*MAT_PIECEWISE_LINEAR_PLASTICITY (MAT_024)

*MAT_DAMAGE_3 (MAT_153, explicit only)

*MAT_CONCRETE_BEAM (MAT_195)

*ELEMENT_BEAM_PULLEY

Purpose: Define pulley for beam elements. This feature is implemented for truss beam elements (*SECTION_BEAM, ELFORM = 3) using materials *MAT_001 and *MAT_156, or discrete beam elements (ELFORM = 6) using *MAT_CABLE_DISCRETE_BEAM.

Card 1	1	2	3	4	5	6	7	8
Variable	PUID	BID1	BID2	PNID	FD	FS	LMIN	DC
Type	I	I	I	I	F	F	F	F
Default	0	0	0	0	0.0	0.0	0.0	0.0

VARIABLE	DESCRIPTION
PUID	Pulley ID. A unique number has to be used.
BID1	Truss beam element 1 ID.
BID2	Truss beam element 2 ID.
PNID	Pulley node, NID.
FD	Coulomb dynamic friction coefficient.
FS	Optional Coulomb static friction coefficient.
LMIN	Minimum length, see notes below.
DC	Optional decay constant to allow smooth transition between the static and dynamic friction coefficient, i.e., $$\mu_c = \mathrm{FD} + (\mathrm{FS} - \mathrm{FD})e^{-\mathrm{DC}\times\lvert v_{\mathrm{rel}}\rvert}$$

Remarks:

Remarks:

Elements 1 and 2 should share a node which is coincident with the pulley node. The pulley node should not be on any beam elements.

Pulleys allow continuous sliding of a truss beam element through a sharp change of angle. Two elements (1 & 2 in Figure 17-21 of *ELEMENT_SEATBELT_SLIPRING) meet at the pulley. Node B in the beam material remains attached to the pulley node, but beam material (in the form of unstretched length) is passed from element 1 to element 2 to achieve slip. The amount of slip at each time step is calculated from the ratio of forces in elements 1 and 2. The ratio of forces is determined by the relative angle between elements 1 and 2 and the coefficient of friction, FD. The tension in the beams are taken as T_1 and T_2, where T_2 is on the high tension side and T_1 is the force on the low tension side. Thus, if T_2 is sufficiently close to T_1, no slip occurs; otherwise, slip is just sufficient to reduce the ratio T_2/T_1 to $e^{\mathrm{FC}\times\theta}$, where θ is the wrap angle, see Figures 17-22 of *ELEMENT_SEATBELT_SLIPRING. The out-of-balance force at node B is reacted on the pulley node; the motion of node B follows that of pulley node.

If, due to slip through the pulley, the unstretched length of an element becomes less than the minimum length LMIN, the beam is remeshed locally: the short element passes through the pulley and reappears on the other side (see Figure 17-21). The new unstretched length of e_1 is 1.1 × minimum length. The force and strain in e_2 and e_3 are unchanged; while the force and strain in e_1 are now equal to those in e_2. Subsequent slip will pass material from e_3 to e_1. This process can continue with several elements passing in turn through the pulley.

To define a pulley, the user identifies the two beam elements which meet at the pulley, the friction coefficient, and the pulley node. If BID1 and BID2 are defined as 0 (zero), adjacent beam elements are automatically detected. The two elements must have a common node coincident with the pulley node. No attempt should be made to restrain or constrain the common node for its motion will automatically be constrained to follow the pulley node. Typically, the pulley node is part of a structure and, therefore, beam elements should not be connected to this node directly, but any other feature can be attached, including rigid bodies.

*DATABASE_PLLYOUT can be used to write a time history output database pllyout for the pulley which records beam IDs, slip, slip rate, resultant force, and wrap angle.

*ELEMENT_BEAM_SOURCE

Purpose: Define a nodal source for beam elements. This feature is implemented *only* for truss beam elements (*SECTION_BEAM, ELFORM = 3) with material *MAT_001 or for discrete beam elements (ELFORM = 6) with material *MAT_071.

Card 1	1	2	3	4	5	6	7	8
Variable	BSID	BSNID	BSEID	NELE	LFED	FPULL	LMIN	
Type	I	I	I	I	F	F	F	
Default	0	0	0	0	0.0	0.0	0.0	

VARIABLE	DESCRIPTION
BSID	Beam Source ID. A unique number has to be used.
BSNID	Source node ID.
BSEID	Source element ID.
NELE	Number of elements to be pulled out.
LFED	Beam element fed length (typical element initial length).
FPULL	Pull-out force. GT.0: Constant value, LT.0: Load curve ID = (-FPULL) which defines pull-out force as a function of time. Either *DEFINE_FUNCTION or *DEFINE_ CURVE_FUNCTION (with argument TIME) can be used.
LMIN	Minimum beam element length, see notes below. One to two tenth of the fed length LFED is usually a good choice.

Remarks:

The source node BSNID can be defined for itself or it can be part of another structure. It is free to move in space during the simulation process. Initially, the source node should have the same coordinates as one of the source element (BSEID) nodes, but not having the same ID. If the pre-defined pull-out force FPULL is exceeded in the element next to the source, beam material gets drawn out by increasing the length of the beam without increasing its axial force (equivalent to ideal plastic flow at a given yield force). If more than the pre-defined length LFED is drawn out, a new beam element is generated. A new beam element has an initial undeformed length of 1.1 × LMIN. The maximum number of elements NELE times the fed length LFED defines the maximum cable length that can be pulled out from the source node.

ELEMENT_BEARING_*OPTION

The available option is:

TITLE

Purpose: Define a bearing between two nodes. A description of this model can be found in Carney, Howard, Miller, and Benson [2014].

Title Card. Additional card for title keyword Option.

Card 1	1	2	3	4	5	6	7	8
Variable	TITLE							
Type	C							
Default	none							
Remarks	1							

Card 1	1	2	3	4	5	6	7	8
Variable	ID	ITYPE	N1	CID1	N2	CID2	NB	
Type	I	I	I	I	I	I	I	
Default	0	0	0	0	0	0	0	

Material Properties Card.

Card 2	1	2	3	4	5	6	7	8
Variable	EBALL	PRBALL	ERACE	PRRACE	STRESL			
Type	F	F	F	F	F			
Default	0.0	0.0	0.0	0.0	0.0			

Geometry Card.

Card 3	1	2	3	4	5	6	7	8
Variable	D	DI	DO	DM				
Type	F	F	F	F				
Default	0.0	0.0	0.0	0.0				

Geometry Card 2.

Card 4	1	2	3	4	5	6	7	8
Variable	A0	BI	BO	PD				
Type	F	F	F	F				
Default	0.0	0.0	0.0	0.0				

Preloading Card.

Card 5	1	2	3	4	5	6	7	8
Variable	IPFLAG	XTRAN	YTRAN	ZTRAN	XROT	YROT		
Type	I	F	F	F	F	F		
Default	00	0.0	0.0	0.0	0.0	0.0		

VARIABLE	DESCRIPTION
ID	Bearing ID.
ITYPE	Bearing type: EQ.1: ball bearing EQ.2: roller bearing
N1	Node on centerline of shaft (the shaft rotates).

VARIABLE DESCRIPTION

CID1 Coordinate ID on shaft. The local z axis defines the axis of rotation.

N2 Node on centerline of bearing (the bearing does not rotate). It should initially coincide with N1.

CID2 Coordinate ID on bearing. The local z axis defines the axis of rotation.

NB Number of balls or rollers.

EBALL Young’s modulus for balls or rollers.

PRBALL Poisson’s ratio for balls or rollers.

ERACE Young’s modulus for races.

PRRACE Poisson’s ratio for races.

STRESL Specified value of the bearing stress required to print a warning message that the value has been reached. If it is 0.0, then no message is printed.

D Diameter of balls or rollers.

DI Bore inner diameter.

DO Bore outer diameter.

DM Pitch diameter. If DM is not specified, it is calculated as the average of DI and DO.

A0 Initial contact angle in degrees

BI Inner groove radius to ball diameter ratio for ball bearings and the roller length for roller bearings..

BO Outer race groove radius to ball diameter ratio. Unused for roller bearings.

PD Total radial clearance between the ball bearings and races when no load is applied.

VARIABLE	DESCRIPTION
IPFLAG	Preload flag EQ.0: no preload. EQ.1: displacement preload specified. EQ.2: force preload specified.
XTRAN	Displacement or force preload in the local x direction.
YTRAN	Displacement or force preload in the local y direction.
ZTRAN	Displacement or force preload in the local z direction.
XROT	Angle (in radians) or moment preload in local x direction.
YROT	Angle (in radians) or moment preload in local y direction.

Remarks:

1. If the bearing stress limit parameter, STRESL, is exceeded, a message is written to the messag and d3hsp files. When this value is exceeded there is no change in element behavior.s

2. Use of double precision for solution stability is strongly suggested.

3. A realistic level of bearing damping (which can be included using *ELEMENT_DISCRETE and *MAT_DAMPER_VISCOUS) may be needed for solution stability.

4. Bearing forces can be output in the brngout file, using *DATABASE_BEARING.

ELEMENT_DIRECT_MATRIX_INPUT_*{OPTION}

Available options include:

<BLANK>

BINARY

Purpose: Define an element consisting of mass, damping, stiffness, and inertia matrices in a specified file which follows the format used in the direct matrix input, DMIG, of NASTRAN. The supported format is the type 6 symmetric matrix in real double precision. LS-DYNA supports both the standard and the extended precision formats. The binary format from *CONTROL_IMPLICIT_MODES or *CONTROL_IMPLICIT_STATIC_CONDENSATION is another input option. The mass and stiffness matrices are required. The inertia matrix is required when using *LOAD_BODY_OPTION to correctly compute the action of a prescribed base acceleration on the superelement, otherwise the inertia matrix is unused. The damping matrix is optional. The combination of these matrices is referred to as a superelement. Three input cards are required for each superelement.

The degrees-of-freedom for this superelement may consist of generalized coordinates as well was nodal point quantities. Degrees-of-freedom, defined using *NODE input, are called attachment nodes. Only attachment nodes are included in the output to the ASCII and binary databases.

The matrices for a given superelement can be of different order. However, the explicit integration scheme requires the inversion of the union of the element mass matrix and nodal masses associated with attachment nodes. Any degree of freedom included in the other (stiffness, damping, inertia) matrices but without nonzero columns in the combined mass matrix will be viewed as massless and constrained not to move. After deleting zero rows and columns the combined mass matrix is required to be positive definite.

The inertia matrix is required to have 3 columns which corresponds to the 3 global coordinates. It is used to compute the forces acting on the superelement by multiplying the inertia matrix times the gravitational acceleration specified via *LOAD_BODY_OPTION.

There is no assumption made on the order of the matrices nor the sparse matrix structure of the element matrices except that they are symmetric and the combined mass matrix is invertible as described above.

Multiple elements may be input using *ELEMENT_DIRECT_MATRIX_INPUT. They may share attachment nodes with other direct matrix input elements. Only *BOUNDARY_PRESCRIBED_MOTION and global constraints imposed via*NODE or *BOUND-

ARY_SPC on attachment nodes can be applied in explicit applications. Implicit applications can have additional constraints on attachment nodes.

Card 1	1	2	3	4	5	6	7	8
Variable	EID	IFRMT						
Type	I	0						

Card 2	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	C							

Card 3	1	2	3	4	5	6	7	8
Variable	MASS	DAMP	STIF	INERT				
Type	C	C	C	C				

VARIABLE	DESCRIPTION
EID	Super element ID.
IFRMT	Format: EQ.0: standard format NE.0: extended precision format
MASS	Name of mass matrix in the file defined by FILENAME. This filename should be no more than eight characters to be compatible with NASTRAN.
DAMP	Name of damping matrix in the file defined by FILENAME. This filename should be no more than eight characters to be compatible with NASTRAN.

VARIABLE	DESCRIPTION
STIF	Name of stiffness matrix in the file defined by FILENAME. This filename should be no more than eight characters to be compatible with NASTRAN.
INERT	Name of inertia matrix in the file defined by FILENAME. This filename should be no more than eight characters to be compatible with NASTRAN. This file must be present when *LOAD_BODY is used to put gravitational forces on the model.

ELEMENT_DISCRETE_*{OPTION}

Available options include:

<BLANK>

LCO

Purpose: Define a discrete (spring or damper) element between two nodes or a node and ground.

An option, LCO, is available for using a load curve(s) to initialize the offset to avoid the excitation of numerical noise that can sometimes result with an instantaneous imposition of the offset. This can be done using a single curve at the start of the calculation or two curves where the second is used during dynamic relaxation prior to beginning the transient part. In the latter case, the first curve would simply specify the offset as constant during time. If the LCO option is active, a second card is read.

Beam type 6, see *ELEMENT_BEAM and SECTION_BEAM, may be used as an alternative to *ELEMENT_DISCRETE and *SECTION_DISCRETE, and is recommended if the discrete element's line of action is not node N1 to N2, i.e., if VID.NE.0.

> **NOTE:** The discrete elements enter into the time step calculations. Care must be taken to ensure that the nodal masses connected by the springs and dampers are defined and unrealistically high stiffness and damping values must be avoided. **All rotations are in radians**.

Card	1	2	3	4	5	6	7	8	9	10
Variable	EID	PID	N1	N2	VID	S		PF	OFFSET	
Type	I	I	I	I	I	F		I	F	
Default	none	none	none	none	0	1.		0	0	

Offset Load Curve Card. Additional card for LCO keyword option.

Card 2	1	2	3	4	5	6	7	8
Variable	LCID	LCIDDR						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
EID	Element ID. A unique number is required. Since null beams are created for visualization, this element ID should not be identical to element ID's defined for ELEMENT_BEAM and ELEMENT_-SEATBELT.
PID	Part ID, see *PART.
N1	Nodal point 1.
N2	Nodal point 2. If zero, the spring/damper connects node N1 to ground.
VID	Orientation option. The orientation option should be used cautiously since forces, which are generated as the nodal points displace, are not orthogonal to rigid body rotation unless the nodes are coincident.. The type 6, 3D beam element, is recommended when orientation is required with the absolute value of the parameter SCOOR set to 2 or 3, since this option avoids rotational constraints. EQ.0: the spring/damper acts along the axis from node N1 to N2, NE.0: the spring/damper acts along the axis defined by the orientation vector, VID defined in the *DEFINE_SD_ORIENTATION section.
S	Scale factor on forces.
PF	Print flag: EQ.0: forces are printed in DEFORC file, EQ.1: forces are not printed DEFORC file.

VARIABLE	DESCRIPTION
OFFSET	Initial offset. The initial offset is a displacement or rotation at time zero. For example, a positive offset on a translational spring will lead to a tensile force being developed at time zero. Ignore this input if LCID is defined below.
LCID	Load curve ID defining the initial OFFSET as a function of time. Positive offsets correspond to tensile forces, and, likewise negative offsets result in compressive forces.
LCIDDR	Load curve ID defining OFFSET as a function of time during the dynamic relaxation phase.

ELEMENT_DISCRETE_SPHERE_*{OPTION}

Available options include:

<BLANK>

VOLUME

Purpose: Define a discrete spherical element for discrete element method (DEM) calculations. Currently, LS-DYNA's implementation of the DEM supports only spherical particles, as discrete element spheres (DES). Each DES consists of a single node with its mass, mass moment of inertia, and radius defined by the input below. Initial coordinates and velocities are specified via the nodal data. The element ID corresponds to the ID of the node. The discrete spherical elements are visualized in LS-PrePost using the same options as the SPH elements.

If the VOLUME option is active, the fields for MASS and INERTIA are based on per unit density.

Please note, the DES part requires *PART, *SECTION, and *MAT keywords. The element type and formulation values in *SECTION are ignored. DEM retrieves the bulk modulus from the *MAT input for coupling stiffness and time step size evaluation, and density from the *MAT input if VOLUME is used to calculate the proper mass. *MAT_-ELASTIC and *MAT_RIGID are most commonly used, but other material models are also permissible.

Card 1	1	2	3	4	5	6	7	8
Variable	NID	PID	MASS/ VOLUME	INERTIA	RADIUS			
Type	I	I	F	F	F			
Default	none	none	none	none	none			

VARIABLE	DESCRIPTION
NID	DES Node ID.
PID	DES Part ID, see *PART.

VARIABLE	DESCRIPTION
MASS/ VOLUME	If the VOLUME keyword option is set, then VOLUME and the mass is calculated from material density, $M = \text{MASS} \times \rho_{\text{mat}}.$ Otherwise this entry is interpreted as mass.
INERTIA	Mass moment of inertia. If the VOLUME option is active, the actual inertia is calculated from material density, $I = \text{INERTIA} \times \rho_{\text{mat}}.$
RADIUS	Particle radius. The particle radius is used for defining contact between particles.

*ELEMENT_GENERALIZED_SHELL

Purpose: Define a general 3D shell element with an arbitrary number of nodes. The formulation of this element is specified in *DEFINE_ELEMENT_GENERALIZED_-SHELL, which is specified through the part ID (see *PART) and the section ID (see *SECTION_SHELL). For an illustration of this referencing, see Figure 15-31. Using this generalized shell implementation allows a rapid prototyping of new shell element formulations without further coding.

The element formulation used in *SECTION_SHELL needs to be greater or equal than 1000.

Card 1	1	2	3	4	5	6	7	8
Variable	EID	PID	NMNP					
Type	I	I	I					
Default	none	none	none					

Connectivity Cards. Define the connectivity of the element by specifying NMNP-nodes (up to eight nodes per card). Include as many cards as needed. For example, for NMNP = 10, the deck should include two additional cards.

Card 2	1	2	3	4	5	6	7	8
Variable	N1	N2	N3	N4	N5	N6	N7	N8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE **DESCRIPTION**

EID Element ID. Chose a unique number with respect to other elements.

PID Part ID, see *PART.

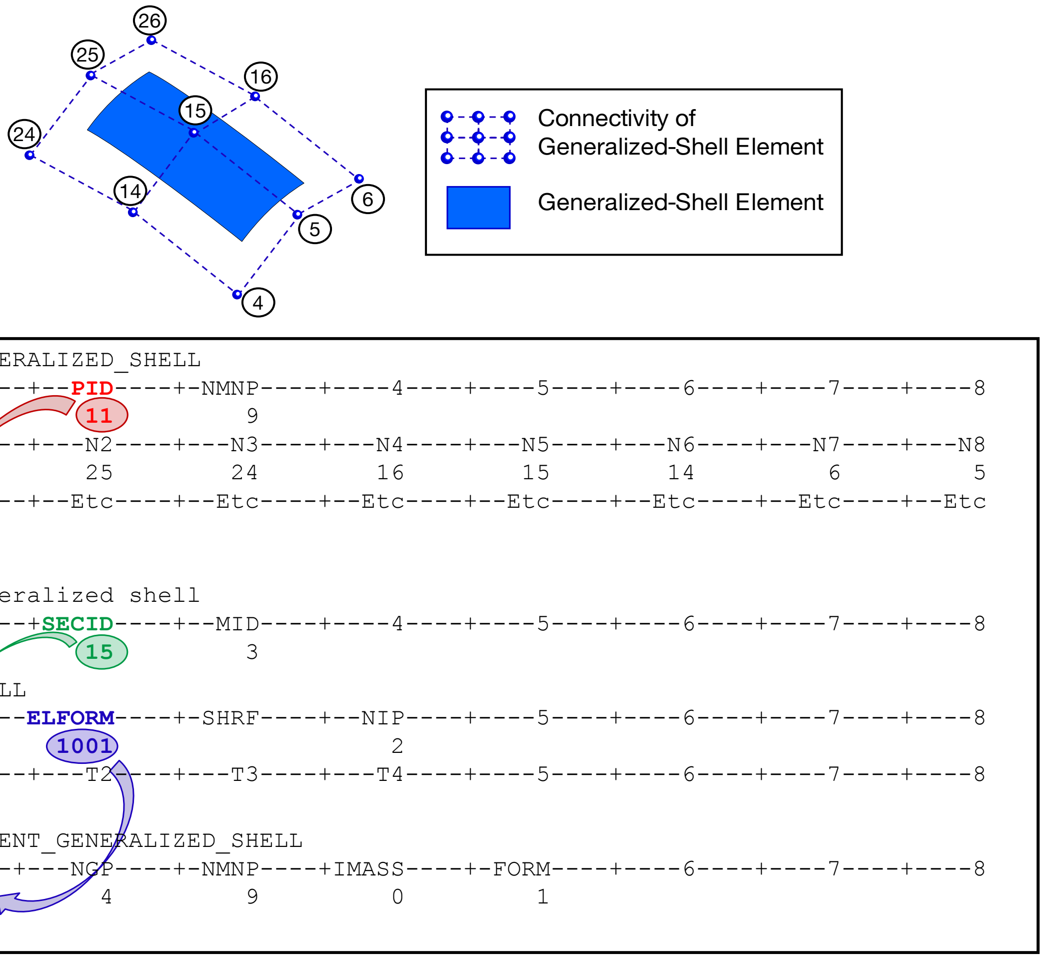
NMNP Number of nodes to define this element.

N_i Nodal point *i* (defined via *NODE) to define connectivity of this element.

Remarks:

1. For post-processing and the treatment of contact boundary conditions, the use of interpolation shell elements (see *ELEMENT_INTERPOLATION_SHELL and *CONSTRAINED_NODE_INTERPOLATION) is necessary.

2. The definition of the connectivity of the element is basically arbitrary but it has to be in correlation with the definition of the element formulation in *DEFINE_ELEMENT_GENERALIZED_SHELL.


26
25
16
15
24
14
6
5
4
Connectivity of Generalized-Shell Element
Generalized-Shell Element

```
*ELEMENT_GENERALIZED_SHELL
$---+--EID----+--PID----+-NMNP----+----4----+----5----+----6----+----7----+----8
         1        11         9
$---+---N1----+---N2----+---N3----+---N4----+---N5----+---N6----+---N7----+---N8
        26        25        24        16        15        14         6         5
$---+---N9----+--Etc----+--Etc----+--Etc----+--Etc----+--Etc----+--Etc----+--Etc
         4

*PART
Part for generalized shell
$---+--PID----+SECID----+--MID----+----4----+----5----+----6----+----7----+----8
        11        15         3

*SECTION_SHELL
$---+SECID----ELFORM----+-SHRF----+--NIP----+----5----+----6----+----7----+----8
        15      1001                   2
$---+---T1----+---T2----+---T3----+---T4----+----5----+----6----+----7----+----8
       1.0

*DEFINE_ELEMENT_GENERALIZED_SHELL
$---ELFORM----+---NGP----+-NMNP----+IMASS----+-FORM----+----6----+----7----+----8
      1001         4         9         0         1
...
```

Figure 17-3. Example of the connection between *ELEMENT_GENERALIZED_SHELL and *DEFINE_ELEMENT_GENERALIZED_SHELL.

*ELEMENT_GENERALIZED_SOLID

Purpose: Define a general 3D solid element with an arbitrary number of nodes. The formulation of this element is specified in *DEFINE_ELEMENT_GENERALIZED_SOLID, which is referenced through the part ID (see *PART) and the section ID (see *SECTION_SOLID). For an illustration of this referencing, see Figure 17-4. Using this generalized solid implementation allows a rapid prototyping of new solid element formulations without further coding.

The element formulation used in *SECTION_SOLID needs to be greater or equal than 1000.

Card 1	1	2	3	4	5	6	7	8
Variable	EID	PID	NMNP					
Type	I	I	I					
Default	none	none	none					

Connectivity Cards. Define the connectivity of the element by specifying NMNP-nodes (up to eight nodes per card). Include as many cards as needed. For example, for NMNP = 10, the deck should include two additional cards.

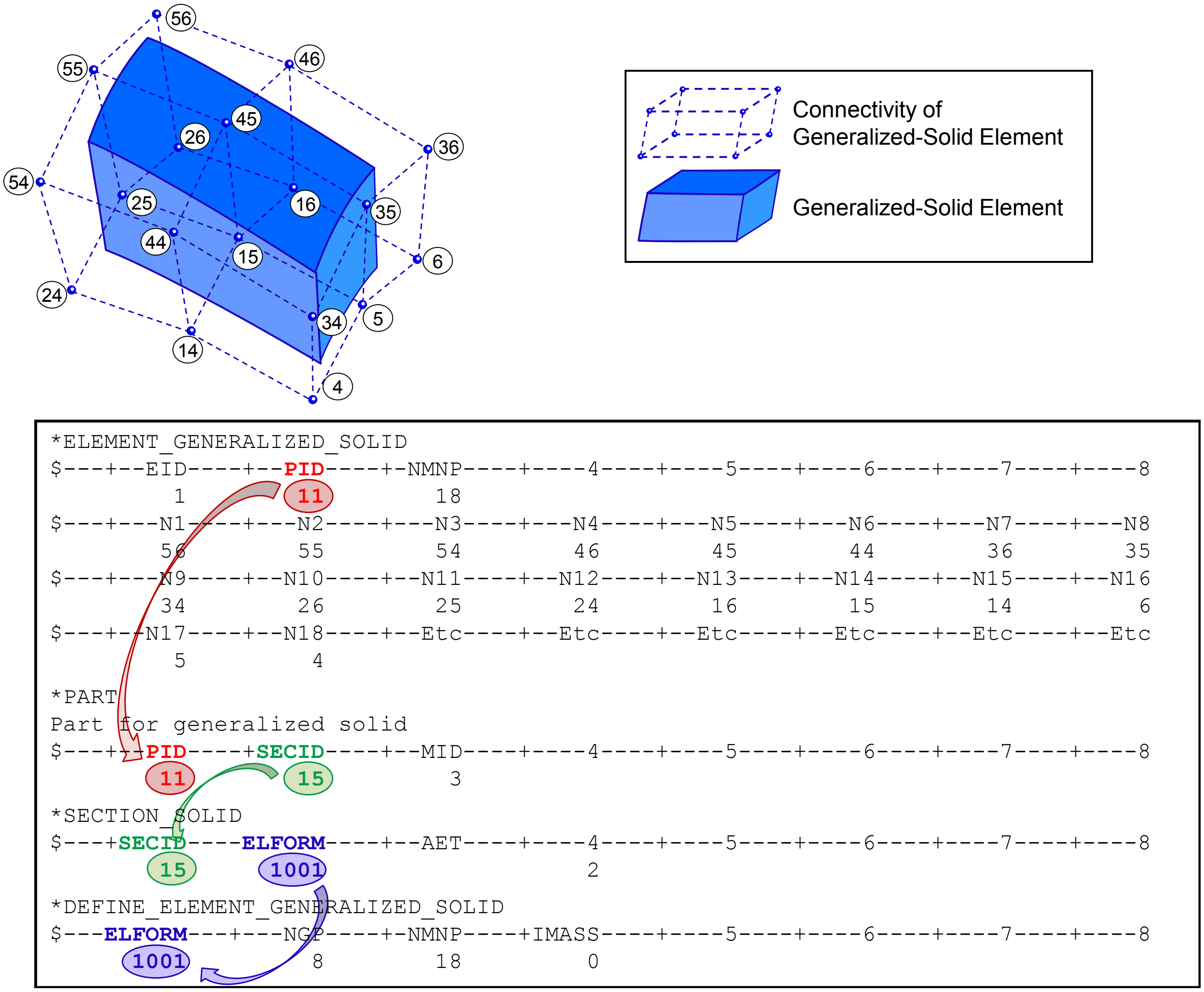
Card 2	1	2	3	4	5	6	7	8
Variable	N1	N2	N3	N4	N5	N6	N7	N8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
EID	Element ID. Chose a unique number with respect to other elements.
PID	Part ID, see *PART.
NMNP	Number of nodes to define this element.
N_i	Nodal point *i* (defined via *NODE) to define connectivity of this element.

Remarks:

1. For post-processing the use of interpolation solid elements (see *ELEMENT_INTERPOLATION_SOLID and *CONSTRAINED_NODE_INTERPOLATION) is necessary.

2. The definition of the connectivity of the element is basically arbitrary but it has to be in correlation with the definition of the element formulation in *DEFINE_ELEMENT_GENERALIZED_SOLID.


56
55
46
45
26
36
54
25
16
35
44
15
6
24
5
34
14
4
Connectivity of Generalized-Solid Element
Generalized-Solid Element

```
*ELEMENT_GENERALIZED_SOLID
$---+--EID----+--PID----+-NMNP----+----4----+----5----+----6----+----7----+----8
         1        11        18
$---+---N1----+---N2----+---N3----+---N4----+---N5----+---N6----+---N7----+---N8
        56        55        54        46        45        44        36        35
$---+---N9----+--N10----+--N11----+--N12----+--N13----+--N14----+--N15----+--N16
        34        26        25        24        16        15        14         6
$---+--N17----+--N18----+--Etc----+--Etc----+--Etc----+--Etc----+--Etc----+--Etc
         5         4

*PART
Part for generalized solid
$---+--PID----+SECID----+--MID----+----4----+----5----+----6----+----7----+----8
        11        15         3
*SECTION_SOLID
$---+SECID----ELFORM----+--AET----+----4----+----5----+----6----+----7----+----8
        15      1001                   2
*DEFINE_ELEMENT_GENERALIZED_SOLID
$---ELFORM---+---NGP----+-NMNP----+IMASS----+----5----+----6----+----7----+----8
      1001         8        18         0
```

Figure 17-4. Example of the connection between *ELEMENT_GENERALIZED_SOLID and *DEFINE_ELEMENT_GENERALIZED_SOLID.

ELEMENT_INERTIA_*{OPTION}

Available options include:

<BLANK>

OFFSET

Purpose: to allow the lumped mass and inertia tensor to be offset from the nodal point. The nodal point can belong to either a deformable or rigid node.

Card 1	1	2	3	4	5	6	7	8	9	10
Variable	EID	NID	CSID							
Type	I	I	I							
Default	none	none	none							
Remarks			1							

Card 2	1	2	3	4	5	6	7	8
Variable	IXX	IXY	IXZ	IYY	IYZ	IZZ	MASS	
Type	F	F	F	F	F	F	F	
Default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
Remarks		2	2		2			

Offset Card. Additional card for offset keyword option.

Card 3	1	2	3	4	5	6	7	8
Variable	X-OFF	Y-OFF	Z-OFF					
Type	F	F	F					
Default	0.	0.	0.					
Remarks		2	2					

VARIABLE	DESCRIPTION
EID	Element ID. A unique number must be used.
NID	Node ID. Node to which the mass is assigned.
CSID	Coordinate system ID EQ.0: global inertia tensor GE.1: principal moments of inertias with orientation vectors defined by Coordinate system CSID. See *DEFINE_COORDINATE_SYSTEM and *DEFINE_COORDINATE_VECTOR.
IXX	*xx*component of inertia tensor.
IXY	*xu* component of inertia tensor.
IXZ	*xz* component of inertia tensor.
IYY	*yy* component of inertia tensor.
IYZ	*yz* component of inertia tensor.
IZZ	*zz* component of inertia tensor.
MASS	Lumped mass
X-OFF	*x*-offset from nodal point.
Y-OFF	*y*-offset from nodal point.
Z-OFF	*z*-offset from nodal point.

Remarks:

1. The coordinate system cannot be defined for this element using the option,

   *DEFINE_COORDINATE_NODE.

2. If CSID is defined then IXY, IXZ and IYZ are set to zero. The nodal inertia tensor must be positive definite, i.e., its determinant must be greater than zero, since its inverse is required. This check is done after the nodal inertia is added to the defined inertia tensor.

*ELEMENT_INTERPOLATION_SHELL

Purpose: With the definition of interpolation shells, the stresses and other solution variables can be interpolated from the generalized shell elements (see *ELEMENT_GENERALIZED_SHELL and *DEFINE_ELEMENT_GENERALIZED_SHELL) permitting the solution to be visualized using standard 4-noded shell elements with one integration point (one value of each solution variable per interpolation shell). The definition of the interpolation shells is based on interpolation nodes (see *CONSTRAINED_NODE_INTERPOLATION). The connections between these various keywords are illustrated in Figure 17-5.

Card 1	1	2	3	4	5	6	7	8
Variable	EIDS	EIDGS	NGP					
Type	I	I	I					
Default	none	none	none					

Weighting Factor Cards. These cards set the weighting factors used for interpolating the solution onto the center of *this* interpolation shell. Set one weight for each of the NGP integration points. Each card can accommodate 4 points; define as many cards as necessary. As an example, for NGP = 10 three cards are required.

Card 2	1	2	3	4	5	6	7	8
Variable	IP1	W1	IP2	W2	IP3	W3	IP4	W4
Type	I	F	I	F	I	F	I	F
Default	none	none	none	none	none	none	none	none

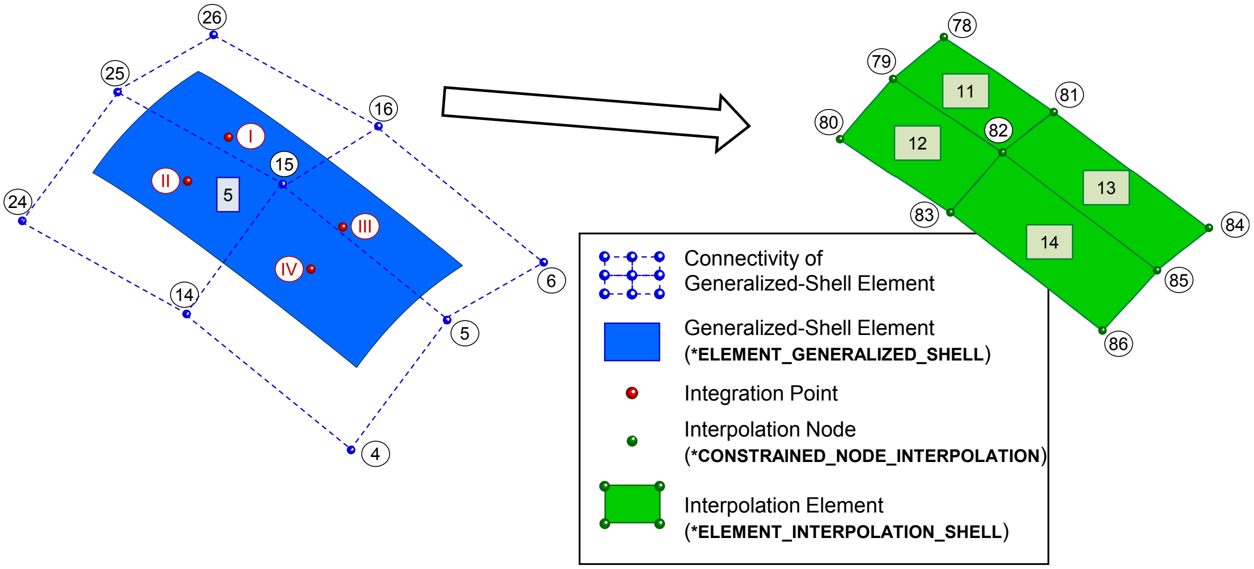
VARIABLE	DESCRIPTION
EIDS	Element ID of the interpolation shell. This needs to coincide with a proper definition of a 4-noded shell element (*ELEMENT_SHELL) using interpolation nodes (*CONSTRAINED_NODE_INTERPOLATION).
EIDGS	Element ID of the master element defined in *ELEMENT_GENERALIZED_SHELL.

VARIABLE	DESCRIPTION
NGP	Number of in-plane integration points of the master element.
IP_i	Integration point number (1 to NGP) in the order how they were defined in *DEFINE_ELEMENT_GENERALIZED_SHELL.
W_i	Interpolation weight of integration point *i*.

Remarks:

1. For each interpolation shell element, one single value (v_{IS}) of a solution variable is interpolated based on values at the integration points (v_i) of the master element (*ELEMENT_GENERALIZED_SHELL) and the appropriate weighting factors (w_i). The interpolation is computed as follows: $v_{IS} = \sum_{i=1}^{NGP} w_i v_i$.

2. To use *ELEMENT_INTERPOLATION_SHELL, ELFORM = 98 has to be used in *SECTION_SHELL


26
25
16
15
I
II
5
III
24
IV
14
6
5
4
78
79
11
81
80
82
12
13
83
84
14
85
86
Connectivity of
Generalized-Shell Element
Generalized-Shell Element
(*ELEMENT_GENERALIZED_SHELL)
Integration Point
Interpolation Node
(*CONSTRAINED_NODE_INTERPOLATION)
Interpolation Element
(*ELEMENT_INTERPOLATION_SHELL)

```
*CONSTRAINED_NODE_INTERPOLATION
$---+--NID----+NUMMN----+----3----+----4----+----5----+----6----+----7----+----8
        78         4
$---+--MN1----+---W1----+--MN2----+---W2----+--MN3----+---W3----+--MN4----+---W4
        26      0.35        25      0.32        15      0.18        16      0.15
*ELEMENT_SHELL
$--+-EID---+ PID---+- N1---+--N2---+--N3---+--N4---+--N5---+--N6---+--N7---+--N8
      11       33      78      79      82      81
*PART
Part for interpolation shell
$---+--PID----+SECID----+--MID----+----4----+----5----+----6----+----7----+----8
        33        45         3
*SECTION_SHELL
$---+SECID----ELFORM----+-SHRF----+--NIP----+----5----+----6----+----7----+----8
        45        98                   2
$---+---T1----+---T2----+---T3----+---T4----+----5----+----6----+----7----+----8
       1.0
*ELEMENT_INTERPOLATION_SHELL
$-----EIDS---+-EIDGS----+--NGP----+----4----+----5----+----6----+----7----+----8
        11         5         4
$---+--IP1---+----W1----+--IP2----+---W2----+--IP3----+---W3----+--IP4----+---W4
         1       0.5         2       0.2         3       0.2         4       0.1
```

Figure 17-5. Example for *ELEMENT_INTERPOLATION_SHELL.

*ELEMENT_INTERPOLATION_SOLID

Purpose: With the definition of interpolation solids, the stresses and other solution variables can be interpolated from the generalized solid elements (see *ELEMENT_-GENERALIZED_SOLID and *DEFINE_ELEMENT_GENERALIZED_SOLID) permitting the solution to be visualized using standard 8-noded solid elements with one integration point (one value of each solution variable per interpolation solid). The definition of the interpolation solids is based on interpolation nodes (see *CONSTRAINED_NODE_INTERPOLATION). The connection between these various keywords are illustrated in Figure17-6.

Card 1	1	2	3	4	5	6	7	8
Variable	EIDS	EIDGS	NGP					
Type	I	I	I					
Default	none	none	none					

Weighting Factor Cards. These cards set the weighting factors used for interpolating the solution onto the center of *this* interpolation solid. Set one weight for each of the element's NGP integration points. Each card can accommodate 4 points; define as many cards as necessary. As an example, for NGP = 10 three cards are required.

Cards	1	2	3	4	5	6	7	8
Variable	IP1	W1	IP2	W2	IP3	W3	IP4	W4
Type	I	F	I	F	I	F	I	F
Default	none	none	none	none	none	none	none	none

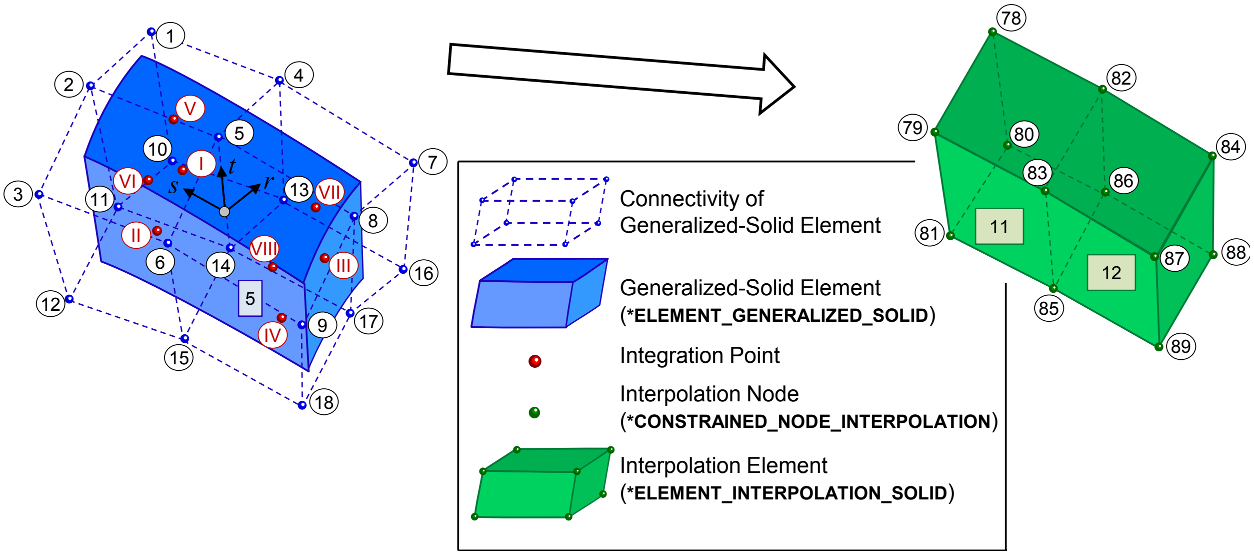
VARIABLE	DESCRIPTION
EIDS	Element ID of the interpolation solid. This needs to coincide with a proper definition of a 8-noded solid element (*ELEMENT_SOLID) using interpolation nodes (*CONSTRAINED_NODE_INTERPOLATION).

VARIABLE	DESCRIPTION
EIDGS	Element ID of the master element defined in *ELEMENT_GENERALIZED_SOLID.
NGP	Number of integration points of the master element.
IP$_i$	Integration point number (1 to NGP) in the order how they were defined in *DEFINE_ELEMENT_GENERALIZED_SOLID.
W$_i$	Interpolation weight of integration point *i*.

Remarks:

1. For each interpolation solid element, one single value (v_{IS}) of a solution variable is interpolated based on values at the integration points (v_i) of the master element (*ELEMENT_GENERALIZED_SOLID) and the appropriate weighting factors (w_i). The interpolation is computed as follows: $v_{IS} = \sum_{i=1}^{NGP} w_i v_i$

2. To use *ELEMENT_INTERPOLATION_SOLID, ELFORM = 98 has to be used in *SECTION_SOLID


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
I
II
III
IV
V
VI
VII
VIII
r
s
t
5
78
79
80
81
82
83
84
85
86
87
88
89
11
12
Connectivity of Generalized-Solid Element
Generalized-Solid Element (*ELEMENT_GENERALIZED_SOLID)
Integration Point
Interpolation Node (*CONSTRAINED_NODE_INTERPOLATION)
Interpolation Element (*ELEMENT_INTERPOLATION_SOLID)

```
*ELEMENT_SOLID
$--+-EID---+ PID---+- N1---+--N2---+--N3---+--N4---+--N5---+--N6---+--N7---+--N8
      11      33
$--+--N1---+--N2---+- N3---+--N4---+--N5---+--N6---+--N7---+--N8---+--N9---+-N10
      80      81      85      86      78      79      83      82

*PART
Part for interpolation solid
$---+--PID----+SECID----+--MID----+----4----+----5----+----6----+----7----+----8
        33        45          3

*SECTION_SOLID
$---+SECID----ELFORM----+--AET----+----4----+----5----+----6----+----7----+----8
        45        98

*ELEMENT_INTERPOLATION_SOLID
$-----EIDS---+-EIDGS----+--NGP----+----4----+----5----+----6----+----7----+----8
        11         5          8
$---+--IP1---+----W1----+--IP2----+---W2----+--IP3----+---W3----+--IP4----+---W4
         1      0.30          2      0.12          3      0.13          4      0.07
$---+--IP5---+----W5----+--IP6----+---W6----+--IP7----+---W7----+--IP8----+---W8
         5      0.20          6      0.08          7      0.07          8      0.03
```

Figure 17-6. Example for *ELEMENT INTERPOLATION SOLID.

*ELEMENT_LANCING

Purpose: This feature models a lancing process during a metal forming process by trimming along a curve. Two types of lancing, instant and progressive, are supported. This keyword is used together with *DEFINE_CURVE_TRIM_3D, and only applies to shell elements. The lanced scraps can be removed (trimming) during or after lancing when used in conjunction with *DEFINE_LANCE_SEED_POINT_COORDINATES, see Trimming.

The element lancing feature is supported in *LS-PrePost 4.3*, under *Application → MetalForming → Easy Setup*.

For each trim include an additional card. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	IDPT	IDCV	IREFINE	SMIN	AT	ENDT	NTIMES	CIVD
Type	I	I	I	F	F	F	I	I
Default	none	none	1	none	none	none	none	none

VARIABLE	DESCRIPTION

IDPT

A flag to indicate if a part to be lanced is a part or a part set.

GT.0: IDPT is the PID of a part to be lanced, see *PART.

LT.0: the absolute value is the part set ID, as in *SET_PART_LIST. This option allows the lancing to be performed across a tailor-welded line.

IDCV

A load curve ID (the variable TCID in *DEFINE_CURVE_TRIM_3D) defining a lancing route (see **Remarks**). XYZ format (TCTYPE = 1) has always been supported, however, IGES format (TCTYPE = 2) is not supported until Revision 110246.

IREFINE

Set IREFINE = 1 to refine elements along lancing route until no adapted nodes exist in the neighborhood. This feature result in a more robust lancing in the form of improved lancing boundary. Available starting in Revision 107708. Values greater than "1" are not allowed. See Figure 17-15 for an example of the mesh refinement.

VARIABLE **DESCRIPTION**

SMIN Minimum element characteristic length to be refined to, to be supported in the future. Currently, one level of refinement will be automatically made.

AT Activation time for lancing operation. This variable needs to be defined for both instant and progressive lancing types (see **Remarks**). If CIVD is defined, AT becomes the distance from punch home position.

ENDT Lancing end time (for progressive lancing only). If CIVD is defined, ENDT becomes the distance from punch home position. Do not define for instant lancing.

NTIMES A progressive lancing operation is evenly divided into NTIMES segments between AT and ENDT; within each segment lancing is done instantly. Do not define for instant lancing.

CIVD The load curve ID (LCID) under the keyword *BOUNDARY_PRESCRIBED_MOTION_RIGID to define the kinematics (time vs. velocity) of the tool, with VAD = 0. Furthermore, when this variable is used, AT and ENDT will become the distances from punch bottom position. See an example inLance-trimming with negative IDPT, IGES curve, a seed node and CIVD.

Remarks:

Lancing the blank during forming at strategic locations under controlled conditions alleviates thinning and necking of sheet metal panels. Typically, the blank is lanced in the last few millimeters before the punch reaches its home position. Being an unstable process, lancing is not favored by all stampers, nevertheless many users have devised process which would be impossible without lancing.

The benefits of lancing are illustrated in Figure 17-7. In this figure two closed-loop holes are instantly lanced each along the C-pillar top (window opening area) and bottom (window regulator area) to improve the formability at those two corners. The right panel of Figure 17-7 is lanced and suffers less thinning compared to the no-lancing case, which is shown in the left-panel. This keyword offers two types of lancing operations:

1. **Instant lancing.** Instant lancing cuts the sheet metal once along the defined curve at a time specified in the AT field.

2. **Progressive lancing.** The cut is spatially divided into NTIMES sub-lances traveling along the curve in the direction of definition. See Figure 17-10. Progressive lancing starts at AT and ends at ENDT thereby achieving a gradual and even release along the curve.

Modeling information:

Some modeling guidelines and limitations are listed below:

1. Both closed-loop (Figure 17-8) and open-loop (Figure 17-9) lancing curves are supported. Lancing curve may not cross each other, and cross itself.

2. Since progressive lancing starts from the beginning of the curve and proceeds towards the end, the direction of the curve needs to be defined to match the direction of the physical cut (Figure 17-10). The direction can be set using *LS-PrePost*. The menu option *GeoTol*→ *Measure* with the *Edge* box checked can be used to show the direction of the curve. If the direction is not as desired, *GeoTol* → *Rever* can be used to reverse the direction.

3. The effect of NTIMES can be seen in Figure 17-11. Compared with NTIMES of 6, setting NTIMES to a value of 20 results in a smoother lancing boundary and less stress concentration along the separated route.

4. Although the IGES format curve is supported in the keyword *DEFINE_-CURVE_TRIM_3D, curves defined with the keyword and used for lancing must be specified using only the XYZ format (TCTYPE = 1 or 0). The manual entry in the keyword manual for the *INTERFACE_BLANKSIZE_DEVELOPMENT keyword outlines a procedure for converting an IGES file into the required XYZ format. Note the IGES format support for lancing is enabled starting in Revision 110246.

5. The first two points and as well as the last two points of the any progressive lancing curve must be separated so that LS-DYNA can correctly determine the direction of the curve.

6. The lancing curve needs to be much longer than the element sizes in the lancing area.

7. To prevent mesh distortion at the end of the lancing route ENDT must be defined to be less than the simulation completion time (slightly less is sufficient).

8. As currently implemented, lancing is assumed to be in the Z-direction. This keyword does not model lancing along the draw wall with surface normals nearly perpendicular to the Z-axis.

9. Tailor-welded blanks are supported; however, the lancing route should not cross the laser line, as currently only one part can be defined with one lancing curve.

10. Both *PARAMETER and *PARAMETER_EXPRESSION are supported for BT and DT as of Revision 92335. This makes it possible for users to input distance from punch home as onset of the lancing. Refer to Figure 17-13, where a punch's velocity profile is shown, the lancing activation time "at", is calculated based on the distance to home, "dhome" (the shaded area), punch travel velocity "vdraw", and total simulation time "ENDTIME". The variable CIVD implemented in Rev 110173 removes the need for the calculation of AT from punch distance to home.

11. Mesh adaptivity (*CONTROL_ADAPTPIVE and the parameter "ADPOPT" under *PART) must be turned on during lancing.

12. All trim curves used to define lancing routes (*DEFINE_CURVE_TRIM_3D) must be placed after all other curves in the input deck. Furthermore, no curves defined by *DEFINE_CURVE_TRIM_3D that are not used for lancing should be present anywhere in the input deck. Note this restriction is removed starting in Revision 110316.

13. Only *DEFINE_CURVE_TRIM_3D (not _NEW) is supported. If defined curve is far away from the blank it will be projected in Z-direction onto the blank.

Application example:

A partial deck implementing instant lancing is listed below. A blank having a PID of 8 is being lanced along curves #119 and #202 instantly at 0.05 and 0.051 seconds, respectively.

```
*ELEMENT_LANCING
$     IDPT      IDCV   IREFINE      SMIN        AT      ENDT    NTIMES
         8       119                          0.0500
         8       202                          0.0510
*DEFINE_CURVE_TRIM_3D
$#    tcid    tctype      tflg      tdir     tctol      toln     nseed
       119         1         1               0.100         1
$#                  cx                  cy                  cz
           172.99310           42.632320           43.736160
           175.69769          -163.08299           46.547531
           177.46982          -278.03793           49.138161
           186.82404          -303.67191           51.217964
           205.16177          -315.33484           53.299248
               ⋮                   ⋮                   ⋮
*DEFINE_CURVE_TRIM_3D
$#    tcid    tctype      tflg      tdir     tctol      toln     nseed
       202         1         1               0.100         1
$#                  cx                  cy                  cz
           187.46982          -578.73793           89.238161
           168.88404          -403.97191           61.417964
```

```
          215.18177          -215.03484           73.899248
               ⋮                   ⋮                    ⋮
```

A partial keyword deck implementing two progressive lances is listed below. Both lances travel along paths starting at the same coordinate value. A sheet blank having a part ID of 9 is progressively lanced along both curves (IDCV = 1 and 2) as defined by *DEFINE_CURVE_TRIM_3D. Both lancing operations commence at 0.05 seconds and finish at 0.053 seconds with 20 cuts along each curve in opposite direction. The lancing results are shown in Figure 17-12. Note that the termination time is 53.875 seconds, which is slightly larger than the ENDT.

```
*ELEMENT_LANCING
$     IDPT      IDCV   IREFINE      SMIN        AT      ENDT    NTIMES
         9         1                        0.0500   5.3E-02        20
         9         2                        0.0500   5.3E-02        20
*DEFINE_CURVE_TRIM_3D
$#    tcid    tctype      tflg      tdir     tctol      toln     nseed
         1         1         1               0.100         1
$#                 cx                  cy                  cz
          172.99310           42.632320           43.736160
          175.69769          -163.08299           46.547531
          177.46982          -278.03793           49.138161
          186.82404          -303.67191           51.217964
          205.16177          -315.33484           53.299248
          223.13152          -308.03534           54.193089
          234.96263          -290.49695           54.885273
          222.03900          -270.08289           53.163551
          199.31226          -251.27985           50.401234
               ⋮                   ⋮                    ⋮
*DEFINE_CURVE_TRIM_3D
         2         1         1               0.100         1
$#                 cx                  cy                  cz
          172.99310           42.632320           43.736160
          171.33121            47.22141           42.513367
          171.28690           128.84601           43.032799
          176.89932           149.39539           43.495331
          192.41418           159.53757           44.756699
          208.39861           158.93469           45.878036
          218.10101           149.34409           47.128345
          218.34503           135.23810           47.682144
          209.05414           122.82422           46.616959
          190.19659           117.66074           44.858204
               ⋮                   ⋮                    ⋮
```

Trimming after lancing:

As shown in Figure 17-14 and the following partial keyword file, the lanced scraps can be removed (trimming) after a lancing simulation. An extra keyword, *DEFINE_LANCE_SEED_POINT_COORDINATES is needed to define the portion that would remain after the lancing and trimming. It should be obvious that the lancing curve defined by *DEFINE_CURVE_TRIM_3D must form a closed loop. The following example will trim a part ID 9 with a fully enclosed lancing curve #1, at time = 0.049 seconds. Since the termination time is 0.0525 seconds, the scrap will be deleted (trimmed off) before the simulation ends. The scrap, enclosed by the curve, is located outside of the seed node, defined by *DEFINE_LANCE_SEED_POINT_COORDINATES (starting in Revision 107262).

```
*CONTROL_TERMINATION
0.0525
*ELEMENT_LANCING
$      IDPT      IDCV   IREFINE      SMIN        AT      ENDT    NTIMES
          9         1                        0.0490
*DEFINE_CURVE_TRIM_3D
$#    tcid    tctype      tflg      tdir     tctol      toln     nseed
         1         1         1               0.100         1
$#                cx                  cy                  cz
           172.99310           42.632320           43.736160
           175.69769          -163.08299           46.547531
           177.46982          -278.03793           49.138161
           186.82404          -303.67191           51.217964
           205.16177          -315.33484           53.299248
           223.13152          -308.03534           54.193089
               ⋮                   ⋮                   ⋮
           172.99310           42.632320           43.736160
           172.99310           42.632320           43.736160
*DEFINE_LANCE_SEED_POINT_COORDINATES
$    NSEED        X1        Y1        Z1        X2        Y2        Z2
         1    -289.4     98.13  2354.679
```

This feature also makes it possible to combine the trimming process together with a forming simulation, saving a trimming step in a line-die process simulation by skipping writing out a formed dynain file and reading in the same file for the trimming simulation.

Lance-trimming with negative IDPT, IGES curve, a seed node and CIVD:

The following partial keyword input shows instant lance-trimming across the weld line of a tailor-welded blank using the part set ID "blksid" 100, which consists of PIDs 1 and 9. The part set ID used for *ELEMENT_LANCING input is "idpt", which is set as the negative of blksid (-100). The lance-trimming curve ID 1117 is defined using the file lance4.iges in IGES format (TCTYPE = 2). The variable CIVD is referred to load curve ID 12, which is the kinematic curve for the punch. The lancing starts at 15.5 mm away from punch bottom (AT = 15.5). A lance seed coordinate (-382.0, -17.0, 76.0) is defined using the keyword *DEFINE_LANCE_SEED_POINT_COORDINATES, resulting in the lanced scrap piece being removed after lancing.

```
*PARAMETER
I blk1pid          1
I blk2pid          9
I blksid         100
*SET_PART_LIST
&blksid
&blk1pid    &blk2pid
*PARAMETER_EXPRESSION
I idpt        -1*blksid
*ELEMENT_LANCING
$      IDPT      IDCV   IREFINE      SMIN        AT      ENDT    NTIMES      CIVD
      &idpt      1117         1                15.5                          1115
*DEFINE_CURVE_TRIM_3D
```

```
$#    tcid    tctype      tflg      tdir     tctol      toln    nseed1    nseed2
      1117         2         1         0    0.1000         0
lance4.iges
*DEFINE_LANCE_SEED_POINT_COORDINATES
$     NSUM        X1        Y1        Z1        X2        Y2        Z2
         1  -382.000   -17.000      76.0
*BOUNDARY_PRESCRIBED_MOTION_RIGID
$   TYPEID       DOF       VAD      LCID        SF       VID     DEATH     BIRTH
  &udiepid         3         0      1113                      &clstime
  &bindpid         3         0      1114                      &clstime
  &udiepid         3         0      1115                      &endtime  &clstime
  &bindpid         3         0      1115                      &endtime  &clstime
```

Revision information:

This feature is available starting in Revision 83562 for SMP and in Revision 94383 for MPP, in explicit dynamic calculation only. Later revisions incorporate various improvements. The list below provides revision information:

14. Revision 92335: support of *PARAMETER and *PARAMETER_EXPRESSION.

15. Revision 94383: MPP support of lancing is available.

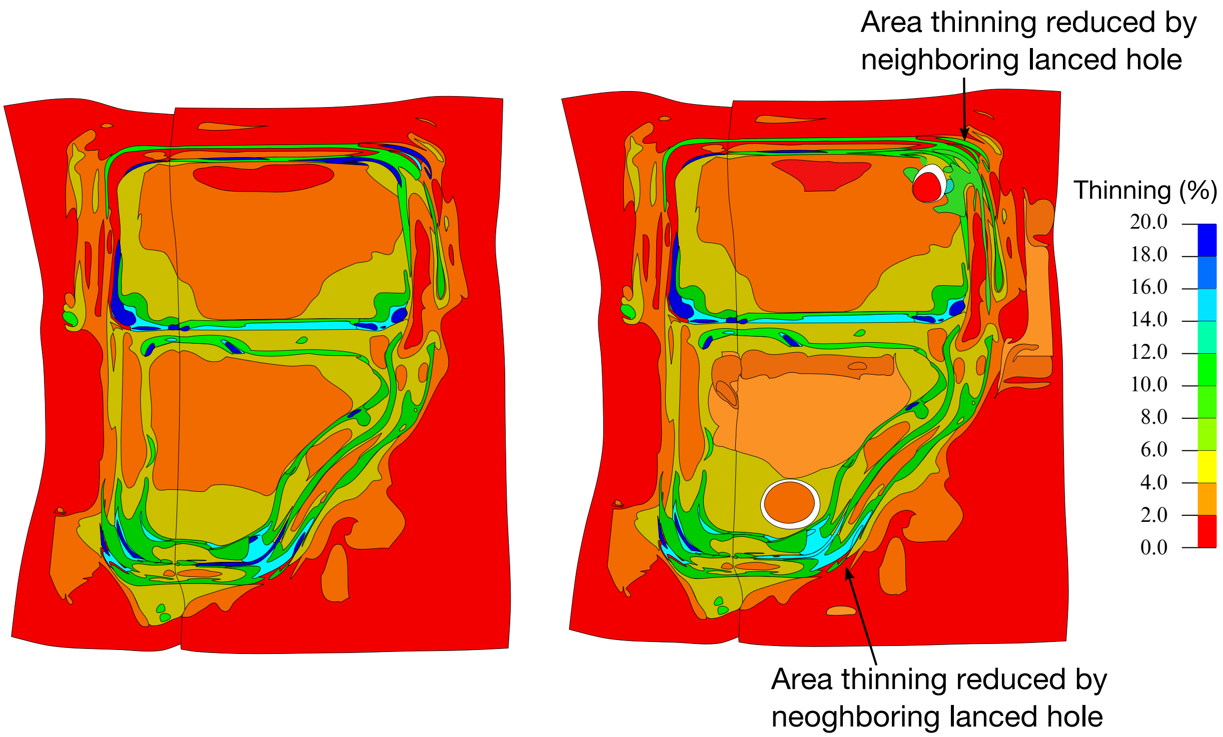
16. Revision 107708: support IREFINE = 1.

17. Revision 107262: lancing with trimming is supported.

18. Revision 110173: CIVD is supported, and AT and ENDT become distances from punch home if CIVD is activated.

19. Revision 110177: support negative IDPT for part set ID, enabling lancing across the laser welded line.

20. Revision 110246: lancing curve definition in IGES format is supported.



Areas of high thinning (in blue) - without lancing

Thinning contour with lancing in upper and lower C-pillar corners

Figure 17-7. Thinning improvement on a door inner as a result of lancing at the upper and lower corner of the C-pillar.

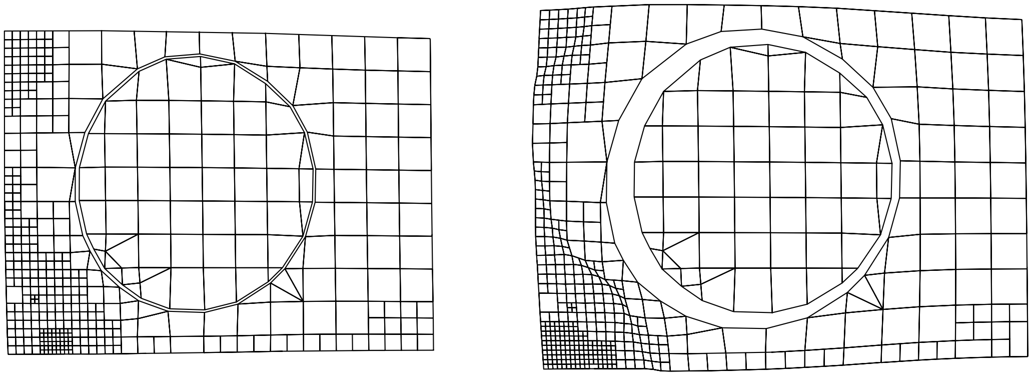


Figure 17-8. Instant lancing – closed-loop hole. The left mesh is immediately after AT while the right one is at punch home.

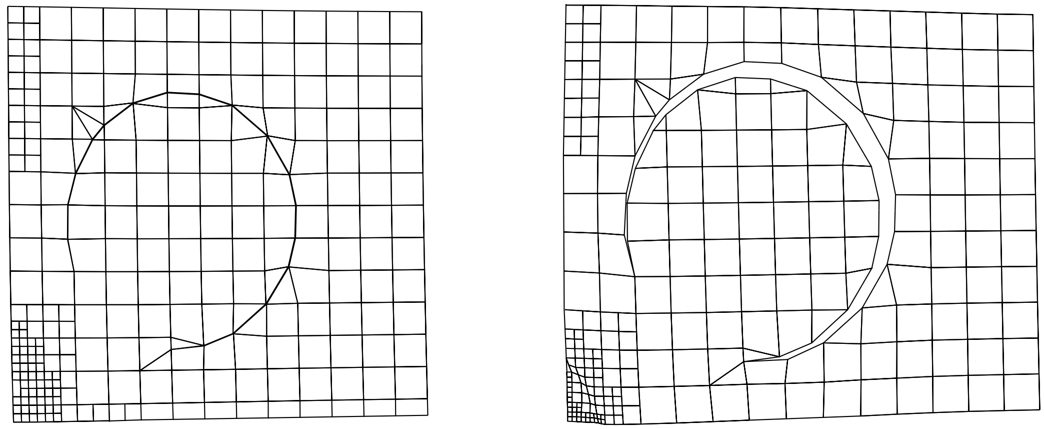


Figure 17-9. Instant lancing – open-loop hole. The left mesh is immediately after AT while the right one is at punch home.

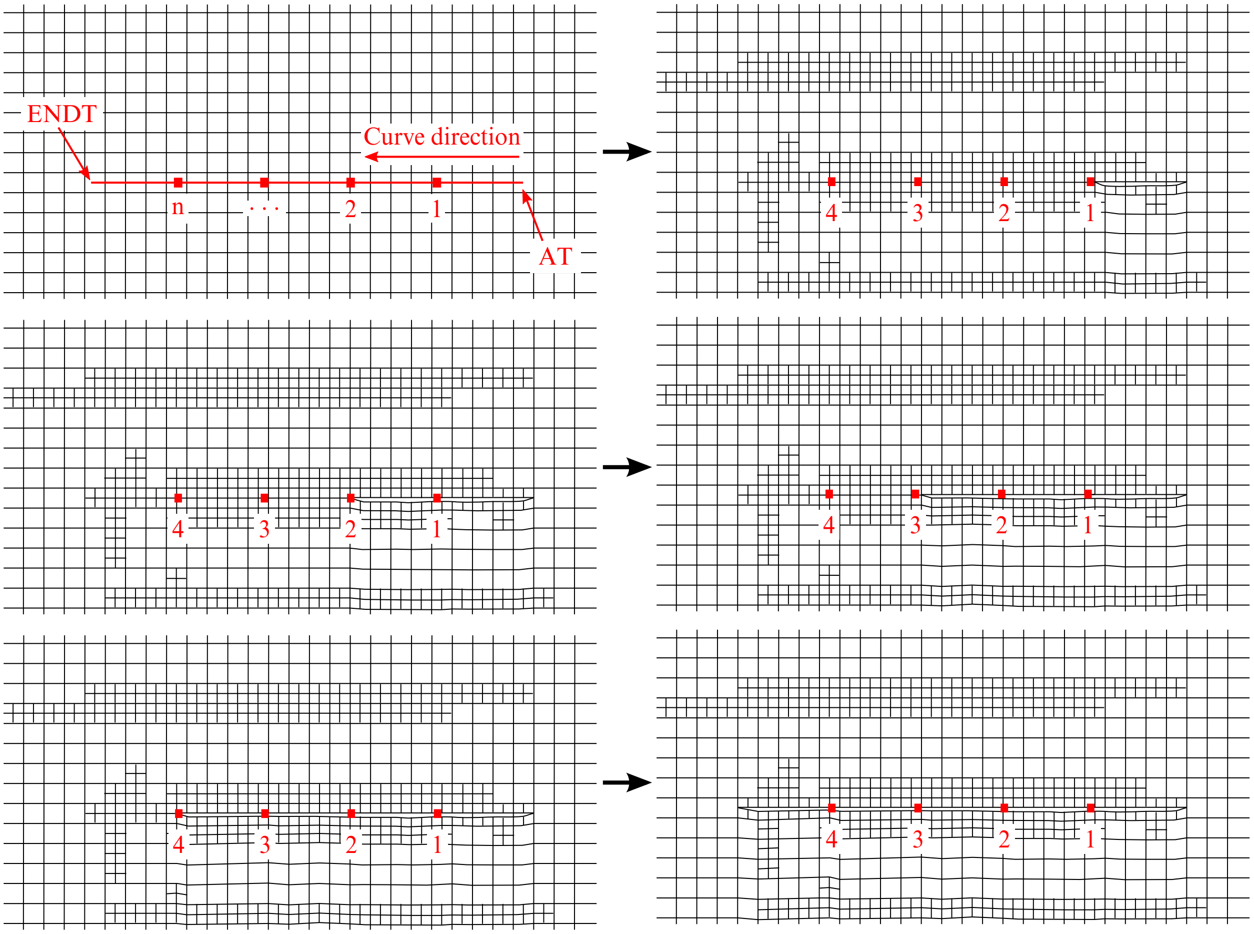


Figure 17-10. Progressive lancing - defining *AT*, *ENDT*, *NTIMES*, curve direction; mesh separation progression during progressive lancing.

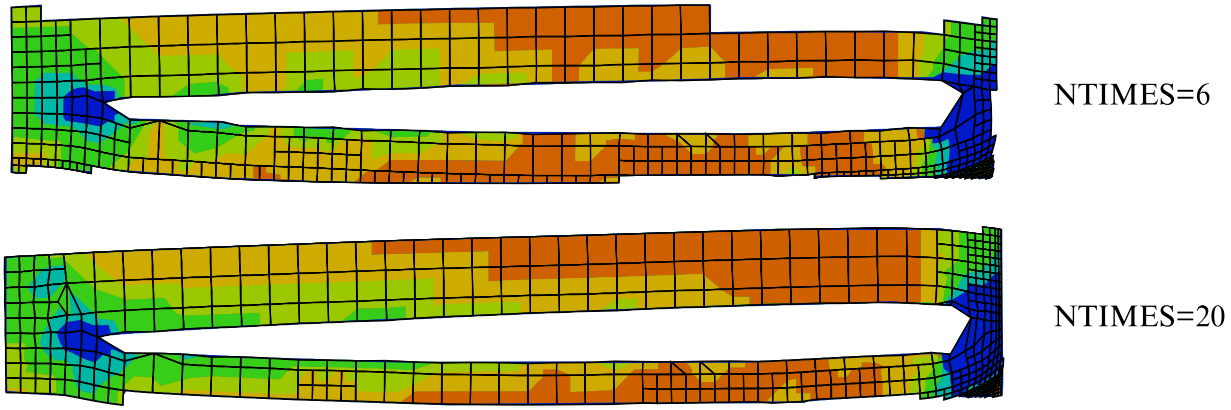


Figure 17-11. More NTIMES gives smoother lancing boundary and less stress concentration.

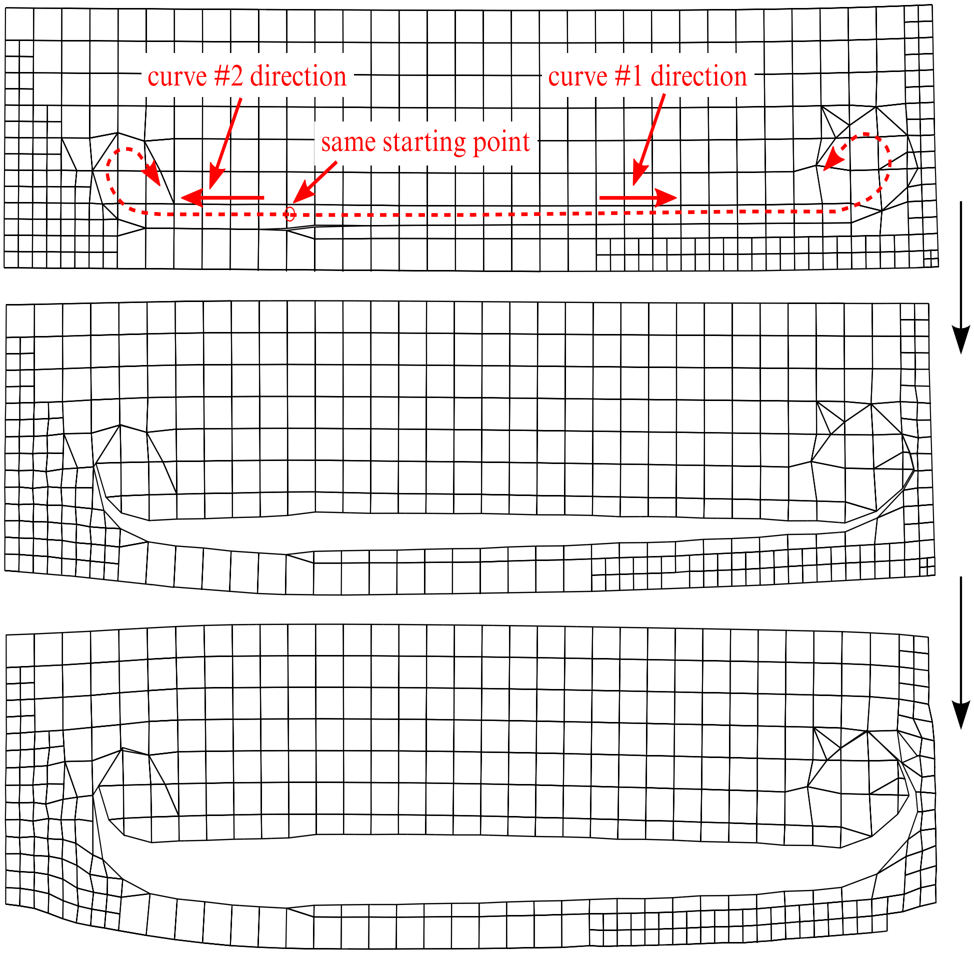
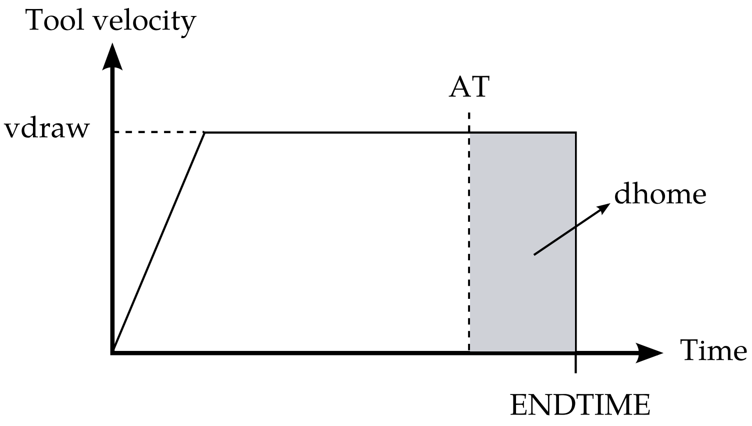


Figure 17-12. Progressive lancing – multiple lancing starting from the same coordinates.



```
*PARAMETER_EXPRESSION
R dhome      12.5
R at         ENDTIME-dhome/vdraw
*ELEMENT_LANCING
$ IDPT       IDCV      IREFINE       SMIN       AT
     9        112                              &at
```

Figure 17-13. An example of defining lancing activation time AT using tool’s distance to home.

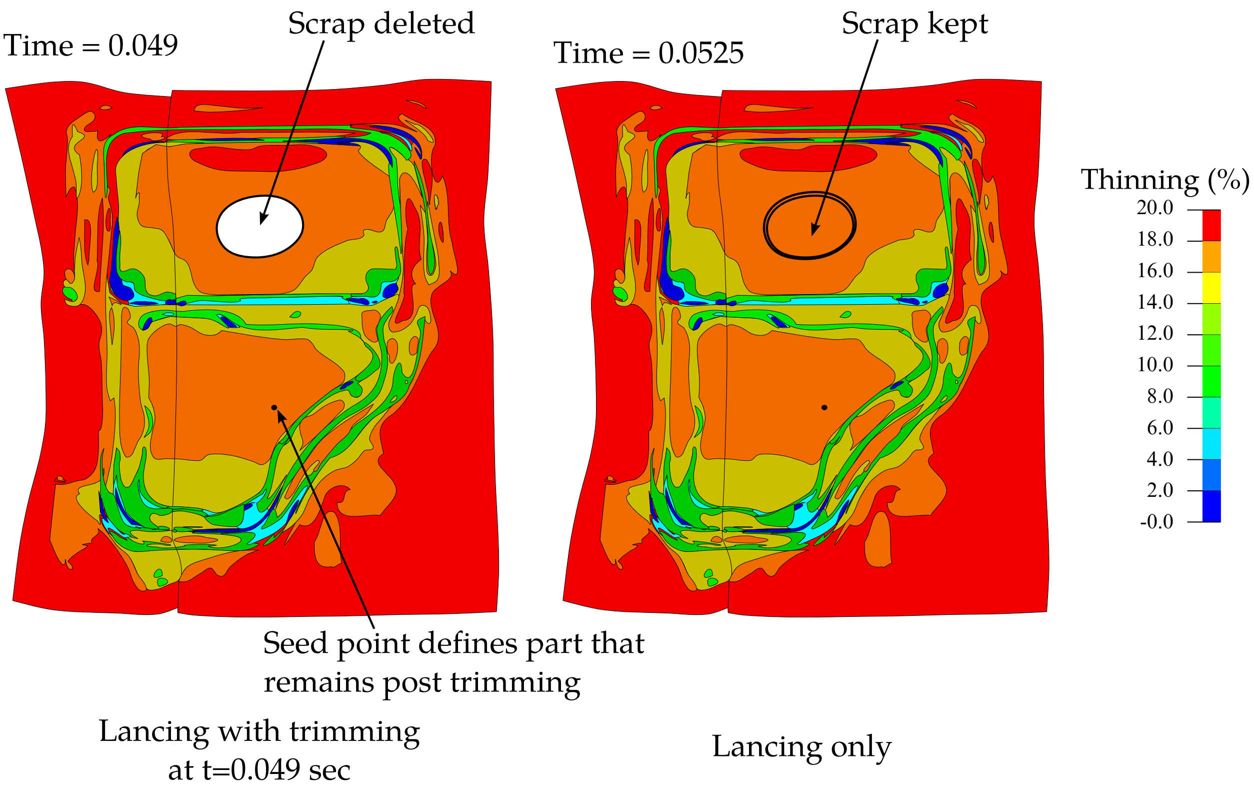
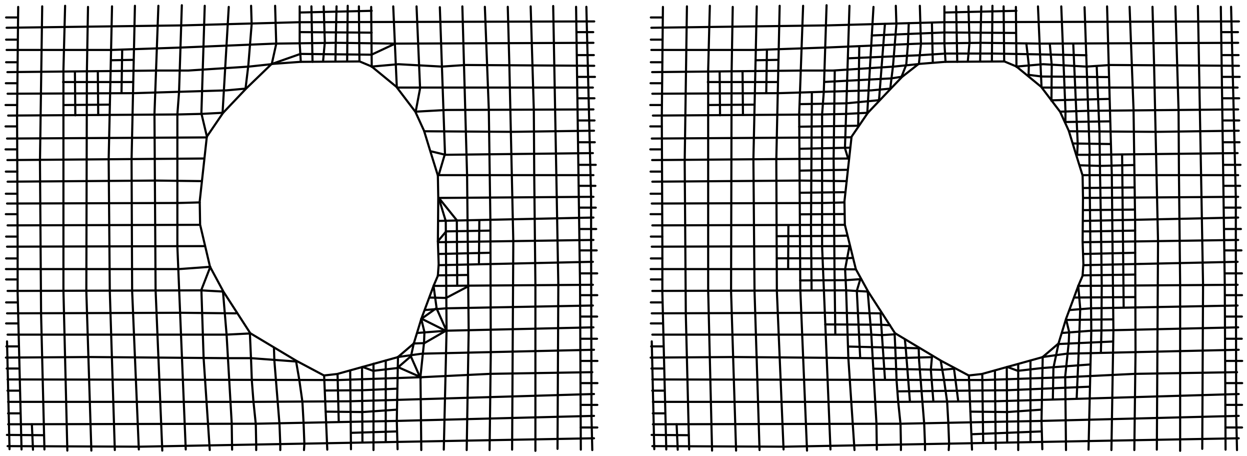


Figure 17-14. Lancing with trimming



Lanced mesh prior to Revision 107708

Improved lanced boundary mesh with IREFINE=1 after Revision 107708

Figure 17-15. Set IREFINE = 1 (>Revision 107708) for improved lanced boundary.

ELEMENT_MASS_*{OPTION}

Available options include:

<BLANK>

NODE_SET

Purpose: Define a lumped mass element assigned to a nodal point or equally distributed to the nodes of a node set.

(Note: NODE_SET option is available starting with the R3 release of Version 971.)

Card	1	2	3	4	5	6	7	8	9	10
Variable	EID	ID	MASS		PID					
Type	I	I	F		I					
Default	none	none	0.		none					

VARIABLE **DESCRIPTION**

EID Element ID. A unique number is recommended. The nodes in a node set share the same element ID.

ID Node ID or node set ID if the NODE_SET option is active. This is the node or node set to which the mass is assigned.

MASS Mass value. When the NODE_SET option is active, the mass is equally distributed to all nodes in a node set.

PID Part ID. This input is optional.

Remarks:

1. Kinetic energy of lumped mass elements is output as kinetic energy of part 0 in matsum (*DATABASE_MATSUM) if IERODE is set to 1 on *CONTROL_OUTPUT.

ELEMENT_MASS_MATRIX_*{OPTION}

Available options include:

<BLANK>

NODE_SET

Purpose: Define a 6 × 6 symmetric nodal mass matrix assigned to a nodal point or each node within a node set. A node may not be included in more than one *ELEMENT_-MASS_MATRIX(_NODE_SET) command.

Card 1	1	2	3	4	5	6	7	8
Variable	EID	ID	CID					
Type	I	I	I					
Default	none	none	0					

Card 2	1	2	3	4	5	6	7	8
Variable	M11	M21	M22	M31	M32	M33	M41	
Type	F	F	F	F	F	F	F	
Default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

Card 3	1	2	3	4	5	6	7	8
Variable	M42	M43	M44	M51	M52	M53	M54	
Type	F	F	F	F	F	F	F	
Default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

Card 4	1	2	3	4	5	6	7	8
Variable	M55	M61	M62	M63	M64	M65	M66	
Type	F	F	F	F	F	F	F	
Default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

VARIABLE	DESCRIPTION
EID	Element ID. A unique number is recommended. The nodes in a node set share the same element ID.
ID	Node ID or node set ID if the NODE_SET option is active. This is the node or node set to which the mass is assigned.
CID	Local coordinate ID which defines the orientation of the mass matrix
M*ij*	The *ij*th term of the symmetric mass matrix. The lower triangular part of the matrix is defined.

ELEMENT_MASS_PART_*{OPTION}

Available options include:

<BLANK>

SET

Purpose: Define additional non-structural mass to be distributed by an area (shell) / volume (solid) or mass weighted distribution to all nodes of a given part, or part set, ID. As an option, the total mass can be defined and the additional non-structural mass is computed. This option applies to all part ID's defined by shell and solid elements.

Card	1	2	3	4	5	6	7	8	9	10
Variable	ID	ADDMASS		FINMASS		LCID		MWD		
Type	I	F		F		I		I		
Default	none	0.		0.		0		0		

VARIABLE	DESCRIPTION
ID	Part or part set ID if the SET option is active. A unique number must be used.
ADDMASS	Added translational mass to be distributed to the nodes of the part ID or part set ID. Set to zero if FINMASS is nonzero. Since the additional mass is not included in the time step calculation of the elements in the PID or SID, ADDMASS must be greater than zero if FINMASS is zero.
FINMASS	Final translational mass of the part ID or part set ID. The total mass of the PID or SID is computed and subtracted from the final mass of the part or part set to obtain the added translational mass, which must exceed zero. Set FINMASS to zero if ADDMASS is nonzero. FINMASS is available in the R3 release of version 971.
LCID	Optional load curve ID to scale the added mass at time = 0. This curve defines the scale factor as a function versus time. The curve must start at unity at t = 0. This option applies to deformable bodies only.

VARIABLE	DESCRIPTION
MWD	Optional flag for mass-weighted distribution, valid for SET option only: EQ.0: non-structural mass is distributed by area(shell)/volume(solid) weighted distribution, EQ.1: non-structural mass is distributed by mass weighted, area*density*thickness(shell)/volume*density(solid), distribution. Mixed uses with MWD for the same part should be avoided.

*ELEMENT_PLOTEL

Purpose: Define a null beam element for visualization.

Card	1	2	3	4	5	6	7	8	9	10
Variable	EID	N1	N2							
Type	I	I	I							
Default	none	none	none							
Remarks	1									

VARIABLE	DESCRIPTION
EID	Element ID. A unique number must be used.
N1	Nodal point (end) 1.
N2	Nodal point (end) 2.

Remarks:

1. Part ID, 10000000, is assigned to PLOTEL elements.
2. PLOTEL element ID's must be unique with respect to other beam elements.

*ELEMENT_SEATBELT

Purpose: Define a seat belt element.

Card	1	2	3	4	5	6	7	8	9	10
Variable	EID	PID	N1	N2	SBRID	SLEN		N3	N4	
Type	I	I	I	I	I	F		I	I	
Default	none	none	none	none	none	0.0		0	0	
Remarks					1	2		3		

VARIABLE	DESCRIPTION
EID	Element ID. A unique number is required. Since null beams are created for visualization, this element ID should not be identical to element ID's defined for ELEMENT_BEAM and ELEMENT_DISCRETE.
PID	Part ID
N1	Node 1 ID
N2	Node 2 ID
SBRID	Retractor ID, see *ELEMENT_SEATBELT_RETRACTOR.
SLEN	Initial slack length
N3	Optional node 3 ID. When N3 > 0 and N4 > 0, the elements becomes a shell seat belt element. The thickness of the shell seatbelt is defined in *SECTION_SHELL, not in *SECTION_SEATBELT. The shell-type seatbelt must be of a rectangular shape as shown in Figure 17-16 and contained in a logically regular mesh.
N4	Node 4 ID, which is required if and only if N3 is defined.

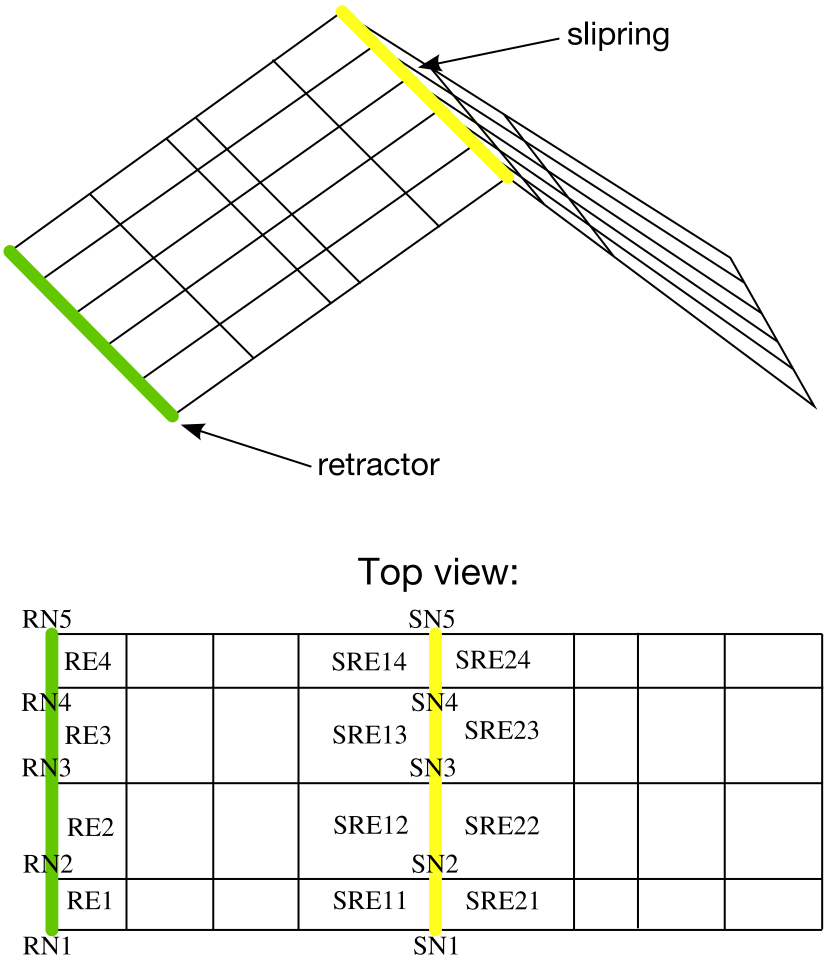


Figure 17-16. Definition of seatbelt shell elements. The ordering of the nodes and elements are important for seatbelt shells. See the input descriptions for SECTION_SHELL, ELEMENT_SEATBELT_RETRACTOR and ELEMENT_SEATBELT_SLIPRING.

Remarks:

1. The retractor ID should be defined only if the element is initially **inside** a retractor, see *ELEMENT_SEATBELT_RETRACTOR.

2. Belt elements are single degree of freedom elements connecting two nodes. When the strain in an element is positive (i.e. the current length is greater then the unstretched length), a tension force is calculated from the material characteristics and is applied along the current axis of the element to oppose further stretching. The unstretched length of the belt is taken as the initial distance between the two nodes defining the position of the element plus the initial slack length.

3. Seatbelt shell elements are a new feature in version 971 and must be used with caution. The seatbelt shells distribute the loading on the surface of the dummy more realistically than the two node belt elements. For the seatbelt shells to work with sliprings and retractors it is necessary to use a logically regular mesh of quadrilateral elements. *A seatbelt defined by a part ID must not be disjoint.*

4. 1D and 2D seatbelt elements may not share the same material ID.

*ELEMENT_SEATBELT_ACCELEROMETER

Purpose: This keyword command defines an accelerometer. Contrary to the keyword name, an accelerometer need not be associated with a seat belt. The accelerometer is fixed to a rigid body containing the three nodes defined below. An accelerometer will exhibit considerably less numerical noise than a deformable node, thereby reporting more meaningful data to the user. Whenever computed accelerations are compared to experimental data, or whenever computed accelerations are compared between different runs, this feature is essential.

Card	1	2	3	4	5	6	7	8
Variable	SBACID	NID1	NID2	NID3	IGRAV	INTOPT	MASS	
Type	I	I	I	I	I	I	F	
Default	0	0	0	0	0	0	0.	

VARIABLE	DESCRIPTION
SBACID	Accelerometer ID. A unique number must be used.
NID1	Node 1 ID
NID2	Node 2 ID
NID3	Node 3 ID
IGRAV	Gravitational accelerations due to body force loads. EQ.-6: z and x components removed from acceleration output EQ.-5: y and z components removed from acceleration output EQ.-4: x and y components removed from acceleration output EQ.-3: z component removed from acceleration output EQ.-2: y component removed from acceleration output EQ.-1: x component removed from acceleration output EQ.0: all components included in acceleration output EQ.1: all components removed from acceleration output GT.1: IGRAV is a curve ID defining the gravitation-flag versus time. The ordinate values, representing the

VARIABLE **DESCRIPTION**

gravitation-flag, can be -6, -5, -4, -3, -2, -1, 0 or 1, as described above. For example, a curve with 4 data points of (0.,1), (10.,1), (10.000001,0), (200.,0) sets gravitation flag to be 1 when time ≤ 10, and 0 when time > 10. In other words, all components of gravitational accelerations are removed when time $\le 10.$, and then included when time > 10.0.

INTOPT

Integration option. If the accelerometer undergoes rigid body translation without rotation this option has no effect; however, if rotation occurs, INTOPT affects how translational velocities (and displacements) are calculated. Note that the acceleration values written to the nodout file are unaffected by INTOPT.

EQ.0: velocities are integrated from the global accelerations and transformed into the local system of the accelerometer.

EQ.1: velocities are integrated directly from the local accelerations of the accelerometer.

MASS

Optional added mass for accelerometer. This mass is equally distributed to nodal points NID1, NID2, and NID3. This option avoids the need to use the *ELEMENT_MASS keyword input if additional mass is required.

Remarks:

The presence of the accelerometer means that the accelerations and velocities of node 1 will be output to **all** output files in local instead of global coordinates.

The local coordinate system is defined by the three nodes as follows:

1. local $\mathbf{x}$ from node 1 to node 2,

2. local $\mathbf{z}$ perpendicular to the plane containing nodes, 1, 2, and 3 ($\mathbf{z} = \mathbf{x} \times \mathbf{a}$), where $\mathbf{a}$ is from node 1 to node 3),

3. local $\mathbf{y} = \mathbf{z} \times \mathbf{x}$.

The three nodes should all be part of the same rigid body. The local axis then rotates with the body.

*ELEMENT_SEATBELT_PRETENSIONER

Purpose: Define seat belt pretensioner. A combination with sensors and retractors is also possible.

Card 1	1	2	3	4	5	6	7	8
Variable	SBPRID	SBPRTY	SBSID1	SBSID2	SBSID3	SBSID4		
Type	I	I	I	I	I	I		
Default	0	0	0	0	0	0		
Remarks			1					

Card 2	1	2	3	4	5	6	7	8
Variable	SBRID	TIME	PTLCID	LMTFRC				
Type	I	F	I	F				
Default	0	0.0	0	0				
Remarks								

VARIABLE	DESCRIPTION
SBPRID	Pretensioner ID. A unique number has to be used.

VARIABLE	DESCRIPTION
SBPRTY	Pretensioner type (see Remark 2 below): EQ.1: pyrotechnic retractor with force limits, EQ.2: pre-loaded spring becomes active, EQ.3: lock spring removed, EQ.4: force versus time retractor. EQ.5: pyrotechnic retractor (old type in version 950) but with optional force limiter, LMTFRC. EQ.6: combination of types 4 and 5 as described in the notes below. EQ.7: independent pretensioner/retractor. EQ.8: energy versus time retractor pretensioner with optional force limiter, LMTFRC. EQ.9: energy versus time buckle or anchor pretensioner.
SBSID1	Sensor 1, see *ELEMENT_SEATBELT_SENSOR.
SBSID2	Sensor 2, see *ELEMENT_SEATBELT_SENSOR.
SBSID3	Sensor 3, see *ELEMENT_SEATBELT_SENSOR.
SBSID4	Sensor 4, see *ELEMENT_SEATBELT_SENSOR.
SBRID	Retractor number (SBPRTY = 1, 4, 5, 6, 7 or 8) or spring element number (SBPRTY = 2, 3 or 9).
TIME	Time between sensor triggering and pretensioner acting.
PTLCID	Load curve for pretensioner (Time after activation, Pull-in) (SBPRTY = 1, 4, 5, 6, 7, 8 or 9).
LMTFRC	Optional limiting force for retractor type 5 or 8. If zero, this option is ignored.

Activation:

To activate the pretensioner, the following sequence of events must occur:

1. Any one of up to four sensors must be triggered.

2. Then a user-defined time delay occurs.

3. Then the pretensioner acts.

At least one sensor should be defined.

Pretensioners allow modeling of seven types of active devices which tighten the belt during the initial stages of a crash. Types 1 and 5 implement a pyrotechnic device which spins the spool of a retractor, causing the belt to be reeled in. The user defines a pull-in versus time curve which applies once the pretensioner activates. Types 2 and 3 implement preloaded springs or torsion bars which move the buckle when released.

Types 2 and 3:

The pretensioner is associated with any type of spring element including rotational. Note that the preloaded spring, locking spring, and any restraints on the motion of the associated nodes are defined in the normal way; the action of the pretensioner is merely to cancel the force in one spring until (or after) it fires. With the second type, the force in the spring element is canceled out until the pretensioner is activated. In this case the spring in question is normally a stiff, linear spring which acts as a locking mechanism, preventing motion of the seat belt buckle relative to the vehicle. A preloaded spring is defined in parallel with the locking spring. This type avoids the problem of the buckle being free to 'drift' before the pretensioner is activated. Types 4, 6, and 7, force types, are described below.

Type 1:

As of version 950 the type 1 (now type 5) pretensioner requires that the user provide a load curve tabulating the pull-in of the pretensioner as a function of time. This pretensioner type interacts with the retractor, forcing it to pull in by the amount of belt indicated. It works well, and does exactly what it says it will do, but it can be difficult to use. The reason for this is that it has no regard for the forces being exerted on the belt. If a pull-in of 20mm is specified at a particular time, then 20mm of belt will be pulled in, even if this results in unrealistic forces in the seatbelt. Furthermore, there was no explicit way to turn this pretensioner off. Once defined, it overrode the retractor completely, and the amount of belt passing into or out of the retractor depended solely on the load curve specified.

For the 970 release of LS-DYNA, the behavior of the type 1 pretensioner was changed due to user feedback regarding these shortcomings. Each retractor has a loading (and optional unloading) curve that describes the force on the belt element as a function of the amount of belt that has been pulled out of the retractor since the retractor locked. The new type 1 pretensioner acts as a shift of this retractor load curve. An example will make this clear. Suppose at a particular time that 5mm of belt material has left the retractor. The retractor will respond with a force corresponding to 5mm pull-out on it's loading curve. But suppose this retractor has a type 1 pretensioner defined, and at this

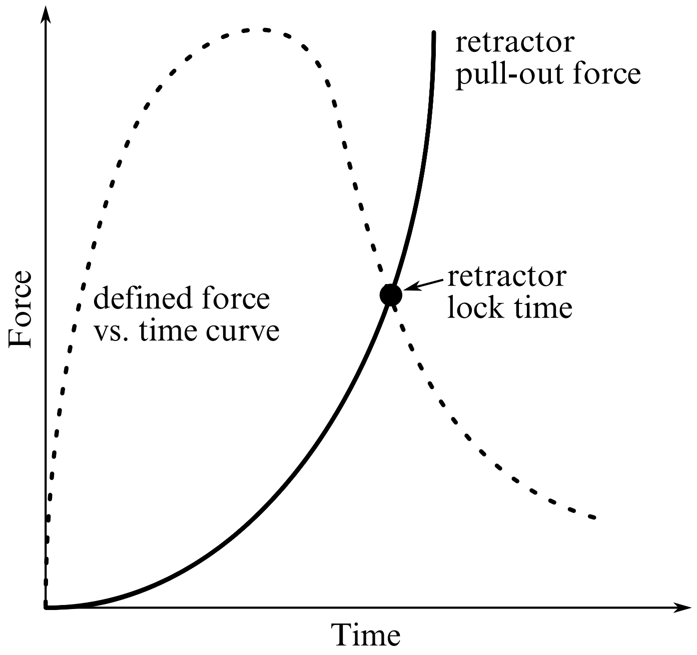


Figure 17-17 Force versus time pretensioner. At the intersection, the retractor locks.

instant of time the pretensioner specifies a pull-in of 20mm. The retractor will then respond with a force that corresponds to (5mm + 20mm) on it's loading curve. This results in a much larger force. The effect can be that belt material will be pulled in, but unlike in the 950 version, there is no guarantee. The benefit of this implementation is that the force vs. pull-in load curve for the retractor is followed and no unrealistic forces are generated. Still, it may be difficult to produce realistic models using this option, so two new types of pretensioners have been added. These are available in 970 versions 1300 and later.

Type 4:

The type 4 pretensioner takes a force vs. time curve, See Figure 17-17. Each time step, the retractor computes the desired force without regard to the pretensioner. If the resulting force is less than that specified by the pretensioner load curve, then the pretensioner value is used instead. As time goes on, the pretensioner load curve should drop below the forces generated by the retractor, and the pretensioner is then essentially inactive. This provides for good control of the actual forces, so no unrealistic values are generated. The actual direction and amount of belt movement is unspecified, and will depend on the other forces being exerted on the belt. This is suitable when the force the pretensioner exerts over time is known.

Type 5:

The type 5 pretensioner is essentially the same as the old type 1 pretensioner, but with the addition of a force limiting value. The pull-in is given as a function of time, and the belt is drawn into the retractor exactly as desired. However, if at any point the forces generated in the belt exceed the pretensioner force limit, then the pretensioner is deactivated and the retractor takes over. In order to prevent a large discontinuity in the force at this point, the loading curve for the retractor is shifted (in the abscissa) by the amount required to put the current (pull-out, force) on the load curve. For example, suppose the current force is 1000, and the current pull-out is -10 (10mm of belt has been pulled IN by the pretensioner). If the retractor would normally generate a force of 1000 after 25mm of belt had been pulled OUT, then the load curve is shifted to the left by 35, and remains that way for the duration of the calculation. So that at the current pull-in of 10, it will generate the force normally associated with a pull out of 25. If the belt reaches a pull out of 5, the force will be as if it were pulled out 40 (5 + the shift of 35), and so on. This option is included for those who liked the general behavior of the old type 1 pretensioner, but has the added feature of the force limit to prevent unrealistic behavior.

Type 6:

The type 6 pretensioner is a variation of the type 4 pretensioner, with features of the type 5 pretensioner. A force vs. time curve is input and the pretensioner force is computed each cycle. The retractor linked to this pretensioner should specify a positive value for PULL, which is the distance the belt pulls out before it locks. As the pretensioner pulls the belt into the retractor, the amount of pull-in is tracked. As the pretensioner force decreases and drops below the belt tension, belt will begin to move back out of the retractor. Once PULL amount of belt has moved out of the retractor (relative to the maximum pull in encountered), the retractor will lock. At this time, the pretensioner is disabled, and the retractor force curve is shifted to match the current belt tension. This shifting is done just like the type 5 pretensioner. It is important that a positive value of PULL be specified to prevent premature retractor locking which could occur due to small outward belt movements generated by noise in the simulation.

Type 7:

The type 7 pretensioner is a simple combination of retractor and pretensioner. It is similar to the type 6 except for the following changes: when the retractor locks, the pretensioner is NOT disabled – it continues to exert force according to the force vs. time curve until the end of the simulation. (The force vs. time curve should probably drop to 0 at some time.) Furthermore, the retractor load curve is not shifted – the retractor begins to exert force according to the force vs. pull-out curve. These two forces are added together and applied to the belt. Thus, the pretensioner and retractor are essentially independent.

Type 8:

The type 8 pretensioner is a variation of type 5 pretensioner. The pretension energy, instead of pull-in for type 5, is given as a function of time. This enables users to use a single pretensioner curve, PTLCID, for various sizes of dummies. The energy could be yielded from the baseline test or simulation by$E(t) = \int_0^t fdp$, where f is the force of the mouth element of the retractor and dp is the incremental pull-in.

Type 9:

The type 9 pretensioner is designed for a pretension-energy based buckle or anchor pretensioner. The pretensioner is modeled as a spring element, SBRID. One end of the spring element is attached to the vehicle. For a buckle pretensioner, the other end of SBRID is the slip ring node, SBRNID, of a slip ring representing the buckle. For an anchor pretensioner, SBRID shares the other end with a belt element, see Figure

*ELEMENT_SEATBELT_RETRACTOR

Purpose: Define seat belt retractor. See remarks below for seatbelt shell elements.

Card 1	1	2	3	4	5	6	7	8
Variable	SBRID	SBRNID	SBID	SID1	SID2	SID3	SID4	
Type	I	I	I	I	I	I	I	
Default	0	0	0	0	0	0	0	
Remarks		1,2	2	3				

Card 2	1	2	3	4	5	6	7	8
Variable	TDEL	PULL	LLCID	ULCID	LFED			
Type	F	F	I	I	F			
Default	0.0	0.0	0	0	0.0			
Remarks			4	5				

VARIABLE	DESCRIPTION
SBRID	Retractor ID. A unique number has to be used.
SBRNID	Retractor node ID
SBID	Seat belt element ID
SID1	Sensor ID 1
SID2	Sensor ID 2
SID3	Sensor ID 3
SID4	Sensor ID 4

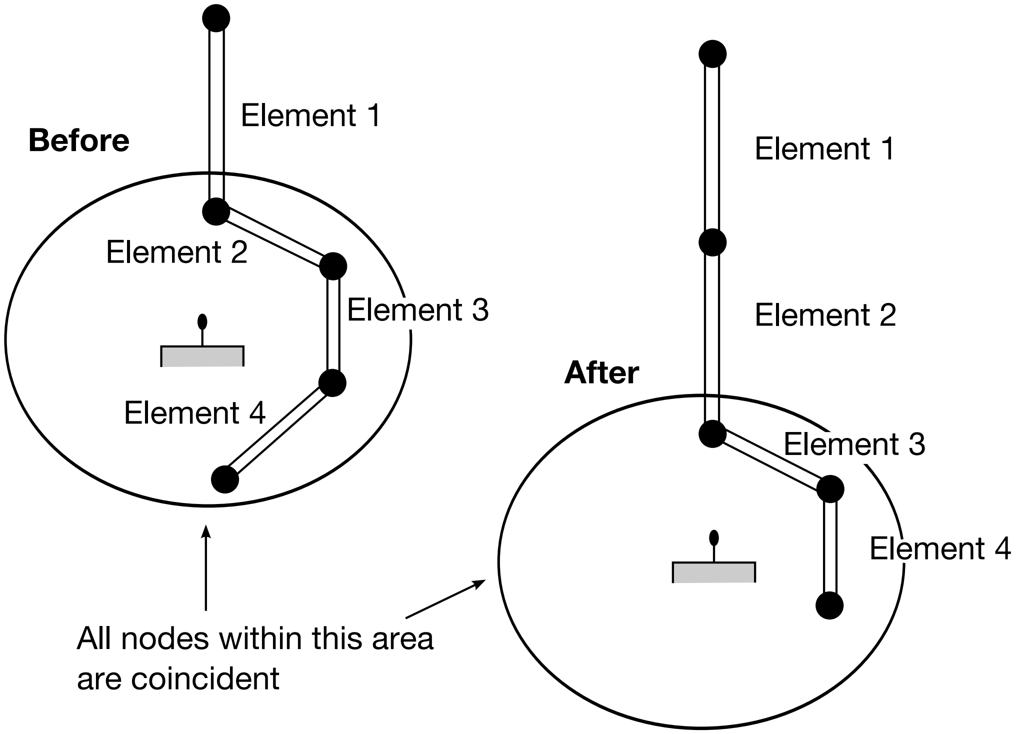


Figure 17-18. Elements in a retractor.

VARIABLE	DESCRIPTION
TDEL	Time delay after sensor triggers.
PULL	Amount of pull-out between time delay ending and retractor locking, a length value.
LLCID	Load curve for loading (Pull-out, Force), see Figure 17-18.
ULCID	Load curve for unloading (Pull-out, Force), see Figure 17-18.
LFED	Fed length, see explanation below.

Remarks:

1. The retractor node should not be on any belt elements. The element defined should have one node coincident with the retractor node but should not be inside the retractor.

2. When SBRNID < 0, this retractor is for shell-type seatbelt, -SBRNID is the *SET_NODE containing RN1, RN2, …RN5. SBID is then *SET_SHELL_LIST. Note that the numbering of –SBRNID, SBID has to be consistent in the direction of numbering. For example, if *SET_NODE for SBRNID has nodes of (RN1,

RN2, RN3, RN4, RN5) then *SET_SHELL_LIST for SBID should have elem. of (RE1, RE2, RE3, RE4). See Figure 17-16.

3. At least one sensor should be defined.

4. The first point of the load curve should be (0, T_{min}). T_{min} is the minimum tension. All subsequent tension values should be greater than T_{min}.

5. The unloading curve should start at zero tension and increase monotonically (i.e., no segments of negative or zero slope).

Retractors allow belt material to be paid out into a belt element. Retractors operate in one of two regimes: unlocked when the belt material is paid out, or reeled in under constant tension and locked when a user defined force-pullout relationship applies.

The retractor is initially unlocked, and the following sequence of events must occur for it to become locked:

a) Any one of up to four sensors must be triggered. (The sensors are described below.)

b) Then a user-defined time delay occurs.

c) Then a user-defined length of belt must be paid out (optional).

d) Then the retractor locks and once locked, it remains locked.

In the unlocked regime, the retractor attempts to apply a constant tension to the belt. This feature allows an initial tightening of the belt and takes up any slack whenever it occurs. The tension value is taken from the first point on the force-pullout load curve. The maximum rate of pull out or pull in is given by 0.01 × fed length per time step. Because of this, the constant tension value is not always achieved.

In the locked regime, a user-defined curve describes the relationship between the force in the attached element and the amount of belt material paid out. If the tension in the belt subsequently relaxes, a different user-defined curve applies for unloading. The unloading curve is followed until the minimum tension is reached.

The curves are defined in terms of initial length of belt. For example, if a belt is marked at 10mm intervals and then wound onto a retractor, and the force required to make each mark emerge from the (locked) retractor is recorded, the curves used for input would be as follows:

0 Minimum tension (should be > zero)

10mm	Force to emergence of first mark
20mm	Force to emergence of second mark
⋮	

Pyrotechnic pretensions may be defined which cause the retractor to pull in the belt at a predetermined rate. This overrides the retractor force-pullout relationship from the moment when the pretensioner activates.

If desired, belt elements may be defined which are initially inside the retractor. These will emerge as belt material is paid out, and may return into the retractor if sufficient material is reeled in during unloading.

Elements e2, e3 and e4 are initially inside the retractor, which is paying out material into element e1. When the retractor has fed L_{crit} into e1, where

$$L_{crit} = \text{fed length} - 1.1 \times \text{minimum length}$$

(minimum length defined on belt material input)

(fed length defined on retractor input)

Element e2 emerges with an unstretched length of 1.1 x minimum length; the unstretched length of element e1 is reduced by the same amount. The force and strain in e1 are unchanged; in e2, they are set equal to those in e1. The retractor now pays out material into e2.

If no elements are inside the retractor, e2 can continue to extend as more material is fed into it.

As the retractor pulls in the belt (for example, during initial tightening), if the unstretched length of the mouth element becomes less than the minimum length, the element is taken into the retractor.

To define a retractor, the user enters the retractor node, the ‘mouth’ element (into which belt material will be fed), e1 in Figure 17-18, up to 4 sensors which can trigger unlocking, a time delay, a payout delay (optional), load and unload curve numbers, and the fed length. The retractor node is typically part of the vehicle structure; belt elements should not be connected to this node directly, but any other feature can be attached including rigid bodies. The mouth element should have a node coincident with the retractor but should not be inside the retractor. The fed length would typically be set either to a typical element initial length, for the distance between painted marks on a real belt for comparisons with high speed film. The fed length should be at least three times the minimum length.

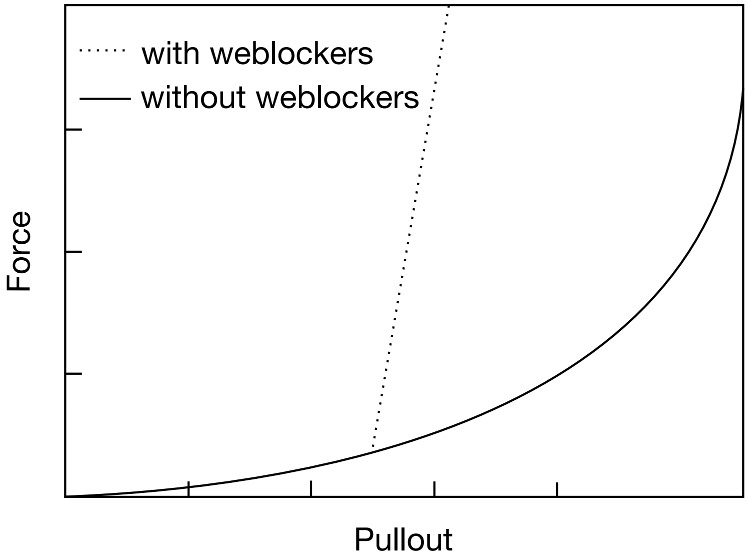


Figure 17-19. Retractor force pull characteristics.

If there are elements initially inside the retractor (e2, e3 and e4 in the Figure) they should not be referred to on the retractor input, but the retractor should be identified on the element input for these elements. Their nodes should all be coincident with the retractor node and should not be restrained or constrained. Initial slack will automatically be set to 1.1 × minimum length for these elements; this overrides any user-defined value.

Weblockers can be included within the retractor representation simply by entering a 'locking up' characteristic in the force pullout curve, see Figure 17-19. The final section can be very steep (but must have a finite slope).

6. In an event when only retractors are used in the model, be aware that the pullout is measured from the point when the retractor is locked. If the belt has been pulled IN since the retractor was locked, then minimum force will be seen in the retractor until the system pays out enough belt to get back to the point when locked

   If the behavior described in the above note undesirable then the type 6 pretensioner model is recommended for the seat belt system. A constant force vs. time load curve with a force equal to minimum tension fwill be defined, with a small PULL value on the retractor. With this set up, the pretensioner will be active until the belt pulls all the way in, but as soon as the belt starts to move back out, the pretensioner will get disabled and the retractor will take over.

*ELEMENT_SEATBELT_SENSOR

Purpose: Define seat belt sensor. Four types are possible, see explanation below.

Card 1	1	2	3	4	5	6	7	8
Variable	SBSID	SBSTYP	SBSFL					
Type	I	I	I					
Default	0	0	0					
Remarks								

Additional card for SBSTYP = 1.

Card 2	1	2	3	4	5	6	7	8
Variable	NID	DOF	ACC	ATIME				
Type	I	I	F	F				
Default	0	0	0.0	0.0				
Remarks	1							

Additional card for SBSTYP = 2.

Card 2	1	2	3	4	5	6	7	8
Variable	SBRID	PULRAT	PULTIM					
Type	I	F	F					
Default	0	0.0	0.0					
Remarks								

Additional card for SBSTYP = 3.

Card 2	1	2	3	4	5	6	7	8
Variable	TIME							
Type	F							
Default	0.0							
Remarks								

Additional card for SBSTYP = 4.

Card 2	1	2	3	4	5	6	7	8
Variable	NID1	NID2	DMX	DMN				
Type	I	I	F	F				
Default	0	0	0.0	0.0				
Remarks			2	2				

VARIABLE **DESCRIPTION**

SBSID Sensor ID. A unique number has to be used.

SBSTYP Sensor type:

EQ.1: acceleration of node,

EQ.2: retractor pull-out rate,

EQ.3: time,

EQ.4: distance between nodes.

SBSFL Sensor flag:

EQ.0: sensor active during dynamic relaxation,

EQ.1: sensor can be triggered during dynamic relaxation.

VARIABLE	DESCRIPTION
NID	Node ID of sensor
DOF	Degree of freedom: EQ.1: x, EQ.2: y, EQ.3: z.
ACC	Activating acceleration
ATIME	Time over which acceleration must be exceeded
SBRID	Retractor ID, see *ELEMENT_SEATBELT_RETRACTOR.
PULRAT	Rate of pull-out (length/time units)
PULTIM	Time over which rate of pull-out must be exceeded
TIME	Time at which sensor triggers
NID1	Node 1 ID
NID2	Node 2 ID
DMX	Maximum distance
DMN	Minimum distance

Remarks:

1. Node should not be on rigid body, velocity boundary condition, or other ‘imposed motion’ feature.

2. Sensor triggers when the distance between the two nodes is $d \geq d_{max}$ or $d \leq d_{min.}$ Sensors are used to trigger locking of retractors and activate pretensioners. Four types of sensors are available which trigger according to the following criteria:

   Type 1 When the magnitude of x-, y-, or z- acceleration of a given node has remained above a given level continuously for a given time, the sensor triggers. This does not work with nodes on rigid bodies.

Type 2 When the rate of belt payout from a given retractor has remained above a given level continuously for a given time, the sensor triggers.

Type 3 The sensor triggers at a given time.

Type 4 The sensor triggers when the distance between two nodes exceeds a given maximum or becomes less than a given minimum. This type of sensor is intended for use with an explicit mass/spring representation of the sensor mechanism.

By default, the sensors are inactive during dynamic relaxation. This allows initial tightening of the belt and positioning of the occupant on the seat without locking the retractor or firing any pretensioners. However, a flag can be set in the sensor input to make the sensors active during the dynamic relaxation phase.

*ELEMENT_SEATBELT_SLIPRING

Purpose: Define seat belt slip ring.

Card	1	2	3	4	5	6	7	8
Variable	SBSRID	SBID1	SBID2	FC	SBRNID	LTIME	FCS	ONID
Type	I	I	I	F	I	F	F	I
Default	0	0	0	0.0	0	10^{20}	0.0	0

Optional Card.

Card 2	1	2	3	4	5	6	7	8
Variable	K	FUNCID	DIRECT	DC		LCNFFD	LCNFFS	
Type	F	I	I	F		I	I	
Default	0.0	0	0	0		0	0	

VARIABLE	DESCRIPTION
SBSRID	Slip ring ID. A unique number has to be used. See remarks below for the treatment of slip rings for shell belt elements.
SBID1	Seat belt element 1 ID
SBID2	Seat belt element 2 ID
FC	Coulomb dynamic friction coefficient. If less than zero, |FC| refers to a curve which defines the dynamic friction coefficient as a function of time.
SBRNID	Slip ring node, NID
LTIME	Slip ring lockup time. After this time no material is moved from one side of the slip ring to the other. This option is not active during dynamic relaxation.

VARIABLE **DESCRIPTION**

FCS

Optional Coulomb static friction coefficient. If less than zero, |FCS| refers to a curve which defines the static friction coefficient as a function of time.

ONID

Optional orientation node ID.

K

Optional coefficient for determining the Coulomb friction coefficient related to angle alpha

FUNCID

Function ID to determine friction coefficient.

DIRECT

Direction of belt movement:

EQ.0: if the belt can move along both directions.

EQ.12: if the belt is only allowed to slip along the direction from SBID1 to SBID2

EQ.21: if the belt is only allowed to slip along the direction from SBID2 to SBID

DC

Optional decay constant to allow a smooth transition between the static and dynamic friction coefficients, i.e.,

$$\mu_c = \mathrm{FC} + (\mathrm{FCS} - \mathrm{FC})e^{-\mathrm{DC}\times|v_{\mathrm{rel}}|}$$

LCNFFD

Optional curve for normal-force-dependent Coulomb dynamic friction coefficient. When defined, the dynamic friction coefficient becomes $\mathrm{FC} + f_{\mathrm{LCNFFD}}(F_n)$, where $f_{\mathrm{LCNFFD}}(F_n)$ is the function value of LCNFFD at contact force F_n.

LCNFFS

Optional curve for normal-force-dependent Coulomb static friction coefficient. When defined, the static friction coefficient becomes $\mathrm{FCS} + f_{\mathrm{LCNFFS}}(F_n)$, where $f_{\mathrm{LCNFFS}}(F_n)$ is the function value of LCNFFS at contact force F_n.

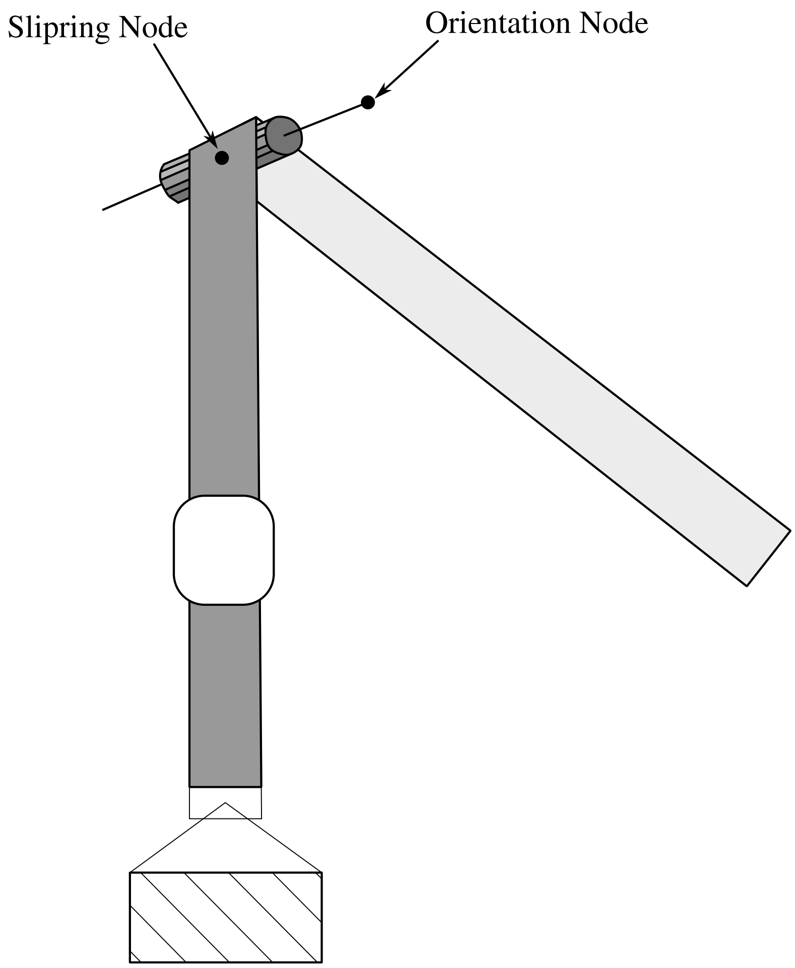


Figure 17-20. Orientation node.

Remarks:

When SBRNID < 0, this slipring is for shell-type seatbelt, -SBRNID is the *SET_NODE containing SN1, SN2, …SN5. SBID1 and SBID2 are then *SET_SHELL_LIST. Note that the numbering of -SBRNID, SBID1 and SBID2 has to be consistent in the direction of numbering. For example if, *SET_NODE for SBRNID has nodes of (SN1, SN2, SN3, SN4, SN5) then *SET_SHELL_LIST for SBID1 should have elem. of (SRE11, SRE12, SRE13, SRE14) and *SET_SHELL_LIST for SBID2 should have elem. of (SRE21, SRE22, SRE23, SRE24). See Figure 17-20.

Elements 1 and 2 should share a node which is coincident with the slip ring node. Elements 1 and 2 should not be referenced in any other slipring definition. The slip ring node should not be on any belt elements.

Sliprings allow continuous sliding of a belt through a sharp change of angle. Two elements (1 & 2 in Figure 17-21) meet at the slipring. Node B in the belt material

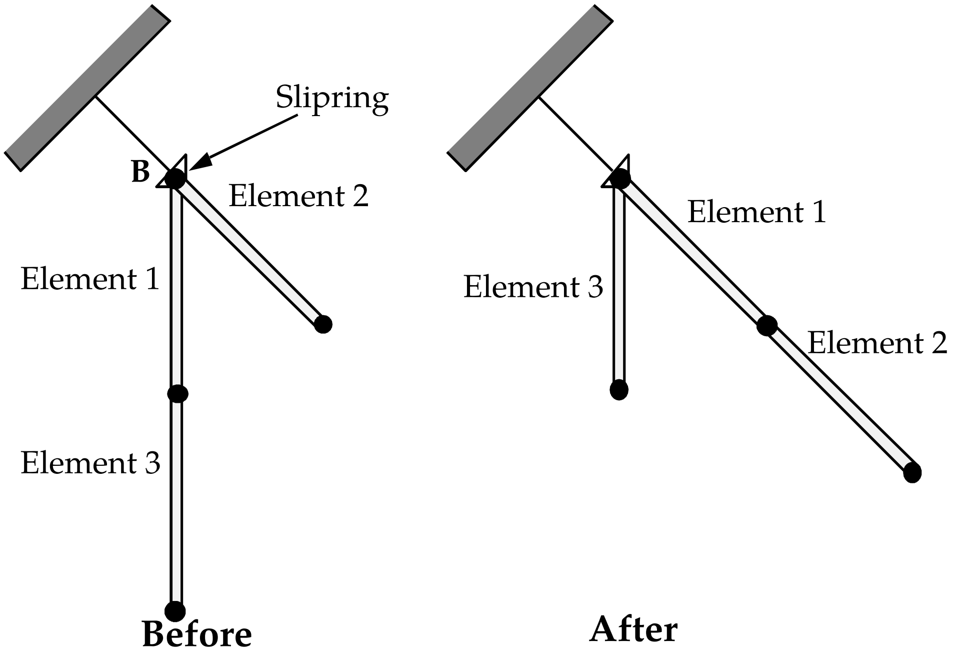


Figure 17-21**.** Elements passing through slipring.

remains attached to the slipring node, but belt material (in the form of unstretched length) is passed from element 1 to element 2 to achieve slip. The amount of slip at each time step is calculated from the ratio of forces in elements 1 and 2. The ratio of forces is determined by the relative angle between elements 1 and 2 and the coefficient of friction, FC. The tension in the belts are taken as T_1 and T_2, where T_2 is on the high tension side and T_1 is the force on the low tension side. Thus, if T_2 is sufficiently close to T_1, no slip occurs; otherwise, slip is just sufficient to reduce the ratio T_2/T_1 to $e^{\mathrm{FC}\times\theta}$, where θ is the wrap angle, see Figures 17-20 and 17-22 No slip occurs if both elements are slack. The out-of-balance force at node B is reacted on the slip ring node; the motion of node B follows that of slip ring node.

If, due to slip through the slip ring, the unstretched length of an element becomes less than the minimum length (as entered on the belt material card), the belt is remeshed locally: the short element passes through the slip ring and reappears on the other side (see Figure 17-21). The new unstretched length of element 1 is 1.1 × minimum length. Force and strain in elements 2 and 3 are unchanged; force and strain in element 1 are now equal to those in element 2. Subsequent slip will pass material from element 3 to element 1. This process can continue with several elements passing in turn through the slip ring.

To define a slip ring, the user identifies the two belt elements which meet at the slip ring, the friction coefficient, and the slip ring node. The two elements must have a common node coincident with the slip ring node. No attempt should be made to restrain or constrain the common node for its motion will automatically be constrained to follow the slip ring node. Typically, the slip ring node is part of the vehicle body structure and, therefore, belt elements should not be connected to this node directly, but any other feature can be attached, including rigid bodies.

If K is undefined, the limiting force ratio is taken as $e^{\mathrm{FC}\times\theta}$. If K is defined, the maximum force ratio is computed as

$$e^{\mathrm{FC}\times\theta(1+\mathrm{K}\times\alpha^2)}$$

where alpha is the angle shown in Figure 17-23. The function is defined using the *DEFINE_FUNCTION keyword input. This function is a function of three variables, and the ratio is given by evaluating

$$\frac{T_2}{T_1} = \mathrm{FUNC}(\mathrm{FCT}, \theta, \alpha)$$

where FCT is the instantaneous friction coefficient at time t, i.e. it has the value of FC if the belt has moved in the last time-step and the value of FCS if the belt has been stationary. For example, the default behavior can be obtained using the function definition (assuming FCT has a value of 0.025 and the function ID is unity)

```
*DEFINE_FUNCTION
    1,
f(fct,theta,alpha) = exp(0.025*theta)
Behavior like default option can be obtained with (K=0.1):
*DEFINE_FUNCTION
  1,
f(fct,theta,alpha) = exp(0.025*theta*(1.+0.1*alpha*alpha))
```

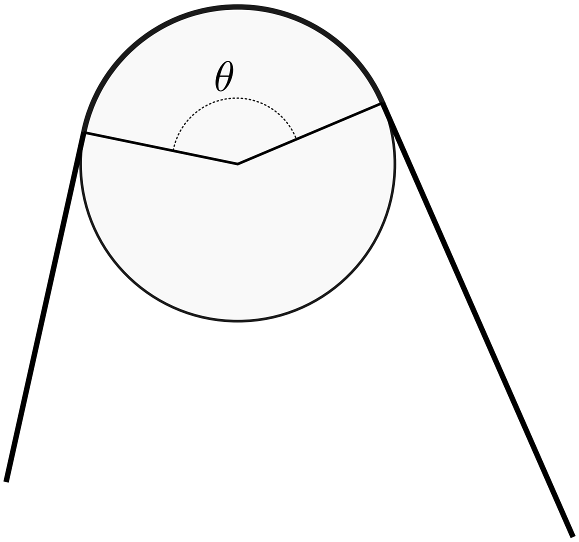


Figure 17-22. Front view showing wrap angle.

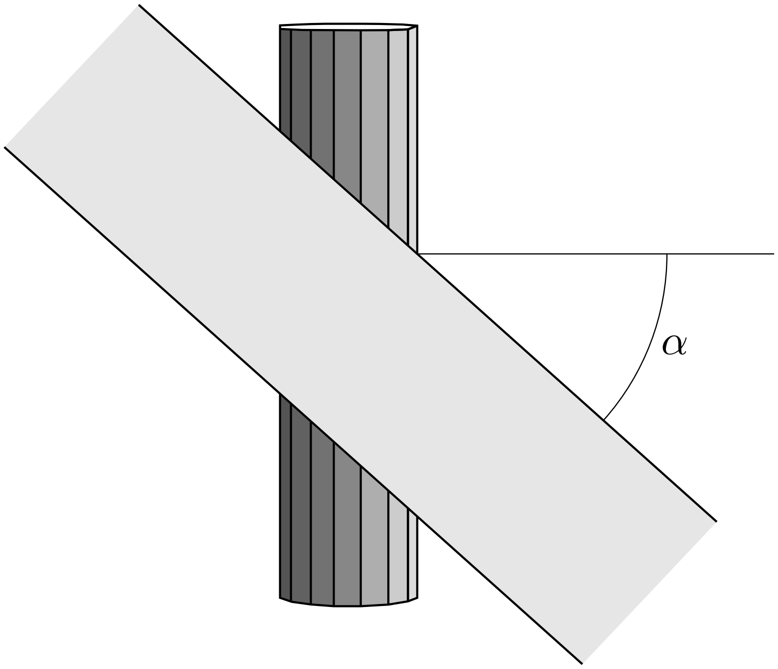


Figure 17-23. Top view shows orientation of belt relative to axis.

ELEMENT_SHELL_*{OPTION}

Available options include:

<BLANK>

THICKNESS

BETA or MCID

OFFSET

DOF

COMPOSITE

COMPOSITE_LONG

SHL4_TO_SHL8

Stacking of options, e.g., THICKNESS_OFFSET, is allowed in some cases. When combining options in this manner, check d3hsp to confirm that all the options are acknowledged.

Purpose: Define three, four, six, and eight node elements including 3D shells, membranes, 2D plane stress, plane strain, and axisymmetric solids. The type of the element and its formulation is specified through the part ID (see *PART) and the section ID (see *SECTION_SHELL). Also, the thickness of each element can be specified when applicable on the element cards or else a default thickness value is used from the section definition.

For orthotropic and anisotropic materials, a local material angle (variable BETA) can be defined which is cumulative with the integration point angles specified in *SECTION_-SHELL, *PART_COMPOSITE, *ELEMENT_SHELL_COMPOSITE, or *ELEMENT_-SHELL_COMPOSITE_LONG. Alternatively, the material coordinate system can be defined as the projection of a local coordinate system, MCID, onto the shell.

An offset option, OFFSET, is available for moving the shell reference surface from the nodal points that define the shell.

The COMPOSITE or COMPOSITE_LONG option allows an arbitrary number of integration points across the thickness of shells sharing the same part ID. This is independent of thickness defined in *SECTION_SHELL. To maintain a direct association of through-thickness integration point numbers with physical plies in the case where the number of plies varies from element to element, see Remark 12.

The option, SHL4_TO_SHL8, converts 3 node triangular and 4 node quadrilateral shell elements to 6 node triangular and 8 node quadrilateral quadratic shell elements, respectively, by the addition of mid-side nodal points. See Remark 9 below.

For the shell formulation that uses additional nodal degrees-of-freedom, the option DOF is available to connect the nodes of the shell to corresponding scalar nodes. Four scalar nodes are required for element type 25 to model the thickness changes that require 2 additional degrees-of-freedom per shell node. Defining these nodes is optional, if left undefined, they will be automatically created.

Card 1	1	2	3	4	5	6	7	8	9	10
Variable	EID	PID	N1	N2	N3	N4	N5	N6	N7	N8
Type	I	I	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	0	0	0	0
Remarks			3	3	3	3				

Thickness Card. Additional card for THICKNESS, BETA, and MCID keyword options.

Card 2	1	2	3	4	5	6	7	8	9	10
Variable	THIC1		THIC2		THIC3		THIC4		BETA or MCID	
Type	F		F		F		F		F	
Default	0.		0.		0.		0.		0.	
Remarks	1								2	

Thickness Card. Additional card for THICKNESS, BETA, and MCID keyword options, is only required if mid-side nodes are defined (N5-N8)..

Card 2	1	2	3	4	5	6	7	8	9	10
Variable	THIC5		THIC6		THIC7		THIC8			
Type	F		F		F		F			
Default	0.		0.		0.		0.		.	
Remarks	6									

Offset Card. Additional card for OFFSET keyword options.

Card 2	1	2	3	4	5	6	7	8	9	10
Variable	OFFSET									
Type	F									
Default	0.									
Remarks	7									

Scalar Node Card. Additional card for DOF keyword option.

Card 2	1	2	3	4	5	6	7	8	9	10
Variable			NS1	NS2	NS3	NS4				
Type			I	I	I	I				
Default										
Remarks			8	8	8	8				

COMPOSITE Cards. Additional set of cards for the COMPOSITE keyword option. Set the material ID, thickness, and material angle for each through-thickness integration point of a composite shell are provided below (up to two integration points per card). The integration point data should be given sequentially starting with the bottommost integration point. The total number of integration points is determined by the number of entries on these cards. The thickness of each shell is the summation of the integration point thicknesses. Define as many cards as needed.

Card 2	1	2	3	4	5	6	7	8
Variable	MID1	THICK1	B1		MID2	THICK2	B2	
Type	I	F	F		I	F	F	

COMPOSITE_LONG Cards. Additional set of cards for the COMPOSITE_LONG keyword option. Set the material ID, thickness, and material angle for each through-thickness integration point of a composite shell are provided below (one integration point per card). The integration point data should be given sequentially starting with the bottommost integration point. The total number of integration points is determined by the number of entries on these cards. The thickness of each shell is the summation of the integration point thicknesses. Define as many cards as needed. Column 4 must be left blank.

Card 2	1	2	3	4	5	6	7	8
Variable	MID1	THICK1	B1		PLYID1			
Type	I	F	F		I			

VARIABLE	DESCRIPTION
EID	Element ID. Chose a unique number with respect to other elements.
PID	Part ID, see *PART.
N1	Nodal point 1
N2	Nodal point 2
N3	Nodal point 3
N4	Nodal point 4
N5-N8	Mid-side nodes for eight node shell
THIC1	Shell thickness at node 1
THIC2	Shell thickness at node 2
THIC3	Shell thickness at node 3
THIC4	Shell thickness at node 4
BETA	Orthotropic material base offset angle (see Remarks 5 and 6 below). The angle is given in degrees. If blank the default is set to zero.
THIC5	Shell thickness at node 5
THIC6	Shell thickness at node 6
THIC7	Shell thickness at node 7
THIC8	Shell thickness at node 8
MCID	Material coordinate system ID. The a-axis of the base (or element-level) material coordinate system is the projection of the x-axis of coordinate system MCID onto the surface of the shell element. The c-axis of the material coordinate system aligns with the shell normal. The b-axis is taken as b = c x a. Each layer in the element can have a unique material orientation by defining a rotation angle for each layer as described in Remark 5.

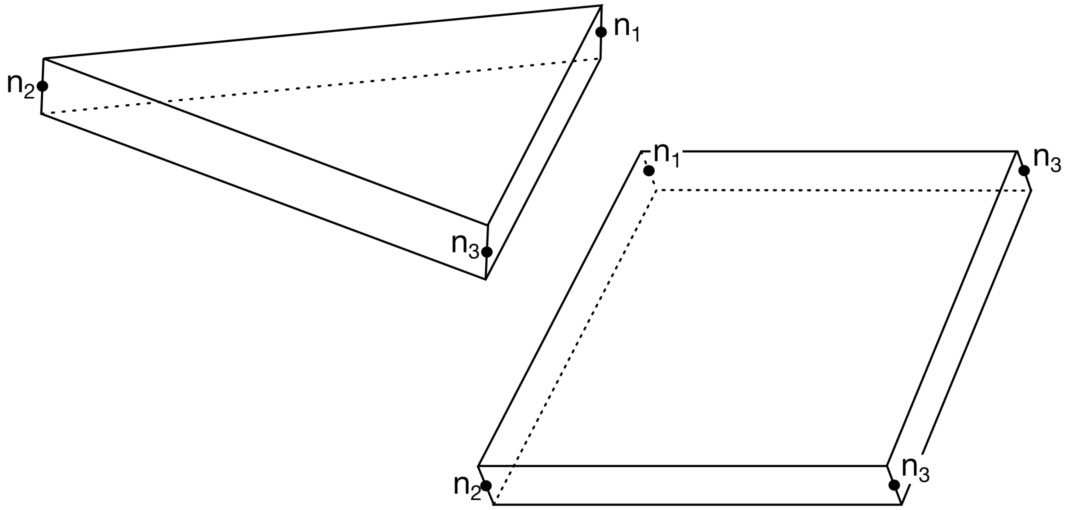


Figure 17-24. LS-DYNA shell elements. Counterclockwise node numbering determines the top surface.

OFFSET The offset distance from the plane of the nodal points to the reference surface of the shell in the direction of the normal vector to the shell.

NS1 Scalar node 1, parameter NDOF on the *NODE_SCALAR is normally set to 2. If the thickness is constrained, set NDOF = 0.

NS2 Scalar node 2

NS3 Scalar node 3

NS4 Scalar node 4

MID Material ID of integration point *i*, see *MAT_… Section.

THICK Thickness of integration point *i*.

B Material angle of integration point *i*.

PLYID Ply ID for integration point *i* (for post-processing purposes).

Remarks:

1. **Default Thickness.** Default values in place of zero shell thicknesses are taken from the cross-section property definition of the PID, see *SECTION_SHELL.

2. **Ordering.** Counterclockwise node numbering determines the top surface, see Figure 17-24

3. **Coordinate Systems.** Stresses and strain output in the binary databases are given in the global coordinate system, whereas stress resultants are output in the local coordinate system for the shell element.

4. **Convexity.** Interior angles must be less than 180 degrees.

5. **Material Orientation.** To allow the orientation of orthotropic and anisotropic materials to be defined for each shell element, a BETA angle can be defined. This BETA angle is used with the AOPT parameter and associated data on the *MAT card to determine an element reference direction for the element. The AOPT data defines a coordinate system and the BETA angle defines a subsequent rotation about the element normal to determine the element reference system. For composite modeling, each layer in the element can have a unique material direction by defining an additional rotation angle for the layer, using either the ICOMP and B*i* parameters on *SECTION_SHELL or the B*i* parameter on *PART_COMPOSITE. The material direction for layer i is then determined by a rotation angle, θ_i as shown in Figures 17-25 and 17-26.

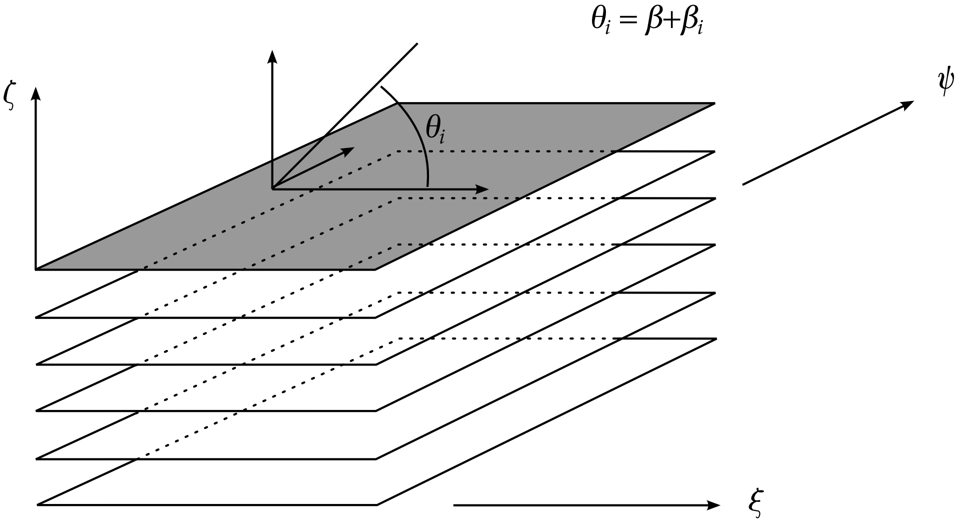


Figure 17-25. A multi-layer laminate can be defined. The angle β_i is defined for the *i*'th lamina (integration point), see *SECTION_SHELL.

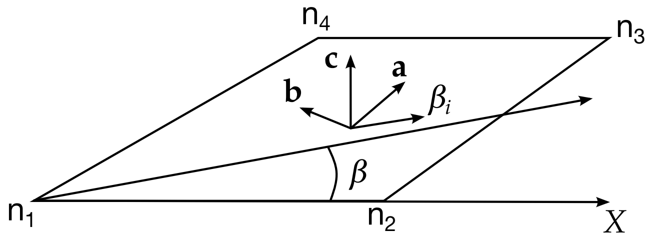


Figure 17-26. Orientation of material directions (shown relative to the 1-2 side as when AOPT = 0 in *MAT).

6. **Activation of the BETA Field and Its Interpretation.** To activate the BETA field, either the BETA or THCKNESS keyword option must be used. There is a difference in how a zero value or empty field is interpreted. When the BETA keyword option is used, a zero value or empty BETA field will override the BETA on *MAT. However, when the THICKNESS keyword option is used, a zero value or empty BETA field will not override the BETA value on *MAT. Therefore, to input BETA=0, the BETA keyword option is recommended. If a THIC value is omitted or input as zero, the thickness will default to the value on the *SECTION_SHELL card. If mid-side nodes are defined (N5-N8), then a second line of thickness values will be read. This line may be left blank, but cannot be omitted.

7. **Offset for the Reference Surface.** The parameter OFFSET gives the offset from the nodal points of the shell to the reference surface. This option applies to most shell formulations excluding two-dimensional elements, membrane elements, and quadratic shell elements. Except for Mortar contacts, the reference surface offset given by OFFSET is not taken into account in the contact

subroutines unless CNTCO is set to 1 in *CONTROL_SHELL. For Mortar contacts the OFFSET determines the location of the contact surface.

8. **Scalar Nodes.** The scalar nodes specified on the optional card refer to the scalar nodes defined by the user to hold additional degrees of freedom for shells with this capability. Scalar nodes are used with shell element type 25 and 26.

9. **Automatic Order Increase.** The option, SHL4_TO_SHL8, converts 3 node triangular and 4 node quadrilateral shell elements to 6 node triangular and 8 node quadrilateral quadratic shell elements, respectively, by the addition of mid-side nodal points. The user node ID's for these generated nodes are offset after the largest user node ID defined in the input file. When defining the *SECTION_SHELL keyword, the element type must be specified as either 23 or 24 corresponding to quadratic quadrilateral and triangular shells, respectively.

10. **Cohesive Elements.** Cohesive elements (ELFORM=29, 46 or 47 on *SECTION_SHELL) may be defined with zero depth to connect surfaces with no gap between them, but must have nodes 1 and 2 on one surface and nodes 3 and 4 on the other.

11. **Contact Thickness when using *ELEMENT_SHELL_THICKNESS in MPP.** When using MPP, THEORY = 1 in *CONTROL_SHELL has special meaning when dealing with non-uniform-thickness shells, that is, it serves to set the nodal contact thickness equal to the average of the nodal thicknesses from the shells sharing that node. Thus when a contact surface is comprised of non-uniform-thickness shells, THEORY = 1 is recommended and the user still has the option of setting the actual shell theory using ELFORM in *SECTION_-SHELL.

12. **Assignment of Zero Thickness to Integration Points.** The ability to assign zero- thickness integration points in the stacking sequence allows the number of integration points to remain constant even as the number of physical plies varies from element to element and eases post-processing since a particular integration point corresponds to a physical ply. Such a capability is important when one or more of the physical plies are not continuous across a part. To represent a missing ply in *ELEMENT_SHELL_COMPOSITE, set THICKi to 0.0 for the corresponding integration point and additionally, either set MID=-1 or, if the LONG option is used, set PLYID to any nonzero value.

    When postprocessing the results using LS-PrePost version 4.5, read both the keyword deck and d3plot database into the code and then select *Option > N/A gray fringe*. Then, when viewing fringe plots for a particular integration point (*FriComp > IPt > intpt#*), the element will be grayed out if the selected integration point is missing (or has zero thickness) in that element.

*ELEMENT_SHELL_NURBS_PATCH

Purpose: Define a NURBS-surface element (patch) based on a rectangular grid of control points. This grid consists of NPR*NPS control points, where NPR and NPS are the number of control points in local r- and s-direction, respectively. The necessary shape functions are defined through two knot-vectors:

1. Knot-Vector in r-direction with length NPR + PR + 1 and

2. Knot-Vector in s-direction with length NPS + PS + 1

There is no limit on the size of the underlying grid to define a NURBS-surface element, so the total number of necessary Keyword-cards depends on the parameters given in the first two cards and is given by

$$\#\text{ of cards } = 2 + \left\lceil \frac{\text{NPR} + \text{PR} + 1}{8} \right\rceil + \left\lceil \frac{\text{NPS} + \text{PS} + 1}{8} \right\rceil + \text{NPS} \times \left\lceil \frac{\text{NPR}}{8} \right\rceil,$$

where $\lceil x \rceil = \text{ceil}(x)$. (NOTE: the last term in the sum is doubled if WFL = 1, indicating that the weights are user-specified).

An example partial keyword deck using this card is given in Figure 17-28.

Card 1	1	2	3	4	5	6	7	8
Variable	NPID	PID	NPR	PR	NPS	PS		
Type	I	I	I	I	I	I		
Default	none	none	none	none	none	none		

Card 2	1	2	3	4	5	6	7	8
Variable	WFL	FORM	INT	NISR	NISS	IMASS	NL	
Type	I	I	I	I	I	I	I	
Default	0	0	0	PR	PS	0	0	
Remarks				Figure 17-28	Figure 17-28			

Knot Vector Cards (for r-direction). The knot-vector in local r-direction with length NPR + PR + 1 is given below (up to eight values per card) requiring a total of Ceil[(NPR + PR + 1)/8] cards.

Cards A	1	2	3	4	5	6	7	8
Variable	RK1	RK2	RK3	RK4	RK5	RK6	RK7	RK8
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

Knot Vector Cards (for s-direction). The knot-vector in local s-direction with length NPS + PS + 1 is given below (up to eight values per card) requiring a total of Ceil[(NPS + PS + 1)/8] cards.

Cards B	1	2	3	4	5	6	7	8
Variable	SK1	SK2	SK3	SK4	SK5	SK6	SK7	SK8
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

Connectivity Cards. The connectivity of the control grid is a two dimensional table of NPS rows and NPR columns. This data fills NPS sets (one *set* for each row) of NPR points tightly packed into Ceil(NPR/8) Connectivity Cards (format C), for a total of NPS × Ceil(NPR/8) cards.

Cards C	1	2	3	4	5	6	7	8
Variable	N1	N2	N3	N4	N5	N6	N7	N8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

Weight cards. Additional cards for WFL ≠ 0. Set a weight for each control point. These cards have an ordering identical to the Connectivity Cards (cards "C").

Cards D	1	2	3	4	5	6	7	8
Variable	W1	W2	W3	W4	W5	W6	W7	W8
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

Trimming Loop cards. Additional cards for NL.gt.0.

For every trimming loop (NL) define a set of Cards (E1 and E2).

Cards E1	1	2	3	4	5	6	7	8
Variable	NLE							
Type	I							

Trimming Loop cards.

Define NLE (Number of loop edges) edges (E_i). Use as many cards (E2) as necessary

Cards E2	1	2	3	4	5	6	7	8
Variable	E1	E2	E3	E4	E5	E6	E7	E8
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
NPID	Nurbs-Patch Element ID. A unique number has to be chosen
PID	Part ID, see *PART.
NPR	Number of control points in local r-direction.
PR	Order of polynomial of univariate nurbs basis functions in local r-direction.

NPS Number of control points in local s-direction.

PS Order of polynomial of univariate nurbs basis functions in local s-direction.

WFL Flag for weighting factors of the control points

EQ.0: all weights at the control points are set to 1.0 (B-spline basis) and no optional cards D are allowed

NE.0: the weights at the control points are defined in optional cards D which must be defined after cards C.

FORM Shell formulation to be used

EQ.0: shear deformable shell theory with rotational DOFs

EQ.1: shear deformable shell theory without rotational DOFs

EQ.2: thin shell theory without rotational DOFs

EQ.-4/4: combination of FORM = 0 and FORM = 1 (see remark 3)

INT In-plane numerical integration rule.

EQ.0: uniformly reduced Gauss integration, NIP = PR × PS.

EQ.1: full Gauss integration, NIP = (PR+1) × (PS+1).

EQ.2: reduced, patch-wise integration rule for C1-continuous quadratic NURBS

NISR Number of (automatically created) Interpolation Shell elements in local r-direction per created Nurbs-element for visualization (postprocessing) and contact (see remark 4 and Figure 17-28).

NISS Number of (automatically created) Interpolation Shell elements in local s-direction per created Nurbs-element for visualization (postprocessing) and contact (see remark 4 and Figure 17-28).

IMASS Option for lumping of mass matrix:

EQ.0: row sum

EQ.1: diagonal weighting.

NL Number of trimming loops (see remarks 5 and 6)

EQ.0: no trimming loops – standard untrimmed NURBS

GT.0: trimmed NURBS with NL trimming loops

RK_i	Values of the univariate knot vector in local r-direction defined in cards A.
SK_i	Values of the univariate knot vector in local s-direction defined in cards B.
N_i	Control points *i* (defined via *NODE) to define the control grid in cards C. LT.0 (FORM = 4/-4): control point with rotational DOFs (6 DOFs/control point, see remark 3)
W_i	Weighting factors of the control point i defined in cards D.
NLE	Number of loop edges to define trimming loop in cards E1 (NL.gt.0)
E_i	Edge (Curve) ID defining this edge – use *DEFINE_CURVE with DATTYP = 6

Remarks:

1. The thickness of the shell is defined in *SECTION_SHELL (referenced via *PART).

2. ELFORM = 201 has to be used in *SECTION_SHELL.

3. FORM = 4 allows the mixture of control points with and without rotational DOFs. This might be useful at the boundaries of Nurbs-patches where the continuity usually drops to C^0 and rotational DOFs are necessary. To indicate control points with rotational DOFs (6 DOFs/control point), the node number of the corresponding control point has to be set as the negative node ID in the connectivity cards C. Positive node IDs indicate control points without rotational DOFs (3 DOFs/control point).

   If FORM = -4 is used, the control points at the patch boundary are automatically treated with rotational DOFs without the need to specify them explicitly in the connectivity cards C. This might be sufficient in many cases.

4. The post-processing and the treatment of contact boundary conditions are presently dealt with interpolation elements, defined via interpolation nodes. These nodes and elements are automatically created, where NISR and NISS indicate the number of interpolation elements to be created per NURBS-element in the local r- and s-direction, respectively (see Figure 17-28).

5. Trimmed NURBS-patches can be analyzed by defining trimming loops (NL.gt.0) in Card 2. For each trimming loop to be defined, a set of cards E (E1 and E2), specifying the number of edges together with the edge (curve) IDs of the individual edges is necessary. One edge itself is defined via *DEFINE_CURVE using DATTYP = 6. The order of the edges as well as the order of the vertex nodes in *DEFINE_CURVE need to be in order, to define either a clockwise or a counter-clockwise orientation of the trimming loop. The orientation of the trimming loop is essential in defining, which part of the patch shall be trimmed away. Travelling along the trimming loop, the right-hand side of the trimming line will be cut away.

6. The trimming loops need to be defined in the parametric (local) rs-space of the NURBS-patch with r/s in [0,1]. The last and the first vertex node of consecutive edges need to coincide.

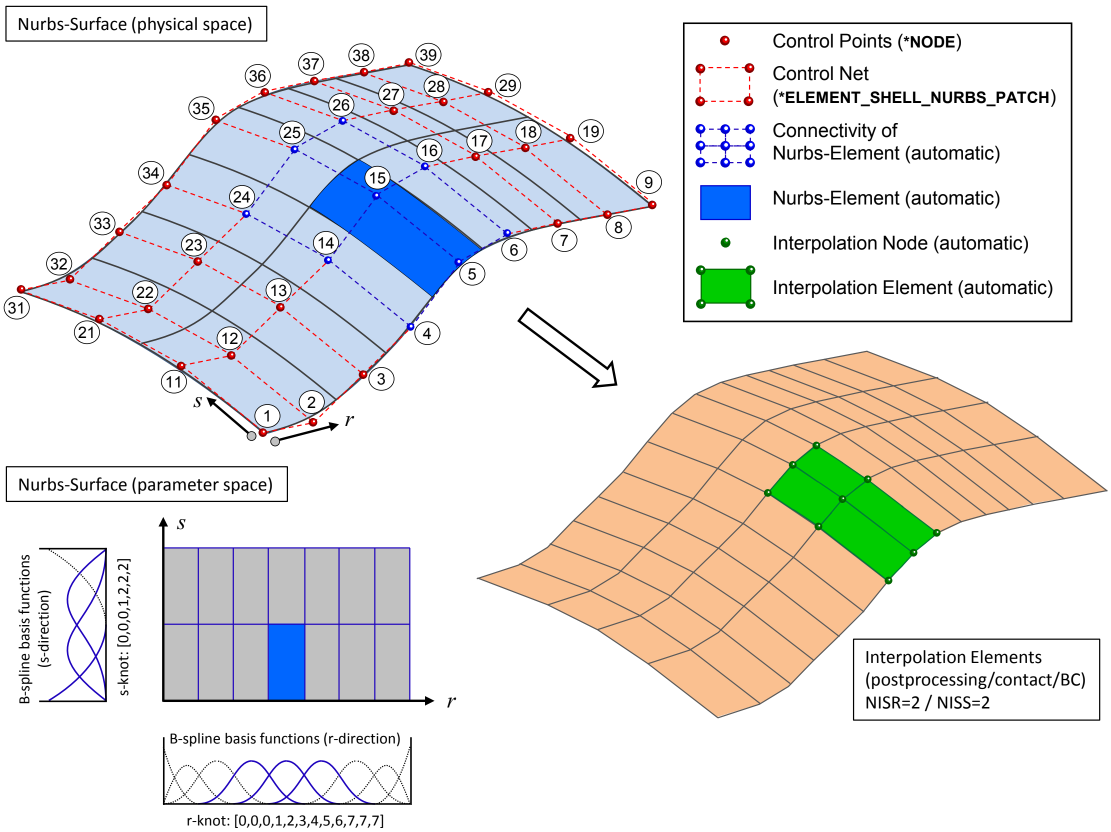
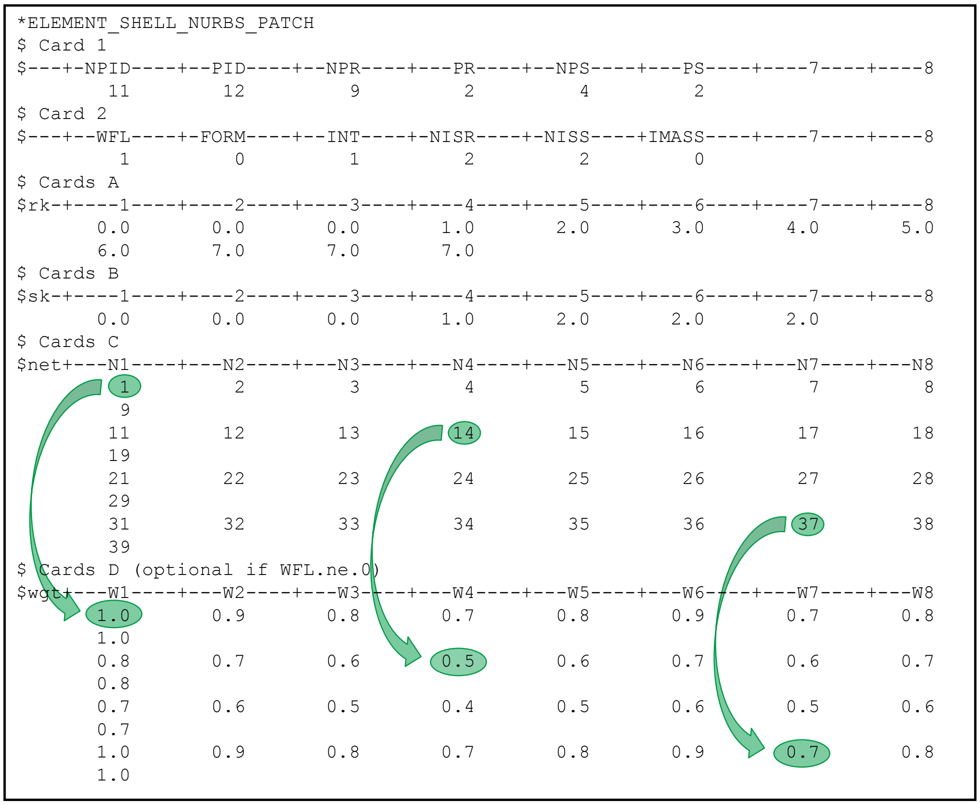

Nurbs-Surface (physical space)
1
2
3
4
5
6
7
8
9
11
12
13
14
15
16
17
18
19
21
22
23
24
25
26
27
28
29
31
32
33
34
35
36
37
38
39
s
r
Control Points (*NODE)
Control Net
(*ELEMENT_SHELL_NURBS_PATCH)
Connectivity of
Nurbs-Element (automatic)
Nurbs-Element (automatic)
Interpolation Node (automatic)
Interpolation Element (automatic)
Nurbs-Surface (parameter space)
B-spline basis functions
(s-direction)
s-knot: [0,0,0,1,2,2,2]
s
r
B-spline basis functions (r-direction)
r-knot: [0,0,0,1,2,3,4,5,6,7,7,7]
Interpolation Elements
(postprocessing/contact/BC)
NISR=2 / NISS=2

Figure 17-27. Illustration of example input deck from Figure 17-28.



```
*ELEMENT_SHELL_NURBS_PATCH
$ Card 1
$---+-NPID----+--PID----+--NPR----+---PR----+--NPS----+---PS----+----7----+----8
        11        12         9         2         4         2
$ Card 2
$---+--WFL----+-FORM----+--INT----+-NISR----+-NISS----+IMASS----+----7----+----8
         1         0         1         2         2         0
$ Cards A
$rk-+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
       0.0       0.0       0.0       1.0       2.0       3.0       4.0       5.0
       6.0       7.0       7.0       7.0
$ Cards B
$sk-+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
       0.0       0.0       0.0       1.0       2.0       2.0       2.0
$ Cards C
$net+---N1----+---N2----+---N3----+---N4----+---N5----+---N6----+---N7----+---N8
         1         2         3         4         5         6         7         8
         9
        11        12        13        14        15        16        17        18
        19
        21        22        23        24        25        26        27        28
        29
        31        32        33        34        35        36        37        38
        39
$ Cards D (optional if WFL.ne.0)
$wgt+---W1----+---W2----+---W3----+---W4----+---W5----+---W6----+---W7----+---W8
       1.0       0.9       0.8       0.7       0.8       0.9       0.7       0.8
       1.0
       0.8       0.7       0.6       0.5       0.6       0.7       0.6       0.7
       0.8
       0.7       0.6       0.5       0.4       0.5       0.6       0.5       0.6
       0.7
       1.0       0.9       0.8       0.7       0.8       0.9       0.7       0.8
       1.0
```

Figure 17-28. Example of a bi-quadratic *ELEMENT_SHELL_NURBS_PATCH keyword definition. See Figure 17-27 below.

*ELEMENT_SHELL_SOURCE_SINK

Purpose: Define a strip of shell elements of a single part ID to simulate a continuous forming operation. This option requires logical regular meshing of rectangular elements, which implies that the number of nodal points across the strip is constant along the length. Elements are created at the source and disappear at the sink. The advantage of this approach is that it is not necessary to define an enormous number of elements to simulate a continuous forming operation. Currently, only one source-sink definition is allowed. The boundary conditions at the source are discrete nodal point forces to keep the work piece in tension. At the sink, displacement boundary conditions are applied.

Card	1	2	3	4	5	6	7	8
Variable	NSSR	NSSK	PID					
Type	I	I	I					
Default	none	none	none					

VARIABLE	DESCRIPTION
NSSR	Node set at source. Provide an ordered set of nodes between corner nodes, which include the corner nodes.
NSSK	Node set at sink. Provide an ordered set of nodes between corner nodes, which include the corner nodes.
PID	Part ID of work piece.

ELEMENT_SOLID_*{OPTION}

Available options include:

<BLANK>

ORTHO

DOF

TET4TOTET10

H20

H8TOH20

H27

H8TOH27

Purpose: Define three-dimensional solid elements including 4 noded tetrahedrons and 8-noded hexahedrons. The type of solid element and its formulation is specified through the part ID (see *PART) and the section ID (see *SECTION_SOLID_*OPTION*). Also, a local coordinate system for orthotropic and anisotropic materials can be defined by using the ORTHO option. If extra degrees of freedom are needed, the DOF option should be used. The option TET4TOTET10 converts 4 node tetrahedrons to 10 node tetrahedrons, and H8TOH20/H8TOH27 converts 8-node hexahedrons to 20-node/27-node hexahedrons. See Remark 1.

Card 1	1	2	3	4	5	6	7	8	9	10
Variable	EID	PID								
Type	I	I								
Default	none	none								
Remarks	2									

Card 2	1	2	3	4	5	6	7	8	9	10
Variable	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10
Type	I	I	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none	none	none

20 Node Element Card. Additional card for the 20 option.

Card 3	1	2	3	4	5	6	7	8	9	10
Variable	N11	N12	N13	N14	N15	N16	N17	N18	N19	N20
Type	I	I	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none	none	none

27 Node Element Card. Additional card for the 27 option.

Card 3	1	2	3	4	5	6	7	8	9	10
Variable	N11	N12	N13	N14	N15	N16	N17	N18	N19	N20
Type	I	I	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none	none	none
Variable	N21	N22	N23	N24	N25	N26	N27			
Type	I	I	I	I	I	I	I			
Default	none	none	none	none	none	none	none			

Orthotropic Card 1. Additional card for ORTHO keyword option.

Card 4	1	2	3	4	5	6	7	8	9	10
Variable	A1 or BETA		A2		A3					
Type	F		F		F					
Default	0.		0.		0.					
Remarks	5, 7									

Orthotropic Card 2. Second additional card for ORTHO keyword option.

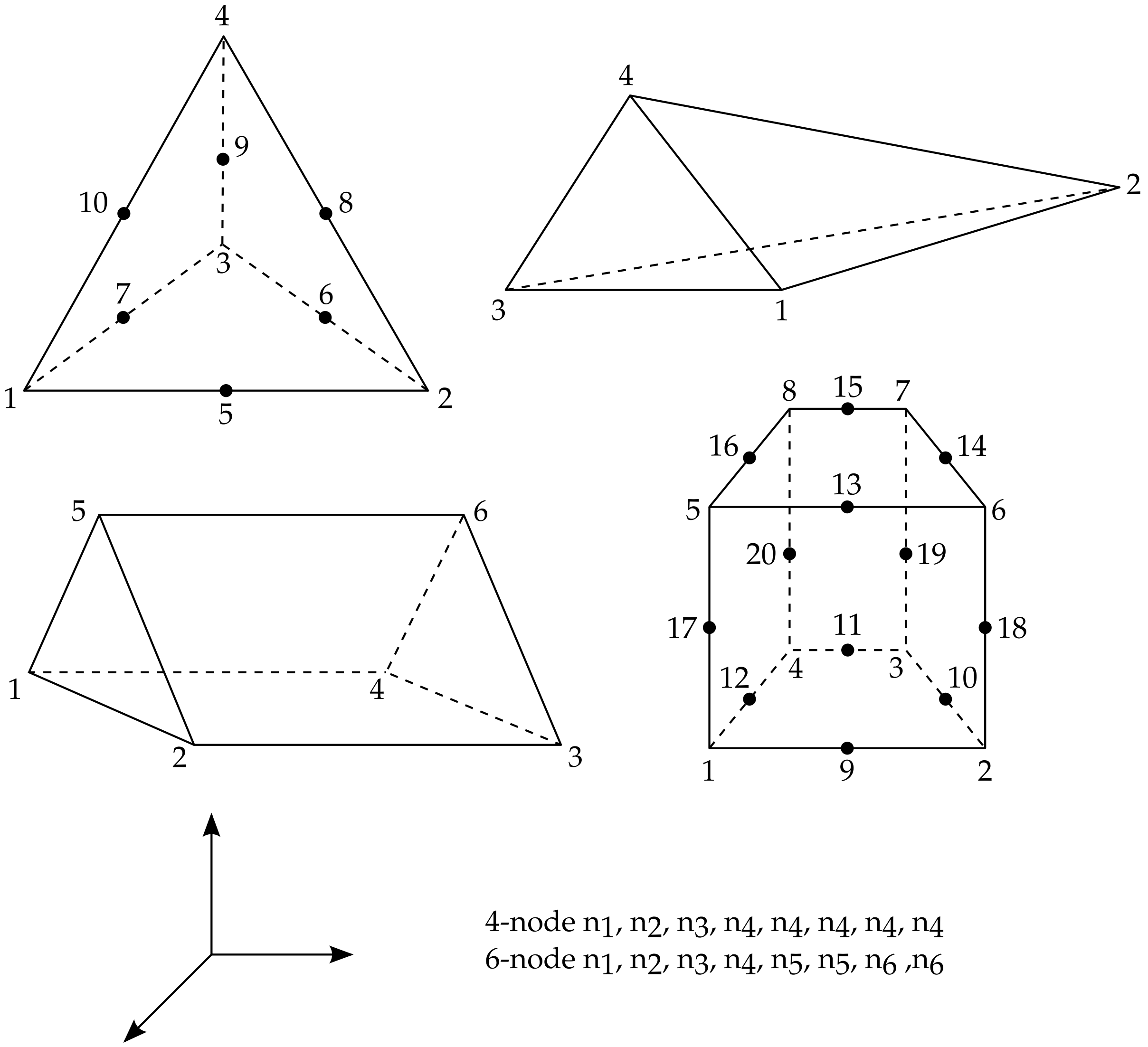
Card 5	1	2	3	4	5	6	7	8	9	10
Variable	D1		D2		D3					
Type	F		F		F					
Default	0.		0.		0.					
Remarks	5									

Scalar Node Card. Additional cards for DOF keyword option. This input ends at the next keyword "*" card.

Card 6	1	2	3	4	5	6	7	8	9	10
Variable			NS1	NS2	NS3	NS4	NS5	NS6	NS7	NS8
Type			I	I	I	I	I	I	I	I
Default			none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
EID	Element ID. A unique number has to be chosen.

VARIABLE	DESCRIPTION
PID	Part ID, see *PART.
N1	Nodal point 1
N2	Nodal point 2
N3	Nodal point 3
⋮	⋮
N27	Nodal point 27
A1 or BETA	*x*-component of local material direction a, or else rotation angle BETA in degrees (see remark 5).



4-node n1, n2, n3, n4, n4, n4, n4, n4
6-node n1, n2, n3, n4, n5, n5, n6 ,n6

Figure 17-29. Four, six, eight, ten, and twenty node solid elements. For the hexahedral and pentahedral shapes, nodes 1-4 are on the bottom surface.

VARIABLE	DESCRIPTION
A2	*y*-component of local material direction **a**.
A3	*z*-component of local material direction **a**.
D1	*x*-component of vector in the plane of the material vectors **a** and **b**.
D2	*y*-component of vector in the plane of the material vectors **a** and **b**.

VARIABLE	DESCRIPTION
D3	*z*-component of vector in the plane of the material vectors **a** and **b**.
NS1	Scalar node 1. See Remark 8.
NS2	Scalar node 2
NS3	Scalar node 3
NS4	Scalar node 4
NS5	Scalar node 5
NS6	Scalar node 6
NS7	Scalar node 7
NS8	Scalar node 8

Remarks:

1. **Automatic Node Generation**. The option TET4TOTET10 automatically converts 4 node tetrahedron solids to 10 node quadratic tetrahedron solids. Additional mid-side nodes are created which are shared by all tetrahedron elements that contain the edge. The user node ID’s for these generated nodes are offset after the largest user node ID defined in the input file. When defining the *SECTION_SOLID keyword, the element type must be specified as either 16 or 17 which are the 10-noded tetrahedrons in LS-DYNA. Mid-side nodes created as a result of TET4TOTET10 will not be automatically added to node sets that include the nodes of the original tetrahedron. So, for example, if the tetrahedrons are to have an initial velocity, velocity initialization by part ID or part set ID using *INITIAL_VELOCITY_GENERATION is necessary as opposed to velocity initialization by node set ID using *INITIAL_VELOCITY. The option H8TOH20/H8TOH27 provides the same functionality for converting 8-node to 20-node/27-node elements.

2. **Node Numbering.** Four, six, and eight node elements are shown in Figure 17-29 where the ordering of the nodal points is shown. 27-node elements are shown in Figure **0-3**. This ordering must be followed or code termination with occur during the initialization phase with a negative volume message. The input of nodes on the element cards for the tetrahedron and pentahedron elements is given by:

4-noded tetrahedron N1, N2, N3, N4, N4, N4, N4, N4, 0, 0

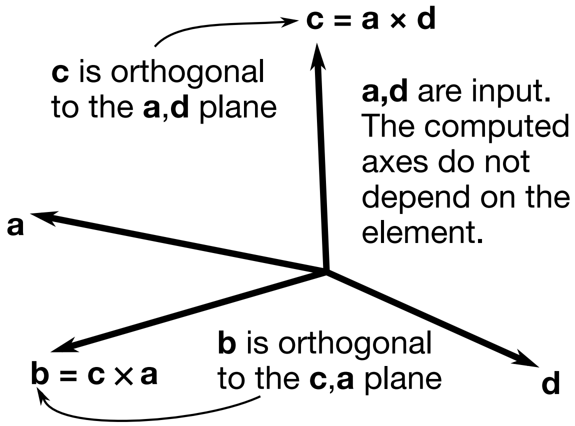

c = a × d
c is orthogonal to the a,d plane
a,d are input. The computed axes do not depend on the element.
a
b = c × a
b is orthogonal to the c,a plane
d

Figure 17-30. Two vectors a and d are defined and the triad is computed and stored.

<u>6-noded pentahedron</u> N1, N2, N3, N4, N5, N5, N6, N6, 0, 0

3. **Degenerate Solids.** If hexahedrons are mixed with tetrahedrons and pentahedrons in the input under the same part ID, *degenerate* tetrahedrons and pentahedrons are used. One problem with degenerate elements is related to an uneven mass distribution (node 4 of the tetrahedron has five times the mass of nodes 1-3) which can make these elements somewhat unstable with the default time step size. By using the control flag under the keyword, *CONTROL_SOLID, automatic sorting can be invoked to treat the degenerate elements as type 10 and type 15 tetrahedron and pentahedron elements, respectively.

4. **Obsolete Card Format.** For elements with 4-8 nodes the cards in the format of LS-DYNA versions 940-970 are still supported. The older format does not include Card 2.

Obsolete Element Solid Card.

Card 1	1	2	3	4	5	6	7	8	9	10
Variable	EID	PID	N1	N2	N3	N4	N5	N6	N7	N8
Type	I	I	I	I	I	I	I	I	I	I

5. **Local Directions.** For the orthotropic and anisotropic material models the local directions may be defined on the second card following the element connectivity definition. The local directions are then computed from the two vectors such that (see Figure 17-30):

$$\mathbf{c} = \mathbf{a} \times \mathbf{d} \text{ and } \mathbf{b} = \mathbf{c} \times \mathbf{a}.$$

These vectors are internally normalized within LS-DYNA. If the material model uses AOPT = 3, the *a* and *b* axes will be rotated about the *c* axis by the BETA angle on the material card.

6. **Stress Output Coordinates.** Stress output for solid elements is in the global coordinate system by default.

7. **Interpretation of A1 Field.** If vector **d** is input as a zero length vector, then A1 is interpreted as an offset rotation angle BETA in degrees which describes a rotation about the **c**-axis of the **a**-**b**-**c** coordinate system that is defined by AOPT and associated parameters on the *MAT input. This BETA angle applies to all values of AOPT, and it overrides the BETA angle on the *MAT card in the case of AOPT=3.

8. **Optional "Scalar" Nodes.** The scalar nodes specified on the optional card refer to extra nodes used by certain features (usual user defined) to store additional degrees of freedom.

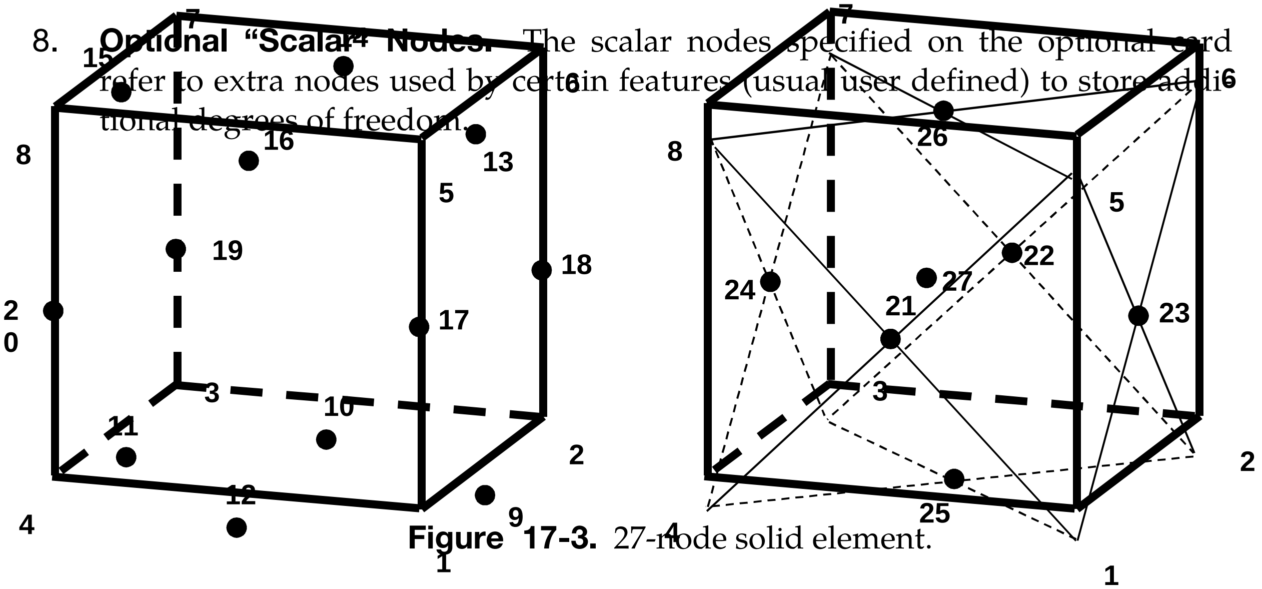


Figure 17-3. 27-node solid element.

*ELEMENT_SOLID_NURBS_PATCH

Purpose: Define a NURBS-block element (patch) based on a cuboid grid of control points. This grid consists of NPR*NPS*NPT control points, where NPR, NPS and NPT are the number of control points in local r-, s- and t-direction, respectively. The necessary shape functions are defined through three knot-vectors:

1. Knot-Vector in r-direction with length NPR + PR + 1 and

2. Knot-Vector in s-direction with length NPS + PS + 1 and

3. Knot-Vector in t-direction with length NPT + PT + 1 and

There is no limit on the size of the underlying grid to define a NURBS-block element, so the total number of necessary Keyword-cards depends on the parameters given in the first two cards and is given by

$$\text{\# of cards } = 2 + \left\lceil \frac{\text{NPR} + \text{PR} + 1}{8} \right\rceil + \left\lceil \frac{\text{NPS} + \text{PS} + 1}{8} \right\rceil + \left\lceil \frac{\text{NPT} + \text{PT} + 1}{8} \right\rceil + \text{NPT} \times \text{NPS} \times \left\lceil \frac{\text{NPR}}{8} \right\rceil,$$

where $\lceil x \rceil = \text{ceil}(x)$. (NOTE: the last term in the sum is doubled if WFL = 1, indicating that the weights are user-specified).

An example partial keyword deck using this card is given at the end.

Card 1	1	2	3	4	5	6	7	8
Variable	NPID	PID	NPR	PR	NPS	PS	NPT	PT
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

Card 2	1	2	3	4	5	6	7	8
Variable	WFL	NISR	NISS	NIST	IMASS			
Type	I	I	I	I	I			
Default	0	PR	PS	PT	0			
Remarks								

Knott Vector Cards (for r-direction). The knot-vector in local r-direction with length NPR + PR + 1 is given below (up to eight values per card) requiring a total of Ceil[(NPR + PR + 1)/8] cards.

Cards A	1	2	3	4	5	6	7	8
Variable	RK1	RK2	RK3	RK4	RK5	RK6	RK7	RK8
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

Knott Vector Cards (for s-direction). The knot-vector in local s-direction with length NPS + PS + 1 is given below (up to eight values per card) requiring a total of Ceil[(NPS + PS + 1)/8] cards.

Cards B	1	2	3	4	5	6	7	8
Variable	SK1	SK2	SK3	SK4	SK5	SK6	SK7	SK8
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

Knott Vector Cards (for t-direction). The knot-vector in local t-direction with length NPT +PT + 1 is given below (up to eight values per card) requiring a total of Ceil[(NPT+PT + 1)/8] cards.

Cards C	1	2	3	4	5	6	7	8
Variable	TK1	TK2	TK3	TK4	TK5	TK6	TK7	TK8
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

Connectivity Cards. The connectivity of the control grid is a two dimensional table of NPT × NPS rows and NPR columns. This data fills NPT × NPS sets (one *set* for each row) of NPR points tightly packed into Ceil(NPR/8) Connectivity Cards (format C), for a total of NPT × NPS × Ceil(NPR/8) cards.

Cards D	1	2	3	4	5	6	7	8
Variable	N1	N2	N3	N4	N5	N6	N7	N8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

Weight cards. Additional cards for WFL ≠ 0. Set a weight for each control point. These cards have an ordering identical to the Connectivity Cards (cards "D").

Cards E	1	2	3	4	5	6	7	8
Variable	W1	W2	W3	W4	W5	W6	W7	W8
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
NPID	Nurbs-Patch Element ID. A unique number has to be chosen

PID Part ID, see *PART.

NPR Number of control points in local r-direction.

PR Order of polynomial of univariate nurbs basis functions in local r-direction.

NPS Number of control points in local s-direction.

PS Order of polynomial of univariate nurbs basis functions in local s-direction.

NPT Number of control points in local t-direction.

PT Order of polynomial of univariate nurbs basis functions in local t-direction.

WFL Flag for weighting factors of the control points

EQ.0: all weights at the control points are set to 1.0 (B-spline basis) and no optional cards E are allowed

NE.0: the weights at the control points are defined in optional cards E which must be defined after cards D.

NISR Number of (automatically created) Interpolation Solid elements in local r-direction per created Nurbs-element for visualization (postprocessing) and contact.

NISS Number of (automatically created) Interpolation Solid elements in local s-direction per created Nurbs-element for visualization (postprocessing) and contact.

NIST Number of (automatically created) Interpolation Solid elements in local t-direction per created Nurbs-element for visualization (postprocessing) and contact.

IMASS Option for lumping of mass matrix:

EQ.0: row sum

EQ.1: diagonal weighting.

RK_i Values of the univariate knot vector in local r-direction defined in cards A.

SK_i Values of the univariate knot vector in local s-direction defined in cards B.

TK_i Values of the univariate knot vector in local t-direction defined in cards C.

N_i Control points *i* (defined via *NODE) to define the control grid in cards D.

W_i Weighting factors of the control point i defined in cards E.

Remarks:

1. ELFORM = 201 has to be used in *SECTION_SOLD.

2. The post-processing and the treatment of contact boundary conditions are presently dealt with interpolation elements, defined via interpolation nodes. These nodes and elements are automatically created, where NISR, NISS and NIST indicate the number of interpolation elements to be created per NURBS-element in the local r-, s- and t-direction, respectively.

3. An input deck can be shown as follows.

Example:

```
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
$ An Isogeometric Solid NURBS Example :
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
*SECTION_SOLID

$#   secid    elform
         1       201
*ELEMENT_SOLID_NURBS_PATCH

$CARD 1

$#   npeid       pid       npr        pr       nps        ps       npt        pt
         1         1         3         2         6         2         3         2

$CARD 2

$#     wfl      nisr      niss      nist     imass
         0         2         2         2         0

$CARD A

$#     rk1       rk2       rk3       rk4       rk5       rk6       rk7       rk8
       0.0       0.0       0.0       1.0       1.0       1.0

$CARD B

       0.0       0.0       0.0      0.25       0.5      0.75       1.0       1.0
       1.0

$CARD C

       0.0       0.0       0.0       1.0       1.0       1.0
$CARD D
      1001      1002      1003
        …
      1052      1053      1054
$CARD E (Optional if wfl .eq.  0)
```

*ELEMENT_SPH

Purpose: Define a lumped mass element assigned to a nodal point.

Card	1	2	3	4	5	6	7	8	9	10
Variable	NID	PID	MASS							
Type	I	I	F							
Default	none	none	0.							
Remarks			1							

VARIABLE **DESCRIPTION**

NID Node ID and Element ID are the same for the SPH option.

PID Part ID to which this node (element) belongs.

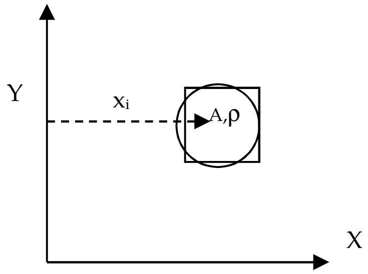
MASS

GT.0: Mass value

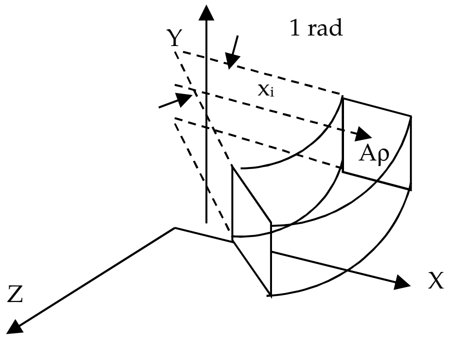
LT.0: Volume. The absolute value will be used as volume. The density (rho) will be retrieved from the material card defined in PID. SPH element mass is calculated by abs(MASS) × ρ.

Remarks:

1. Axisymmetric SPH, IDIM = -2 in CONTROL_SPH, is defined on global X-Y plane, with Y-axis as the axis of rotation. An axisymmetric SPH element has a mass of $A\rho$, where ρ is its density, A is the area of the SPH element and can be approximated by the area of its corresponding axisymmetric shell element, Fig. 1. The mass printout in d3hsp is the mass per radian, i.e., $A\rho x_i$, Fig. 1 & 2.



Axisymmetric SPH / corresponding shell
Mass printout in d3hsp, mass/radian



Mass printout in d3hsp, mass/radian

*ELEMENT_TRIM

NOTE: This keyword was replaced by *CONTROL_FORMING_TRIMING starting in Revision 87566.

ELEMENT_TSHELL_*{OPTION}

Available options include:

<BLANK>

BETA

COMPOSITE

Purpose: Define an eight node thick shell element which is available with either fully reduced or selectively reduced integration rules. Thick shell formulations 1, 2, and 6 are plane stress elements that can be used as an alternative to the 4 node shell elements in cases where an 8-node element is desired. Thick shell formulations 3, 5 and 7 are layered solids with 3D stress updates. Formulation 5, 6, and 7 are based on an enhanced strain. The number of through-thickness integration points is specified by the user.

For orthotropic and anisotropic materials, a local material angle (variable BETA) can be defined which is cumulative with the integration point angles specified in *SECTION_-TSHELL or *PART_COMPOSITE_TSHELL.

The COMPOSITE option for *ELEMENT_TSHELL allows a unique stackup of integration points for each element sharing the same part ID, and is available only when combined with *PART. The COMPOSITE option is not available in combination with *PART_COMPOSITE_TSHELL. To maintain a direct association of through-thickness integration point numbers with physical plies in the case where the number of plies varies from element to element, see Remark 5.

Card 1	1	2	3	4	5	6	7	8	9	10
Variable	EID	PID	N1	N2	N3	N4	N5	N6	N7	N8
Type	I	I	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none	none	none
Remarks			1							

Beta Card. Additional card for BETA keyword option.

Card 2	1	2	3	4	5	6	7	8	9	10
Variable									BETA	
Type									F	
Default									0.	
Remarks									4	

Composite Card. Additional card for COMPOSITE keyword option. The material ID, thickness, and material angle for each through-thickness integration point of a composite shell are defined. The integration point data should be given sequentially starting with the bottommost integration point. The total number of integration points is determined by the number of entries on these cards. The total thickness is the distance between the top and bottom surface as determined by the element connectivity, so the THICK*i* values are scaled to fit the element. Define as many cards as needed. The input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	MID1	THICK1	B1		MID2	THICK2	B2	
Type	I	F	F		I	F	F	

Card 2	1	2	3	4	5	6	7	8
Variable	MID3	THICK3	B3		Etc.			
Type	I	F	F		I	F	F	

VARIABLE	DESCRIPTION
EID	Element ID. Unique numbers have to be used.
PID	Part ID, see *PART.
N1	Nodal point 1

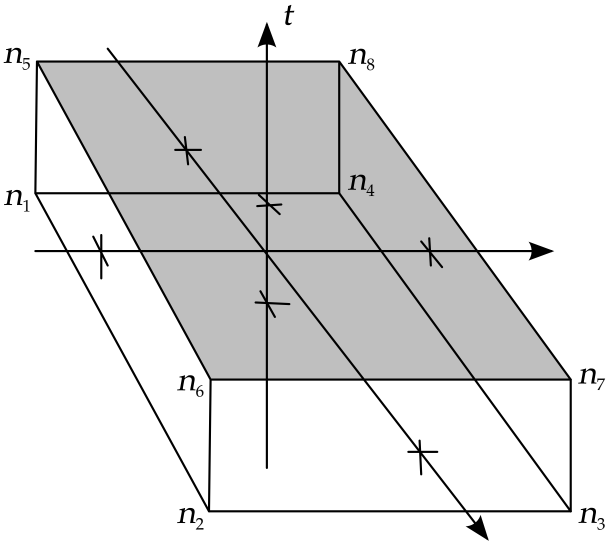


Figure 17-31. 8-node Thick Shell Element.

VARIABLE	DESCRIPTION
N2	Nodal point 2
N3	Nodal point 3
⋮	⋮
N8	Nodal point 8
BETA	Orthotropic material base offset angle (see Remark 4). The angle is given in degrees. If blank the default is set to zero.
MID*i*	Material ID of integration point *i*, see the *MAT_… cards.
THICK*i*	Thickness of integration point *i*
B*i*	Material angle of integration point *i*

Remarks:

1. **Orientation.** Extreme care must be used in defining the connectivity to ensure proper orientation of the through-thickness direction. For a hexahedron, nodes n_1 to n_4 define the lower surface, and nodes n_5 to n_8 define the upper surface. For a pentahedron, nodes n_1, n_2, n_3 form the lower triangular surface and the eight variables N1 to N8 should be defined using nodes n_1, n_2, n_3, n_3, n_4, n_5, n_6, n_6, respectively. Note that node n_3 and node n_6 are each repeated.

2. **Integration.** Element formulations 1 and 5 (see *SECTION_TSHELL), use one point integration and the integration points then lie along the *t*-axis as shown in

Figure 17-31. Element forumulations 2 and 3 use two by two selective reduced integration in each layer.

3. **Stress Output.** The stresses for thick shell elements are output in the global coordinate system.

4. **Local Coordinate System.** To allow the orientation of orthotropic and anisotropic materials to be defined for each thick shell element, a beta angle can be defined. This beta angle is used with the AOPT parameter and associated data on the *MAT card to determine an element reference direction for the element.

   The AOPT data defines a coordinate system and the BETA angle defines a subsequent rotation about the element normal to determine the element reference system. For composite modeling, each layer in the element can have a unique material direction by defining an additional rotation angle for the layer, using either the ICOMP and Bi parameters on *SECTION_TSHELL or the B*i* parameter on *PART_COMPOSITE_TSHELL. The material direction for layer *i* is then determined by a rotation angle, θ_i.

5. **Assignment of Zero Thickness to Integration Points.** The ability to assign zero- thickness integration points in the stacking sequence allows the number of integration points to remain constant even as the number of physical plies varies from element to element and eases post-processing since a particular integration point corresponds to a physical ply. Such a capability is important when one or more of the physical plies are not continuous across a part. To represent a missing ply in *ELEMENT_TSHELL_COMPOSITE, set THICKi to 0.0 for the corresponding integration point and additionally, set MID to -1.

   When postprocessing the results using LS-PrePost version 4.5, read both the keyword deck and d3plot database into the code and then select *Option > N/A gray fringe*. Then, when viewing fringe plots for a particular integration point (*FriComp > IPt > intpt#*), the element will be grayed out if the selected integration point is missing (or has zero thickness) in that element.

*END

The *END command is optional and signals the conclusion of a keyword input file. Data in a keyword file beyond a *END command are not read by LS-DYNA.

*EOS

Please see LS-DYNA Keyword User’s Manual, Volume II (Material Models).

*FREQUENCY_DOMAIN

Purpose: The keyword *FREQUENCY_DOMAIN provides a way of defining and solving frequency domain vibration and acoustic problems. The keyword cards in this section are defined in alphabetical order:

*FREQUENCY_DOMAIN_ACCELERATION_UNIT

FREQUENCY_DOMAIN_ACOUSTIC_BEM_*{OPTION}

*FREQUENCY_DOMAIN_ACOUSTIC_FEM

FREQUENCY_DOMAIN_ACOUSTIC_FRINGE_PLOT_*{OPTION}

*FREQUENCY_DOMAIN_ACOUSTIC_INCIDENT_WAVE

*FREQUENCY_DOMAIN_ACOUSTIC_SOUND_SPEED

*FREQUENCY_DOMAIN_FRF

FREQUENCY_DOMAIN_MODE_*{OPTION}

*FREQUENCY_DOMAIN_PATH

FREQUENCY_DOMAIN_RANDOM_VIBRATION_*{OPTION}

*FREQUENCY_DOMAIN_RESPONSE_SPECTRUM

*FREQUENCY_DOMAIN_SSD

*FREQUENCY_DOMAIN_ACCELERATION_UNIT

Purpose: LS-DYNA's default behavior is to *assume* that accelerations are derived:

$$[\text{acceleration unit}] = \frac{[\text{length unit}]}{[\text{time unit}]^2}.$$

This card extends LS-DYNA to support *other* units for acceleration.

Card 1	1	2						
Variable	UNIT	UMLT						
Type	I	F						

VARIABLE	DESCRIPTION
UNIT	Flag for acceleration unit conversion: EQ.0: use [length unit]/[time unit]^2 as unit of acceleration. EQ.1: use g as unit for acceleration, and SI units (Newton, kg, meter, second, etc.) elsewhere. EQ.2: use g as unit for acceleration, and Engineering units (lbf, lbf × second^2/inch, inch, second, etc.) elsewhere. EQ.3: use g as unit for acceleration, and units (kN, kg, mm, ms, GPa, etc.) elsewhere. EQ.-1: use g as unit for acceleration and provide the multiplier for converting g to [length unit]/[time unit]^2.
UMLT	Multiplier for converting g to [length unit]/[time unit]^2 (used only for UNIT = -1).

Remarks:

LS-DYNA uses consistent units. With *consistent units* acceleration is defined using:

$$[\text{acceleration unit}] = \frac{[\text{length unit}]}{[\text{time unit}]^2}.$$

However, it is the convention of many industries to use g (gravitational acceleration on the Earth's surface) as the base unit for acceleration. Usually, data from vibration tests, both random and sine sweep, are expressed in systems for which g is the unit of acceleration. With this keyword, LS-DYNA supports such conventions. Internally, LS-DYNA implements this keyword by converting the input deck into consistent units, and then proceeding with the calculation as usual. However, results are output in the unit system specified with this keyword.

FREQUENCY_DOMAIN_ACOUSTIC_BEM_*{OPTION1}_{OPTION2}

Available options include:

ATV

MATV

HALF_SPACE

PANEL_CONTRIBUTION

Purpose: Activate the boundary element method in frequency domain for acoustic problems. This keyword is ignored unless the BEM=**filename** option is included in the LS-DYNA command line:

Control Card. To use this card LS-DYNA must be run with a BEM option, as in "LS-DYNA I=**inf** BEM=**filename**".

Card 1	1	2	3	4	5	6	7	8
Variable	R0	C	FMIN	FMAX	NFREQ	DTOUT	TSTART	PREF
Type	F	F	F	F	I	F	F	F
Default	none	none	none	none	0	0	0	0
Remark							1	2

Card 2	1	2	3	4	5	6	7	8
Variable	NSIDEXT	TYPEXT	NSIDINT	TYPINT	FFTWIN	TRSLT	IPFILE	IUNITS
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0
Remark					3	4		5

Additional card for FFTWIN = 5.

Card 2a	1	2	3	4	5	6	7	8
Variable	T_HOLD	DECAY						
Type	F	F						
Default	0.0	0.02						

Card 3	1	2	3	4	5	6	7	8
Variable	METHOD	MAXIT	TOLITR	NDD	TOLLR	TOLFCT	IBDIM	NPG
Type	I	I	F	I	F	F	I	I
Default		100	10^{-4}	1	10^{-6}	10^{-6}	1000	2
Remark	6			7				

Card 4	1	2	3	4	5	6	7	8
Variable		NBC	RESTRT	IEDGE	NOEL	NFRUP	VELOUT	DBA
Type		I	I	I	I	I	I	I
Default		1	0	0	0	0	0	0
Remark			8	9	10	11		

Boundary Condition Cards. The deck must include NBC cards in this format: one for each boundary condition.

Card 5	1	2	3	4	5	6	7	8
Variable	SSID	SSTYPE	NORM	BEMTYP	LC1	LC2		
Type	I	I	I	I	I	I		
Default	0	0	0	0				
Remark	12							

Panel Contribution Card. Additional for PANEL_CONTRIBUTION keyword option.

Card 6	1	2	3	4	5	6	7	8
Variable	NSIDPC							
Type	I							
Default	0							
Remark	13							

Half Space Card. Additional card for HALF_SPACE keyword option.

Card 6	1	2	3	4	5	6	7	8
Variable	PID							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
RO	Fluid density.

VARIABLE **DESCRIPTION**

C

Sound speed of the fluid.

GT.0: real constant sound speed.

LT.0: |C| is the load curve ID, which defines the frequency dependent complex sound speed. See *FREQUENCY_-DOMAIN_ACOUSTIC_SOUND_SPEED.

FMIN

Minimum value of output frequencies.

FMAX

Maximum value of output frequencies.

NFREQ

Number of output frequencies.

DTOUT

Time interval between writing velocity or acceleration, and pressure at boundary elements in the binary file, to be proceeded at the end of LS-DYNA simulation.

TSTART

Start time for recording velocity or acceleration in LS-DYNA simulation.

PREF

Reference pressure to be used to output pressure in dB, in the file Press_dB. If PREF = 0, the Press_dB file will not be generated. A file called Press_Pa is generated and contains the pressure at the output nodes (see Card 2).

NSIDEXT

Node or segment set ID of output exterior field points.

TYPEXT

Output exterior field point type.

EQ.0: node ID.

EQ.1: node set ID.

EQ.2: segment set ID.

NSIDINT

Node or segment set ID of output interior field points.

TYPINT

Output interior field point type.

EQ.0: node ID.

EQ.1: node set ID.

EQ.2: segment set ID.

VARIABLE **DESCRIPTION**

FFTWIN

FFT windows (Default = 0).

EQ.0: rectangular window.

EQ.1: Hanning window.

EQ.2: Hamming window.

EQ.3: Blackman window.

EQ.4: raised cosine window.

EQ.5: exponential window.

TRSLT

Request time domain results:

EQ.0: no time domain results are requested.

EQ.1: time domain results are requested (Press_Pa_t gives absolute value pressure vs. time).

EQ.2: time domain results are requested (Press_Pa_t gives real value pressure vs. time).

VARIABLE **DESCRIPTION**

IPFILE

Flag for output files (default = 0):

EQ.0: Press_Pa (magnitude of pressure vs. frequency), Press_dB (sound pressure level vs. frequency) and bepres (ASCII database file for LS-Prepost) are provided.

EQ.1: Press_Pa_real (the real part of the pressure vs. frequency) and Press_Pa_imag (the imaginary part of the pressure vs. frequency) are provided, in addition to Press_Pa, Press_dB and bepres.

EQ.10: files for IPFILE = 0, and fringe files for acoustic pressure.

EQ.11: files for IPFILE = 1, and fringe files for acoustic pressure.

EQ.20: files for IPFILE = 0, and fringe files for sound pressure level.

EQ.21: files for IPFILE = 1, and fringe files for sound pressure level.

EQ.31: files for IPFILE = 1, and fringe files for acoustic pressure (real part).

EQ.41: files for IPFILE = 1, and fringe files for acoustic pressure (imaginary part).

IUNITS

Flag for unit changes

EQ.0: do not apply unit change.

EQ.1: MKS units are used, no change needed.

EQ.2: units: lbf × s^2/in, inch, s, lbf, psi, etc. are used, changed to MKS in BEM Acoustic computation.

EQ.3: units: kg, mm, ms, kN, GPa, etc. are used, changed to MKS in BEM acoustic computation.

EQ.4: units: ton, mm, s, N, MPa, etc. are used, changed to MKS in BEM acoustic computation.

T_HOLD

Hold-off period before the exponential window. The length of the hold-off period should coincide with the pre-trigger time to reduce the effects of noise in the captured time domain data. It is only used when FFTWIN = 5.

VARIABLE DESCRIPTION

DECAY

Decay ratio at the end of capture duration. For example, if the DECAY = 0.02, it means that the vibration is forced to decay to 2% of its amplitude within the capture duration. This field is only used when FFTWIN = 5.

METHOD

Method used in acoustic analysis

EQ.0: Rayleigh method (very fast).

EQ.1: Kirchhoff method coupled to FEM for acoustics (*MAT_ACOUSTIC). See Remark 6.

EQ.2: variational Indirect BEM.

EQ.3: collocation BEM.

EQ.4: collocation BEM with Burton-Miller formulation for exterior problems (no irregular frequency phenomenon).

MAXIT

Maximum number of iterations for iterative solver (default = 100) if METHOD ≥ 2.

TOLITR

Tolerance for the iterative solver. The default value is 10^{-4}.

NDD

Number of domain decomposition, used for memory saving. For large problems, the boundary mesh is decomposed into NDD domains for less memory allocation. This option is only used if METHOD ≥ 2.

TOLLR

Tolerance for low rank approximation of dense matrix (default = 10^{-6}).

TOLFCT

Tolerance in factorization of the low rank matrix (default = 10^{-6}).

IBDIM

Inner iteration limit in GMRES iterative solver (default = 1000).

NPG

Number of Gauss integration points (default = 2).

NBC

Number of boundary condition cards. See Card 5. (default = 1).

RESTRT

This flag is used to save an LS-DYNA analysis if the binary output file in the (bem=**filename**) option has not been changed (default = 0).

EQ.0: LS-DYNA time domain analysis is processed and generates a new binary file.

EQ.1: LS-DYNA time domain analysis is not processed. The

VARIABLE **DESCRIPTION**

binary files from previous run are used. The files include the binary output file filename, and the binary file bin_velfreq, which saves the boundary velocity from FFT.

EQ.2: LS-DYNA restarts from d3dump file by using "R=" command line parameter. This is useful when the last run was interrupted by sense switches such as "sw1".

EQ.3: LS-DYNA reads in user provided velocity history saved in an ASCII file, bevel.

EQ.4: run acoustic computation on a boundary element mesh with velocity information given with a denser finite element mesh in last run. This option requires both "bem = filename" and "lbem = filename2" in the command line, where filename2 is the name of the binary file generated in the last run with denser mesh.

EQ.5: LS-DYNA time domain analysis is not processed. The binary file filename from previous run is used. An FFT is performed to get the new frequency domain boundary velocity and the results are saved in bin_velfreq.

IEDGE

Free edge and multi-connection constraints option (default = 0).

EQ.0: free edge and multi-connection constraints not considered.

EQ.1: free edge and multi-connection constraints considered.

EQ.2: only free edge constraints are considered.

EQ.3: only multi-connection constraints are considered.

NOEL

Location where normal velocity or acceleration is taken (default = 0).

EQ.0: elements or segments.

EQ.1: nodes.

NFRUP

Preconditioner update option.

EQ.0: updated at every frequency.

GE.1: updated for every NFRUP frequencies.

VARIABLE **DESCRIPTION**

VELOUT

Flag for writing out nodal or elemental velocity data.

EQ.0: No writing out velocity data.

EQ.1: write out time domain velocity data (in x, y and z directions).

EQ.2: write out frequency domain velocity data (in normal direction).

DBA

Flag for writing out weighted SPL files with different weighting options.

EQ.0: No writing out weighted SPL files.

EQ.1: write out Press_dB(A) by using A-weighting.

EQ.2: write out Press_dB(B) by using B-weighting.

EQ.3: write out Press_dB(C) by using C-weighting.

EQ.4: write out Press_dB(D) by using D-weighting.

SSID

Part, part set ID, or segment set ID of boundary elements.

SSTYPE

Boundary element type:

EQ.0: part Set ID

EQ.1: part ID

EQ.2: segment set ID.

NORM

NORM should be set such that the normal vectors point away from the fluid.

EQ.0: normal vectors are not inverted (default).

EQ.1: normal vectors are inverted.

VARIABLE **DESCRIPTION**

BEMTYP

Type of input boundary values in BEM analysis.

EQ.0: boundary velocity will be processed in BEM analysis.

EQ.1: boundary acceleration will be processed in BEM analysis.

EQ.2: pressure is prescribed and the real and imaginary parts are given by LC1 and LC2.

EQ.3: normal velocity is prescribed and the real and imaginary parts are given by LC1 and LC2.

EQ.4: impedance is prescribed and the real and imaginary parts are given by LC1 and LC2.

LT.0: normal velocity (only real part) is prescribed, through load curve n. An amplitude as a function of frequency load curve with curve ID |BEMTYP|.

LC1

Load curve ID for defining real part of pressure, normal velocity or impedance.

LC2

Load curve ID for defining imaginary part of pressure, normal velocity or impedance.

NSIDPC

Node set ID for the field points where panel contributions to SPL (Sound Pressure Level) are requested.

PID

Plane ID for defining the half-space problem, see keyword *DEFINE_PLANE.

Remarks:

1. **TSART Field.** TSTART indicates the time at which velocity or acceleration and pressure are stored in the binary file.

2. **PREF Field.** This reference pressure is required for the computation of the pressure in dB. Usually, in International Unit System the reference pressure is 20μPa.

3. **FFT Windowing.** Velocity or acceleration (pressure) is provided by LS-DYNA analysis. They are written in a binary file (bem=**filename**). The boundary element method is processed after the LS-DYNA analysis. An FFT algorithm is used to transform time domain data into frequency domain in order to use the boundary element method for acoustics. In order to overcome the FFT leakage problem due to the truncation of the temporal response, several windows are

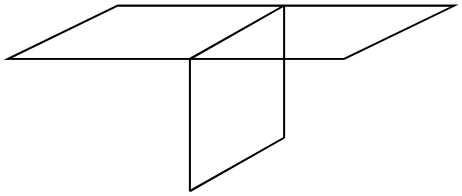


Figure 20-1. T-section.

proposed. Windowing is used to have a periodic velocity, acceleration and pressure in order to use the FFT.

4. **TRSLT Field.** If time domain results are requested, FMIN is changed to 0 in the code.

5. **IUNITS Field.** Units are automatically converted into kg, m, s, N, and Pa so that the reference pressure will not be too small. For example, it may be as low as 20.E-15 GPa if one uses the units kg, mm, ms, kN, and GPa and this may result in truncation error in the computation, especially in single precision version.

6. **METHOD Field.** The Rayleigh method is an approximation suitable only for external radiation problems. It is very fast since there is no linear system to solve. The Kirchhoff method involves coupling the BEM and FEM for acoustics (*MAT_ACOUSTIC) with a Non Reflecting Boundary condition, see *BOUNDARY_NON_REFLECTING. In this case, at least one fluid layer with non-reflecting boundary condition is merged with the vibrating structure. This additional fluid is given in *MAT_ACOUSTIC by the same density and sound speed as used in this keyword. When used appropriately both methods provide a good approximation to a full BEM calculation for external problems.

7. **NDD Field.** BEM formulation for large and medium size problems (more than 2000 boundary elements) is memory and time consuming. In this case, user may run LS-DYNA using the memory option. In order to save memory, domain decomposition can be used.

8. **RESTRT Field.** The binary file generated by a previous run can be used for the next run by using the restart option. The restart option allows the user to use the binary file generated from a previous calculation in order to run BEM. In this case, the frequency range can be changed. However, the time paraemeters should not be modified between calculations.

9. **IEDGE Field.** This option only applies to METHOD = 2, the Variational Indirect BEM.

10. **NOEL Field.** This field specified whether the element or nodal velocity (or acceleration) is taken from FEM computation. NOEL should be 0 if Kirchhoff method (METHOD = 1) is used since elemental pressure is processed in FEM.

NOEL should be 0 if Burton-Miller collocation method (METHOD = 4) is used since a constant strength element formulation is adopted. In other cases, it is strongly recommended to use element velocity or acceleration (NOEL = 0) if "T-Section" appears in boundary element mesh. See Figure 20-1.

11. **NFRUP Field.** The preconditioner is obtained with the factorization of the influence coefficient matrix. To conserve CPU time, It can be retained for several frequencies. By default (NFRUP=0), the preconditioner is updated for every frequency. Note that in MPP version, the preconditioner is updated every NFRUP frequencies on each processor.

12. **Boundary Condition Cards.** The Card 5 can be defined if the boundary elements are composed of several panels. It can be defined multiple times if more than 2 panels are used. Each card 5 defines one panel.

13. **NSIDPC Field.** The field points where the panel contribution analysis is requested must be one of the field points for acoustic computation (it must be included in the nodes specified by the NSIDEXT or NSIDINT). The panels are defined by card 4 and card 5, etc. Each card defines one panel.

14. **Element Sizing.** To obtain accurate results, the element size should not be greater than 1/6 of the wave length λ ($\lambda = c/f$ where c is the wave speed and f is the frequency).

15. **Acoustic Transfer Vector.** The Acoustic Transfer Vector can be obtained by including the option ATV in the keyword. It calculates acoustic pressure (and sound pressure level) at field points due to unit normal velocity of each surface node. ATV is dependent on structure model, properties of acoustic fluid as well as location of field points. When ATV option is included, the structure does not need any external excitation, and the curve IDs LC1 and LC2 are ignored. A binary plot database d3atv can be obtained by setting BINARY = 1 in *DATABASE_FREQUENCY_BINARY_D3ATV.

16. **Modal Acoustic Transfer Vector.** The Modal Acoustic Transfer Vector (MATV) is calculated when the MATV keyword option is included. The MATV option requires that the implicit eigenvalue solver be used, which is activated by keywords *CONTROL_IMPLICIT_GENERAL, and *CONTROL_IMPLICIT_EIGENVALUE. It calculates acoustic pressure (and sound pressure level) at field points due to vibration in the form of eigenmodes. For each excitation frequency f, LS-DYNA generates the psedo-velocity boundary condition $2\pi if\{\phi\}_j$, where $i = \sqrt{-1}$ is the imaginary unit and runs acoustic computation for each field point, based on the psedo-velocity boundary conditions, to get the MATV matrices. The MATV matrices are saved in binary file "bin_bepressure" for future use. Like ATV, MATV is also only dependent on structure model, properties of acoustic fluids as well as the location of field points.

17. **Output Files.** The result files: Press_Pa, Press_dB, Press_Pa_real, Press_Pa_imag, Press_Pa_t and Press_dB_t have a xyplot format that LS-PrePost can read and plot.

FREQUENCY_DOMAIN_ACOUSTIC_FEM_*{OPTION}

Available options include:

EIGENVALUE

Purpose: Define an interior acoustic problem and solve the problem with a frequency domain finite element method. When EIGENVALUE option is used, compute eigenvalues and eigenvectors of the acoustic system.

Card 1	1	2	3	4	5	6	7	8
Variable	RO	C	FMIN	FMAX	NFREQ	DTOUT	TSTART	PREF
Type	F	F	F	F	I	F	F	F
Default	none	none	none	none	0	0	0	0

Card 2	1	2	3	4	5	6	7	8
Variable		FFTWIN						
Type		I						
Default		0						

Card 3	1	2	3	4	5	6	7	8
Variable	PID	PTYP						
Type	I	I						
Default	none	0						

Boundary Condition Definition Card. It can be repeated if multiple boundary conditions are present. This card is optional when option EIGENVALUE is present.

Card 4	1	2	3	4	5	6	7	8
Variable	SID	STYP	VAD	DOF	LCID1	LCID2	SF	VID
Type	I	I	I	I	I	I	F	I
Default	none	0	0	none	0	0	1.0	0

Field Points Definition Card. Not used when option EIGENVALUE is present.

Card 5	1	2	3	4	5	6	7	8
Variable	NID	NTYP	IPFILE	DBA				
Type	I	I	I	I				
Default	none	0	0	0				

VARIABLE	DESCRIPTION
RO	Fluid density.
C	Sound speed of the fluid. GT.0: real constant sound speed. LT.0: |C| is the load curve ID, which defines the frequency dependent complex sound speed. See *FREQUENCY_DOMAIN_ACOUSTIC_SOUND_SPEED.
FMIN	Minimum value of output frequencies.
FMAX	Maximum value of output frequencies.
NFREQ	Number of output frequencies.
DTOUT	Time step for writing velocity or acceleration in the binary file.
TSTART	Start time for recording velocity or acceleration in transient analysis.

VARIABLE	DESCRIPTION
PREF	Reference pressure, for converting the acoustic pressure to dB.
FFTWIN	FFT windows (Default = 0): EQ.0: rectangular window. EQ.1: Hanning window. EQ.2: Hamming window. EQ.3: Blackman window. EQ.4: Raised cosine window.
PID	Part ID, or part set ID to define the acoustic domain.
PTYP	Set type: EQ.0: part, see *PART. EQ.1: part set, see *SET_PART.
SID	Part ID, or part set ID, or segment set ID, or node set ID to define the boundary where vibration boundary condition is provided
STYP	Set type: EQ.0: part, see *PART. EQ.1: part set, see *SET_PART. EQ.2: segment set, see *SET_SEGMENT. EQ.3: node set, see *SET_NODE.
VAD	Boundary condition flag: EQ.0: velocity by steady state dynamics (SSD). EQ.1: velocity by transient analysis. EQ.2: opening (zero pressure). EQ.11: velocity by LCID1 (amplitude) and LCID2 (phase). EQ.12: velocity by LCID1 (real) and LCID2 (imaginary). EQ.21: acceleration by LCID1 (amplitude) and LCID2 (phase). EQ.22: acceleration by LCID1 (real) and LCID2 (imaginary). EQ.31: displacement by LCID1 (amplitude) and LCID2 (phase). EQ.32: displacement by LCID1 (real) and LCID2 (imaginary).

VARIABLE **DESCRIPTION**

EQ.41: impedance by LCID1 (amplitude) and LCID2 (phase).

EQ.42: impedance by LCID1 (real) and LCID2 (imaginary).

EQ.51: pressure by LCID1 (amplitude) and LCID2 (phase).

EQ.52: pressure by LCID1 (real) and LCID2 (imaginary).

DOF

Applicable degrees-of-freedom:

EQ.0: determined by steady state dynamics.

EQ.1: x-translational degree-of-freedom,

EQ.2: y-translational degree-of-freedom,

EQ.3: z-translational degree-of-freedom,

EQ.4: translational motion in direction given by VID,

EQ.5: normal direction of the element or segment.

LCID1

Load curve ID to describe the amplitude (or real part) of velocity, see *DEFINE_CURVE.

LCID2

Load curve ID to describe the phase (or imaginary part) of velocity, see *DEFINE_CURVE.

SF

Load curve scale factor.

VID

Vector ID for DOF values of 4.

NID

Node ID, or node set ID, or segment set ID for acoustic result output.

NTYP

Set type:

EQ.0: Node, see *NODE.

EQ.1: Node set, see *SET_NODE.

IPFILE

Flag for output files (default = 0):

EQ.0: Press_Pa (magnitude of pressure vs. frequency), Press_dB (sound pressure level vs. frequency) are provided.

EQ.1: Press_Pa_real (real part of pressure vs. frequency) and Press_Pa_imag (imaginary part of pressure vs. frequency) are provided, in addition to Press_Pa, Press_dB.

VARIABLE	DESCRIPTION
DBA	Flag for writing out weighted SPL files with different weighting options. EQ.0: No writing out weighted SPL files. EQ.1: write out Press_dB(A) by using A-weighting. EQ.2: write out Press_dB(B) by using B-weighting. EQ.3: write out Press_dB(C) by using C-weighting. EQ.4: write out Press_dB(D) by using D-weighting.

Remarks:

1. This command solves the interior acoustic problems which is governed by Helmholtz equation $\nabla^2 p + k^2 p = 0$ with the boundary condition $\partial p / \partial n = -i\omega\rho v_n$, where, p is the acoustic pressure; $k = \omega / c$ is the wave number; ω is the round frequency; c is the acoustic wave speed (sound speed); $i = \sqrt{-1}$ is the imaginary unit; ρ is the mass density and v_n is the normal velocity. This command solves the acoustic problem in frequency domain.

2. If mass density RO is not given, the mass density of PID (the part which defines the acoustic domain), will be used

3. PREF is the reference pressure to convert the acoustic pressure to dB $L_p = 10\,\log_{10}(p^2 / p_{\text{ref}}^2)$ Note that generally $p_{\text{ref}} = 20\mu\text{Pa}$ for air.

4. If the boundary velocity is obtained from steady state dynamics (VAD = 0) using the keyword *FREQUENCY_DOMAIN_SSD, the part (PID) which defines the acoustic domain has to use one of the following material models,

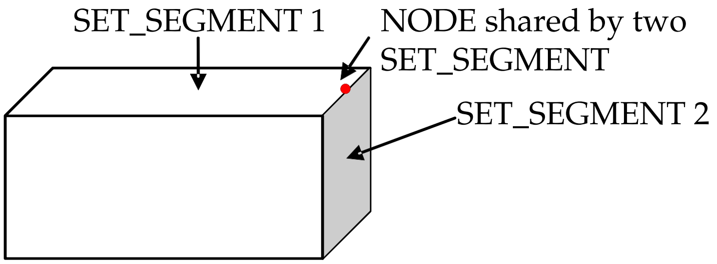
   a) MAT_ELASTIC_FLUID

   b) MAT_NULL (and EOS_IDEAL_GAS)

   Since only the above material models enable implicit eigenvalue analysis. If the boundary excitation is given by load curves LCID1 and LCID2 (VAD > 0), the part (PID) which defines the acoustic domain can use any material model which is compatible with 8-node solid elements, as only the mesh of the PID will be utilized in the computation. For example, MAT_ACOUSTIC and MAT_ELASTIC_FLUID can be used.

5. If VAD = 0, the boundary excitation is given as velocity obtained from steady state dynamics. The other parameters in Card 3 (DOF, LCID1, LCID2, SF and VID) are ignored.

6. If a node's vibration boundary condition is defined multiple times, only the last definition is considered. This happens usually when a node is on edge and shared by two or more PART, SET_PART, SET_NODE, or SET_SEGMENT and different vibration condition is defined on each of the SET_NODE or SET_SEGMENT.



7. Results including acoustic pressure and SPL are given in d3acs binary files, which can be accessed by LS-PrePost. Nodal pressure and SPL values for nodes specified by NID and NTYP are given in ASCII file Press_Pa and Press_dB, which can be accessed by LS-PrePost. Press_Pa gives magnitude of the pressure. Press_dB gives Sound Pressure Level in terms of dB.

8. If the boundary velocity condition is given by Steady State Dynamics (VAD = 0), the range and number of frequencies (FMIN, FMAX and NFREQ) should be compatible with the corresponding parameters in Card 1 of the keyword *FREQUENCY_DOMAIN_SSD

9. For acoustic eigenvalue analysis (OPTION = EIGENVALUE), Card 4 is optional and Card 5 is not used.

FREQUENCY_DOMAIN_ACOUSTIC_FRINGE_PLOT_*{OPTION}

Available options include:

PART

PART_SET

NODE_SET

SPHERE

PLATE

Purpose: Define field points for acoustic pressure computation by BEM acoustic solver, and save the results to D3ACS binary database.

Card 1 for option PART, PART_SET or NODE_SET.

Card 1	1	2	3	4	5	6	7	8
Variable	PID/SID							
Type	I							
Default	none							

Card 1 for option SPHERE.

Card 1	1	2	3	4	5	6	7	8
Variable	CENTER	R	DENSITY	X	Y	Z		
Type	I	F	I	F	F	F		
Default	1	none	none	none	none	none		

Card 1 for option PLATE.

Card 1	1	2	3	4	5	6	7	8
Variable	NORM	LEN_X	LEN_Y	X	Y	Z	NELM_X	NELM_Y
Type	I	F	F	F	F	F	I	I
Default	1	none	none	none	none	none	10	10

VARIABLE **DESCRIPTION**

PID/SID Part ID or part set ID or node set ID.

CENTER Flag for defining the center point for the sphere.

EQ.1: mass center of the original structure.

EQ.2: geometry center of the original structure.

EQ.3: defined by (x, y, z).

R Radius of the sphere.

DENSITY Parameter to define how coarse or dense the created sphere mesh is. It is a number between 3 and 39, where “3” gives you 24 elements while “39” gives you 8664 elements.

X x-coordinate of the center.

Y y-coordinate of the center.

Z z-coordinate of the center.

NORM Norm direction of the plate.

EQ.1: x-direction

EQ.2: y-direction

EQ.3: z-direction

LEN_X Length of longer side of the plate.

LEN_Y Length of shorter side of the plate.

NELM_X Number of elements on longer side of the plate.

NELM_Y Number of elements on shorter side of the plate.

Remarks:

1. This command defines field points where the acoustic pressure will be computed by *FREQUENCY_DOMAIN_ACOUSTIC_BEM. The field points can be defined as existing structure components if option PART, PART_SET or NODE_SET is used. The field points can be created by LS-DYNA if option SPHERE or PLATE is used.

2. The acoustic pressure results at those field points are saved in D3ACS binary database and are accessible by LS-PrePost. With FCOMP tool in LS-PrePost, the fringe plot of the results (real part acoustic pressure, imaginary part acoustic pressure, magnitude of acoustic pressure and Sound Pressure Level) can be generated.

3. The field points defined by this keyword are separate from the field points defined in Card 2 of *FREQUENCY_DOMAIN_ACOUSTIC_BEM. The acoustic pressure results for the latter are only saved in ASCII database Press_Pa and Press_dB, etc. (in a tabular format that can be plotted in LS-PrePost by using the XYPlot tool).

*FREQUENCY_DOMAIN_ACOUSTIC_INCIDENT_WAVE

Purpose: Define incident sound wave for acoustic scattering problems.

Wave Definition Cards. This card may be repeated to define multiple incident waves. Input stops when the next "*" Keyword is found.

Card 1	1	2	3	4	5	6	7	8
Variable	TYPE	MAG	XC	YC	ZC			
Type	I	F	F	F	F			
Default	1	none	none	none	none			

VARIABLE	DESCRIPTION
TYPE	Type of incident sound wave: EQ.1: plane wave. EQ.2: spherical wave.
MAG	Magnitude of the incident sound wave. GT.0: constant magnitude. LT.0: |MAG| is a curve ID, which defines the frequency dependent magnitude. See *DEFINE_CURVE.
XC, YC, ZC	Direction cosines for the place wave (TYPE = 1), or coordinates of the point source for the spherical wave (TYPE = 2).

Remarks:

1. For plane wave, the incident wave is defined as

$$p(x,y,z) = Ae^{-ik(\alpha x+\beta y+\gamma z)}$$

where, A is the magnitude of the incident wave and α, β and γ are the direction cosines along the incident direction. $i = \sqrt{-1}$ is the imaginary unit and $k = \omega/c$ is the wave number. ω is the round frequency and c is the sound speed.

2. For spherical wave, the incident wave is defined as

$$p(r) = A\frac{e^{-ikr}}{r}$$

where, A is the magnitude of the incident wave and r is the distance measured from the position of the point source.

*FREQUENCY_DOMAIN_ACOUSTIC_SOUND_SPEED

Purpose: Define frequency dependent complex sound speed to be used in frequency domain finite element method or boundary element method acoustic analysis.

Card 1	1	2	3	4	5	6	7	8
Variable	ID							
Type	I							
Default	none							

Card 2	1	2	3	4	5	6	7	8
Variable	LCID1	LCID2						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
ID	Complex sound speed ID.
LCID1	Curve ID for real part of frequency dependent complex sound speed.
LCID2	Curve ID for imaginary part of frequency dependent complex sound speed.

Remarks:

1. The sound speed in an acoustic medium is usually defined as a constant real value. But it can also be defined as a complex value which is dependent on frequency, to introduce damping in the system.

2. To use the frequency dependent complex sound speed defined here, set the sound speed C = -ID in *FREQUENCY_DOMAIN_ACOUSTIC_FEM, or *FREQUENCY_DOMAIN_ACOUSTIC_BEM keywords.

*FREQUENCY_DOMAIN_FRF

Purpose: This keyword computes frequency response functions due to nodal excitations.

> NOTE: Natural frequencies and mode shapes are needed for computing the frequency response functions. Thus, keyword *CONTROL_IMPLICIT_EIGENVALUE *must* be included in input. See Remark 1.

Card 1	1	2	3	4	5	6	7	8
Variable	N1	N1TYP	DOF1	VAD1	VID1	FNMAX	MDMIN	MDMAX
Type	I	I	I	I	I	F	I	I
Default	none	0	none	3	0	0.0	0	0

Card 2	1	2	3	4	5	6	7	8
Variable	DAMPF	LCDAM	LCTYP	DMPMAS	DMPSTF			
Type	F	I	I	F	F			
Default	0.0	0	0	0.0	0.0			

Card 3	1	2	3	4	5	6	7	8
Variable	N2	N2TYP	DOF2	VAD2	VID2	RELATV		
Type	I	I	I	I	I	I		
Default	none	0	none	2	0	0		

Card 4	1	2	3	4	5	6	7	8
Variable	FMIN	FMAX	NFREQ	FSPACE	LCFREQ	RESTRT	OUTPUT	
Type	F	F	I	I	I	I	I	
Default	none	none	2	0	none	0	0	

VARIABLE	DESCRIPTION

N1

Node / Node set/Segment set ID for excitation input. When VAD1, the excitation type, is set to 1, which is acceleration, this field is ignored.

N1TYP

Type of N1:

EQ.0: node ID,

EQ.1: node set ID,

EQ.2: segment set ID.

When VAD1, the excitation type, is set to 1, which is acceleration, this field is ignored.

DOF1

Applicable degrees-of-freedom for excitation input (ignored if VAD1 = 4):

EQ.0: translational movement in direction given by vector VID1,

EQ.1: x-translational degree-of-freedom, or x-rotational degree-of-freedom (for torque excitation, VAD1 = 8)

EQ.2: y-translational degree-of-freedom, or y-rotational degree-of-freedom (for torque excitation, VAD1 = 8),

EQ.3: z-translational degree-of-freedom, or z-rotational degree-of-freedom (for torque excitation, VAD1 = 8).

VARIABLE	DESCRIPTION
VAD1	Excitation input type: EQ.0: base velocity, EQ.1: base acceleration, EQ.2: base displacement, EQ.3: nodal force, EQ.4: pressure, EQ.5: enforced velocity by large mass method, EQ.6: enforced acceleration by large mass method, EQ.7: enforced displacement by large mass method. EQ.8: torque, EQ.9: base angular velocity, EQ.10: base angular acceleration, EQ.11: base angular displacement
VID1	Vector ID for DOF1 = 0 for excitation input, see *DEFINE_VECTOR.
FNMAX	Optional maximum natural frequency employed in FRF computation. See Remark 3.
MDMIN	The first mode employed in FRF computation (optional). See Remarks 3 and 4.
MDMAX	The last mode employed in FRF computation (optional). It should be set as a positive integer in a restart run (RESTRT = 1 or 3) based on the number of eigenmodes available in the existing d3eigv database. See Remarks 3 and 4.
DAMPF	Modal damping coefficient, ζ. See Remark 5.
LCDAM	Load Curve ID defining mode dependent modal damping coefficient, ζ. See Remark 5.
LCTYP	Type of load curve defining modal damping coefficient: EQ.0: Abscissa value defines frequency, EQ.1: Abscissa value defines mode number. See Remark 5.

VARIABLE	DESCRIPTION
DMPMAS	Mass proportional damping constant, α, in Rayleigh damping. See Remark 5.
DMPSTF	Stiffness proportional damping constant, β, in Rayleigh damping. See Remark 5.
N2	Node / Node set/Segment set ID for response output.
N2TYP	Type of N2: EQ.0: node ID, EQ.1: node set ID, EQ.2: segment set ID.
DOF2	Applicable degrees-of-freedom for response output: EQ.0: direction given by vector VID2, EQ.1: x-translational degree-of-freedom, EQ.2: y-translational degree-of-freedom, EQ.3: z-translational degree-of-freedom, EQ.4: x-rotational degree-of-freedom, EQ.5: y-rotational degree-of-freedom, EQ.6: z-rotational degree-of-freedom, EQ.7: x, y and z-translational degrees-of-freedom, EQ.8: x, y and z-rotational degrees-of-freedom.
VAD2	Response output type: EQ.0: velocity, EQ.1: acceleration, EQ.2: displacement, EQ.3: nodal force (see Remark 8).
VID2	Vector ID for DOF2 = 0 for response direction, see *DEFINE_-VECTOR.
RELATV	FLAG for displacement, velocity and acceleration results: EQ.0: absolute values are requested, EQ.1: relative values are requested (for VAD1 = 0, 1, 2 only).

VARIABLE	DESCRIPTION
FMIN	Minimum frequency for FRF output (cycles/time). See Remark 6.
FMAX	Maximum frequency for FRF output (cycles/time). See Remark 6.
NFREQ	Number of frequencies for FRF output. See Remark 6.
FSPACE	Frequency spacing option for FRF output: EQ.0: linear, EQ.1: logarithmic, EQ.2: biased. See Remark 6.
LCFREQ	Load Curve ID defining the frequencies for FRF output. See Remark 6.
RESTRT	Restart option: EQ.0: initial run, EQ.1: restart with d3eigv family files, EQ.2: restart with dumpfrf, EQ.3: restart with d3eigv family files and dumpfrf. See Remark 7.
OUTPUT	Output option: EQ.0: write amplitude and phase angle pairs, EQ.1: write real and imaginary pairs.

Remarks:

1. **Frequency Response Functions.** The FRF (frequency response functions) can be given as Displacement/Force (called Admittance, Compliance, or Receptance), Velocity/Force (called Mobility), Acceleration/Force (called Accelerance, Inertance), etc.

2. **Enforced Motion.** The excitation input can be given as enforced motion (VAD1 = 5, 6, 7). Large mass method is used for this type of excitation input. The user need to attach a large mass to the nodes where the enforced motion is applied by using the keyword *ELEMENT_MASS_*{OPTION}*, and report the large mass per node (MPN) in the keyword

*CONTROL_FREQUENCY_DOMAIN. For more details, please refer to *CONTROL_FREQUENCY_DOMAIN.

3. **Maximum Frequency.** FNMAX decides how many natural vibration modes are adopted in FRF computation. LS-DYNA uses only modes with lower or equal frequency than FNMAX in FRF computation. If FNMAX is not given, the number of modes in FRF computation is same as the number of modes, NEIG, from the *CONTROL_IMPLICIT_EIGENVALUE keyword card, unless MDMIN and MDMAX are prescribed (see remark 5).

4. **Maximum/Minimum Mode.** MDMIN and MDMAX decide which mode(s) are adopted in FRF computation. This option is useful for calculating the contribution from a single mode (MDMIN = MDMAX) or several modes (MDMIN < MDMAX). If only MDMIN is given, LS-DYNA uses the single mode (MDMIN) to compute FRF. In a restart run based on existing eigenmode database d3eigv (RESTRT = 1 or 3), MDMAX should be a positive integer which is equal to or less than the number of eigenmodes available in d3eigv.

5. **Damping.** Damping can be prescribed in several ways:

   a) To use a constant modal damping coefficient ζ for all the modes, define DAMPF only. LCDMP, LCTYP, DMPMAS and DMPSTF are ignored.

   b) To use mode dependent modal damping, define a load curve (*DEFINE_CURVE) and specify that if the abscissa value defines the frequency or mode number by LCTYP. DMPMAS and DMPSTF are ignored.

   c) To use Rayleigh damping, define DMPMAS (α) and DMPSTF (β) and keep DAMPF as 0.0, and keep LCDMP, LCTYP as 0. The damping matrix in Rayleigh damping is defined as $\mathbf{C} = \alpha\mathbf{M} + \beta\mathbf{K}$, where, **C**, **M** and **K** are the damping, mass and stiffness matrices respectively.

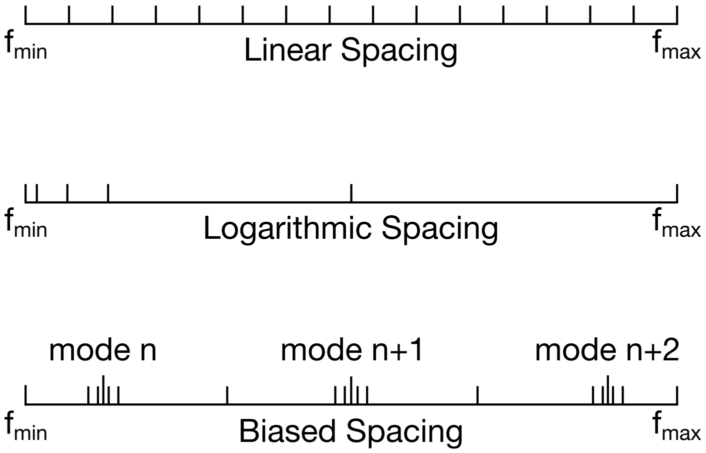


Figure 20-2. Spacing options of the frequency points.

6. **Frequency Points.** There are two methods to define the frequencies.

   a) The first method is to define FMIN, FMAX, NFREQ and FSPACE. FMIN and FMAX specify the frequency range of interest and NFREQ specifies the number of frequencies at which results are required. FSPACE specifies the type of frequency spacing (linear, logarithmic or biased) to be used.

   These frequency points for which results are required can be spaced equally along the frequency axis (on a linear or logarithmic scale). Or they can be biased toward the eigenfrequencies (the frequency points are placed closer together at eigenfrequencies in the frequency range) so that the detailed definition of the response close to resonance frequencies can be obtained. See Figure 20-2.

   b) The second method is to use a load curve (LCFREQ) to define the frequencies of interest.

7. **RESTRT Field.** To save time in subsequent runs, the modal analysis stored in the d3eigv file during the first run can be reused by setting RESTRT=1.

   RESTRT = 2 or 3 is used when user wants to add extra vibration modes to FRF computation. After initial FRF computation, user may find that the number of vibration modes is not enough. For example, in the initial computation, user may use only vibration modes up to 500 Hz. Later it is found that vibration modes at higher frequencies are needed. Then it would be more efficient to just compute the extra modes (frequencies above 500 Hz), and add the contribution from these extra modes to the previous FRF results.

   In this case, user may use the option RESTRT = 2 or 3. For RESTRT = 2, LS-DYNA runs a new modal analysis, reads in the previous FRF results (stored in the binary dump file dumpfrf), and add the contribution from the new modes. For RESTRT = 3, LS-DYNA reads in d3eigv family files generated elsewhere and reads in also dumpfrf, and add the contribution from the new modes.

8. **Nodal Force Response Output.** For nodal force response (VAD2=3), the same nodes or node set need to be defined in *DATABASE_NODAL_FORCE_GROUP. In addition the MSTRES field for the *CONTROL_IMPLICIT_EIGENVALUE keyword must be set to 1.

*FREQUENCY_DOMAIN_MODE_*{OPTION1}*_*{OPTION2}*

Available options for *OPTION1* include:

LIST

GENERATE

SET

For *OPTION2* the available option is:

EXCLUDE

Purpose: Define vibration modes to be used in modal superposition, modal acceleration or modal combination procedures for mode-based frequency domain analysis (such as frequency response functions, steady state dynamics, random vibration analysis and response spectrum analysis). When the option2 EXCLUDE is used, the modes defined in this keyword are excluded from participating in the modal superposition, modal acceleration or modal combination procedures.

Mode ID Cards. For LIST keyword option list the mode IDs. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	MID1	MID2	MID3	MID4	MID5	MID6	MID7	MID8
Type	I	I	I	I	I	I	I	I

Mode Block Cards. For GENERATE keyword option specify ranges of modes. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	M1BEG	M1END	M2BEG	M2END	M3BEG	M3END	M4BEG	M4END
Type	I	I	I	I	I	I	I	I

Mode Set Card. For SET keyword option specify a mode set. Include only one card.

Card 1	1	2	3	4	5	6	7	8
Variable	SID							
Type	I							

VARIABLE	DESCRIPTION
MIDn	Mode ID n.
MnBEG	First mode ID in block n.
MnEND	Last mode ID in block n. All mode ID's between and including MnBEG and MnEND are added to the list.
SID	Mode set identification (see *SET_MODE).

Remarks:

1. User may use this keyword if some of the vibration modes have less contribution to the total structural response and can be removed from the modal superposition, modal acceleration or modal combination procedures in the mode-based frequency domain analysis.

2. The mode list defined by this keyword overrides the modes specified by MDMIN, MDMAX (or FNMIX, FNMAX) in the keywords *FREQUENCY_DOMAIN_FRF, *FREQUENCY_DOMAIN_SSD, etc.

FREQUENCY_DOMAIN_PATH_*{OPTION}

Available options include:

<BLANK>

PARTITION

Purpose: Specify the path and file name of binary databases (e.g. d3eigv) containing mode information for restarting frequency domain analyses such as FRF, SSD, Random vibration, and Response spectrum analysis.

The PARTITION option supports assigning different binary databases to different frequency ranges. Specifically, each frequency range can be associated with different eigenmodes and modal shape vectors provided by the binary database. This option provides a model for materials that have frequency-dependent properties.

Partition Cards. Card 1 for the PARTITION keyword option. Include one card for each frequency range. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	FBEG	FEND	FILENAME					
Type	F	F	C					
Default	none	none	none					

Filename Card. Card 1 format used with the keyword option left blank.

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	C							
Default	none							

VARIABLE	DESCRIPTION
FBEG	Beginning frequency for using this database
FEND	Ending frequency for using this database

FILENAME	Path and name of the database which contains modal information.

Remarks:

1. If the binary database files are in the runtime directory, this card is not needed for the case *without* partitioning.

2. When the option PARTITION is active, the binary database designated by FILENAME is used for the frequency range starting from (and including) FBEG and ending at (not including) FEND.

*FREQUENCY_DOMAIN_RANDOM_VIBRATION

Available options include:

<BLANK>

FATIGUE

Purpose: Set random vibration control options. When FATIGUE option is used, compute fatigue life of structures or parts under random vibration.

Card 1	1	2	3	4	5	6	7	8
Variable	MDMIN	MDMAX	FNMIN	FNMAX	RESTRT		RESTRM	
Type	I	I	F	F	I		I	
Default	1		0.0		0		0	

Card 2	1	2	3	4	5	6	7	8
Variable	DAMPF	LCDAM	LCTYP	DMPMAS	DMPSTF	DMPTYP		
Type	F	I	I	F	F	I		
Default	0.0	0	0	0.0	0.0	0		

Card 3	1	2	3	4	5	6	7	8
Variable	VAFLAG	METHOD	UNIT	UMLT	VAPSD	VARMS	NAPSD	NCPSD
Type	I	I	I	F	I	I	I	I
Default	none	0					1	0

Card 4	1	2	3	4	5	6	7	8
Variable	LDTYP	IPANELU	IPANELV	TEMPER		LDFLAG		
Type	I	I	I	F		I		
Default				0.0		0		

Auto PSD Cards. Include NAPSD cards of this format, one per excitation.

Card 5a	1	2	3	4	5	6	7	8
Variable	SID	STYPE	DOF	LDPSD	LDVEL	LDFLW	LDSPN	CID
Type	I	I	I	I	I	I	I	I
Default					0	0	0	0

Cross PSD Cards. Include NCPSD cards of this format, one per excitation.

Card 5b	1	2	3	4	5	6	7	8
Variable	LOAD_I	LOAD_J	LCTYP2	LDPSD1	LDPSD2			
Type	I	I	I	I	I			
Default			0					

Fatigue Card. Additional card for FATIGUE keyword option.

Card 6	1	2	3	4	5	6	7	8
Variable	MFTG	NFTG	SNTYPE	TEXPOS	STRSF	INFTG		
Type	I	I	I	F	F	I		
Default	0	1	0	0.0	1.0	0		

S-N Curve Cards. NFTG additional cards for FATIGUE keyword option. Each Card 7 defines one zone for fatigue analysis and the corresponding S-N fatigue curve for that zone.

Card 7	1	2	3	4	5	6	7	8
Variable	PID	LCID	PTYPE	LTYPE	A	B	STHRES	SNLIMT
Type	I	I	I	I	F	F	F	I
Default			0	0			0.	0

Initial Damage Card. INFTG additional cards for FATIGUE keyword option when INFTG > 0.

Card 8	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	C							
Default	d3ftg							

VARIABLE	DESCRIPTION
MDMIN	The first mode in modal superposition method (optional).
MDMAX	The last mode in modal superposition method (optional).

VARIABLE	DESCRIPTION
FNMIN	The minimum natural frequency in modal superposition Method (optional).
FNMAX	The maximum natural frequency in modal superposition method (optional).
RESTRT	Restart option. EQ.0: A new modal analysis is performed, EQ.1: Restart with d3eigv.
RESTRM	Restart option when different types of loads are present. EQ.0: don't read the dump file for PSD and RMS, EQ.1: read in PSD and RMS values from the dump file and add them to the values computed in the current load case.
DAMPF	Modal damping coefficient, ζ.
LCDAM	Load Curve ID defining mode dependent modal damping coefficient ζ.
LCTYP	Type of load curve defining modal damping coefficient EQ.0: Abscissa value defines frequency, EQ.1: Abscissa value defines mode number.
DMPMAS	Mass proportional damping constant α, in Rayleigh damping.
DMPSTF	Stiffness proportional damping constant β, in Rayleigh damping.
DMPTYP	Type of damping EQ.0: modal damping. EQ.1: broadband damping.

VARIABLE	DESCRIPTION
VAFLAG	Loading type: EQ.0: No random vibration analysis. EQ.1: Base acceleration. EQ.2: Random pressure. EQ.3: Plane wave. EQ.4: Shock wave. EQ.5: Progressive wave. EQ.6: Reverberant wave. EQ.7: Turbulent boundary layer wave. EQ.8: Nodal force.
METHOD	Method for modal response analysis. EQ.0: method set automatically by LS-DYNA (recommended) EQ.1: modal superposition method EQ.2: modal acceleration method EQ.3: modal truncation augmentation method
UNIT	Flag for acceleration unit conversion: EQ.0: use [length unit]/[time unit]^2 as unit of acceleration. EQ.1: use *g* as unit for acceleration, and SI units (Newton, kg, meter, second, etc.) elsewhere. EQ.2: use *g* as unit for acceleration, and Engineering units (lbf, lbf × second^2/inch, inch, second, etc.) elsewhere. EQ.3: use *g* as unit for acceleration, and units (kN, kg, mm, ms, GPa, etc.) elsewhere. EQ.-1: use *g* as unit for acceleration and provide the multiplier for converting g to [length unit]/[time unit]^2.
UMLT	Multiplier for converting g to [length unit]/[time unit]^2 (used only for UNIT = -1).

VARIABLE DESCRIPTION

VAPSD

Flag for PSD output:

EQ.0: Absolute PSD output is requested.

EQ.1: Relative PSD output is requested (used only for VAFLAG = 1)

VARMS

Flag for RMS output:

EQ.0: Absolute RMS output is requested.

EQ.1: Relative RMS output is requested (used only for VAFLAG = 1)

NAPSD

Number of auto PSD load definition. Card 5a is repeated "NAPSD" times, one for each auto PSD load definition. The default value is 1.

NCPSD

Number of cross PSD load definition. Card 5b is repeated "NCPSD" times, one for each cross PSD load definition. The default value is 0.

LDTYP

Excitation load (LDPSD in card 5) type:

EQ.0: PSD.

EQ.1: SPL (for plane wave only).

EQ.2: time history load.

IPANELU

Number of strips in U direction (used only for VAFLAG = 5, 6, 7)

IPANELV

Number of strips in V direction (used only for VAFLAG = 5, 6, 7)

TEMPER

Temperature

LDFLAG

Type of loading curves.

EQ.0: Log-Log interpolation (default)

EQ.1: Semi-Log interpolation

EQ.2: Linear-Linear interpolation

VARIABLE **DESCRIPTION**

SID

GE.0: Set ID for the panel exposed to acoustic environment, or the nodes subjected to nodal force excitation, or nodal acceleration excitation. For VAFLAG = 1, base acceleration, leave this as blank

LT.0: used to define the cross-PSD. |SID| is the ID of the load cases.

STYPE

Flag specifying meaning of SID.

EQ.0: Node

EQ.1: Node Set

EQ.2: Segment Set

EQ.3: Part

EQ.4: Part Set

LT.0: used to define the cross-psd. |STYPE| is the ID of the load cases.

DOF

Applicable degrees-of-freedom for nodal force excitation or base acceleration (DOF = 1, 2, and 3), or wave direction:

EQ.0: translational movement in direction given by vector VID.

EQ.±1: x-translational degree-of-freedom (positive or negative)

EQ.±2: y-translational degree-of-freedom (positive or negative)

EQ.±3: z-translational degree-of-freedom (positive or negative)

LDPSD

Load curve for PSD, SPL, or time history excitation.

LDVEL

Load curve for phase velocity.

LDFLW

Load curve for exponential decay for TBL in flow-wise direction

LDSPN

Load curve for exponential decay for TBL in span-wise direction

CID/VID

Coordinate system ID for defining wave direction, see *DEFINE_COORDINATE_SYSTEM; or Vector ID for defining load direction for nodal force, or base excitation, see *DEFINE_VECTOR.

LOAD_I

ID of load i for cross PSD.

LOAD_J

ID of load j for cross PSD.

VARIABLE	DESCRIPTION
LCTYP2	Type of load curves (LDPSD1 and LDPSD2) for defining cross PSD: EQ.0: LDPSD1 defines real part and LDPSD2 defines imaginary part EQ.1: LDPSD1 defines magnitude and LDPSD2 defines phase angle
LDPSD1	Load curve for real part or magnitude of cross PSD
LDPSD2	Load curve for imaginary part or phase angle of cross PSD
MFTG	Method for random fatigue analysis (for option_FATIGUE). EQ.0: no fatigue analysis, EQ.1: Steinberg’s three-band method, EQ.2: Dirlik method, EQ.3: Narrow band method, EQ.4: Wirsching method, EQ.5: Chaudhury and Dover method, EQ.6: Tunna method, EQ.7: Hancock method.
NFTG	Field specifying the number of S - N curves to be defined. GE.0: Number of S - N curves defined by card 6. Card 6 is repeated “NFTG” number of times, one for each S - N fatigue curve definition. The default value is 1. EQ.-999: S - N curves are defined through *MAT_ADD_FATIGUE. If the option FATIGUE is not used, ignore this parameter.
SNTYPE	Stress type of S - N curve in fatigue analysis. EQ.0: von-mises stress EQ.1: maximum principal stress (not implemented) EQ.2: maximum shear stress (not implemented) EQ.-n: The n^{th} stress component.
TEXPOS	Exposure time (used if option FATIGUE is used)

VARIABLE	DESCRIPTION
STRSF	Stress scale factor to accommodate different ordinates in S - N curve. EQ.1: used if the ordinate in S - N curve is stress range (default) EQ.2: used if the ordinate in S - N curve is stress amplitude
INFTG	Flag for including initial damage ratio. EQ.0: no initial damage ratio, GT.0: read existing d3ftg files to get initial damage ratio. When INFTG > 1, it means that the initial damage ratio comes from multiple loading cases (correspondingly, multiple binary databases, defined by Card 7). The value of INFTG should be ≤ 10.
PID	Part ID, or Part Set ID, or Element (solid, shell, beam, thick shell) Set ID.
LCID	S - N fatigue curve ID for the current Part or Part Set. GT.0: S - N fatigue curve ID EQ.-1: S - N fatigue curve uses equation $NS^b = a$ EQ.-2: S - N fatigue curve uses equation $\log(S) = a - b\log(N)$ EQ.-3: S - N fatigue curve uses equation $S = a\,N^b$
PTYPE	Type of PID. EQ.0: Part (default) EQ.1: Part Set EQ.2: SET_SOLID EQ.3: SET_BEAM EQ.4: SET_SHELL EQ.5: SET_TSHELL
LTYPE	Type of LCID. EQ.0: Semi-log interpolation (default) EQ.1: Log-Log interpolation EQ.2: Linear-Linear interpolation

VARIABLE **DESCRIPTION**

A Material parameter *a* in S - N fatigue equation.

B Material parameter *b* in S - N fatigue equation.

STHRES Fatigue threshold (applicable only if LCID < 0).

SNLIMT If LCID > 0

Flag setting algorithm used when stress is lower than the lowest stress on S - N curve (if LCID > 0), or lower than STHRES (if LCID < 0).

EQ.0: use the life at the last point on S - N curve

EQ.1: extrapolation from the last two points on S - N curve (only applicable if LCID > 0)

EQ.2: infinity.

If LCID < 0

Flag setting algorithm used when stress is lower STHRES

EQ.0: use the life at STHRES

EQ.1: *Ingnored.* only applicable for LCID > 0

EQ.2: infinity.

FILENAME Path and name of existing binary database for fatigue information.

Remarks:

1. **Historical Background.** This command evaluates the structural random vibration response due to aero acoustic loads, base excitation, or nodal force. This capability originated in Boeing's in-house code N-FEARA, which is a NIKE3D-based Finite Element tool for performing structural analysis with vibro-acoustic loads. The main developer of N-FEARA is Mostafa Rassaian from the Boeing Company.

2. **Fatigue.** To run this option, it is required that MSTRES = 1 in the keyword *CONTROL_IMPLICIT_EIGENVALUE. This is because the fatigue analysis is depends on stresses.

3. **IPANEL.** The number of strips the in U and V direction are used to group the elements and thereby reduce the number integration domains reducing computational expense. This option is only available for VAFLAG = 5, 6, and 7.

4. **Restarting.** Restart option RESTRT = 1 is used when mode analysis has been done previously. In this case, LS-DYNA skips modal analysis and reads in the d3eigv files from the prior execution. For RESTMD = 1, always use MDMIN = 1 and set MDMAX to the number of modes in the previous run (this can be found in the ASCII file eigout, or it can be extracted from the d3eigv files using LS-PrePost).

5. **Accumulated Fatigue.** The fatigue damage ratio can be accumulated over multiple load cases by setting INFTG = 1. This is useful when a structure is subjected to multiple independent random vibrations. LS-DYNA calculates the total damage ratio by adding the damage ratio from the current calculation to the damage ratio of the previous calculation which are stored in the previous calculation's fatigue database (d3ftg by default). The previous d3ftg file will be overwritten by the new one, if it is in the same directory.

6. **Automatic Method Selection.** If METHOD = 0, LS-DYNA uses modal superposition method for cases (VAFLAG) 4, 5, 6, 7; For cases 1, 2, 3 and 8, LS-DYNA uses modal superposition method when preload condition is present and uses modal acceleration method when preload condition is not present.

7. **Units.** In a set of consistent units, the unit for acceleration is defined as

$$1 \ \text{(acceleration unit)} = \frac{1(\text{length unit})}{[1(\text{time unit})]^2}$$

Some users in industry prefer to use g (acceleration due to gravity) as the unit for acceleration. For example,

$$1\text{g} = 9.81\frac{\text{m}}{\text{s}^2} = 386.089\frac{\text{inch}}{\text{s}^2}$$

If the input and output use g as the unit for acceleration, select UNIT = 1, 2, or 3.

If UNIT = 3, a multiplier (UMLT) for converting g to [length unit]/[time unit]^2 is needed and it is defined by

$$1\text{g} = \text{UMLT} \times \frac{[\text{length unit}]}{[\text{time unit}]^2}$$

For more information about the consistent units, see GS.21 (GETTING STARTED).

8. **Restrictions on Load Curves.** The load curves LDPSD, LDVEL, LDFLW, and LDSPN must all be defined using the same number of points. The number of points in the load curve LDDAMP can be different from those for LDPSD, LDVEL, LDFLW, and LDSPN.

9. **Wave direction.** Wave direction is determined DOF and CID/VID. CID/VID represents a local U-V-W coordinate system for defining acoustic wave direction, only partially correlated waves (VAFLAG=5, 6, 7) need this local coordinate system. For nodal force, base excitation, plane wave or random pressure, CID represents a vector ID defining the load direction (DOF = ±4).

10. **Stress / Strain computation.** To get stress results (PSD and RMS) from random vibration analysis, MSTRES field of the *CONTROL_IMPLICIT_-EIGENVALUE keyword should be set to 1. To get strain results (PSD and RMS) the STRFLG field of *DATABASE_EXTENT_BINARY should be set to 1. To get stress component results for beam elements (which are not based on resultant formulation), the BEAMIP field of *DATABASE_EXTENT_BINARY should be set greater than 0.

11. **Binary plot databases.** PSD and RMS results are given for all nodes and elements. PSD results are written in binary to a file named d3psd. Similarly, RMS results are written in binary to a file named d3rms. See the keyword *DATABASE_FREQUENCY_BINARY_*{OPTION}* for more details.

12. **ASCII Output for Displacement.** Displacement, velocity, and acceleration PSD results are output into ASCII file nodout_psd. The set nodes for which data is written to nodout_psd is specified with the *DATABASE_HISTORY_-NODE keyword.

13. **ASCII Output for Stress.** Stress PSD results are output into ASCII file elout_psd. The set of solid, beam, shell, and thick shell elements to be written to the elout_psd file are specified with the following keywords: *DATABASE_HISTORY_SOLID, *DATABASE_HISTORY_BEAM, *DATABASE_HISTORY_SHELL, *DATABASE_HISTORY_TSHELL.

14. **Cross PSD.** The cross PSD can be defined as complex variables to consider phase difference. In that case, two curves are needed to define the cross PSD (LCPSD1 and LCPSD2). Two load IDs are needed to define the cross PSD (LOAD_I and LOAD_J). They are simply the ordering numbers by which the auto PSDs are defined. For example, the first Card 5a defines load 1 and the second Card 5a defines the load 2. No cross PSD is required if two loads are uncorrelated. Cross PSD for any pair of two correlated loads is defined only once – from lower load ID to higher load ID (e.g. 1->2, 1->3, 2->3, …). The cross PSD from higher load ID to lower load ID (e.g. 2->1, 3->1, 3->2, …) is added by LS-DYNA automatically by using the relationship

$$G_{ji} = \overline{G_{ıȷ}}$$

where G_{ij} is the cross PSD from load i to load j, and $\overline{}$ represents the complex conjugate.

15. **Cross Correlation.** Cross correlation can be defined only for same type of excitations (e.g. nodal force, random pressure). Correlation between different types of excitations is not allowed

16. **Output for Fatigue Data.** When the FATIGUE option is used, a binary plot file, d3ftg, is written. 5 results are included in d3ftg:

    Result 1. Cumulative damage ratio

    Result 2. Expected fatigue life

    Result 3. Zero-crossing frequency

    Result 4. Peak-crossing frequency

    Result 5. Irregularity factor

    These results are given as element variables. Irregularity factor is a real number from 0 to 1. A sine wave has irregularity factor as 1, while white noise has irregularity factor as 0. The lower the irregularity factor, the closer the process is to the broad band case.

17. **Stress Threshold for Fatigue.** In some materials, the S–N curve flattens out eventually, so that below a certain threshold stress STHRES failure does not occur no matter how long the loads are cycled. SNLIMT can be set to 2 in this case; For other materials, such as aluminum, no threshold stress exists and SNLIMT should be set to 0 or 1 for added level of safety.

18. **Restriction on Fatigue Cards.** When the FATIGUE option is used, all fatigue cards (Card 6) *must* be of the same PTYPE (PART or SET of ELEMENTS).

19. **Format for S - N Curves.** S - N curves can be defined by *DEFINE_CURVE, or for LCID<0 by

    $$NS^b = a$$

    when LCID = -1 or for LCID = -2

    $$\log(S) = a - b\log(N)$$

    or for LCID = -3

    $$S = a\,N^b$$

    where N is the number of cycles for fatigue failure and S is the stress amplitude. Please note that the two equations can be converted to each other, with some minor manipulation on the constants a and b.

References:

Mostafa Rassaian, Jung-Chuan Lee, N-FEARA – NIKE3D-based FE tool for structural analysis of vibro-acoustic loads, Boeing report, 9350N-GKY-02-036, December 5, 2003.

*FREQUENCY_DOMAIN_RESPONSE_SPECTRUM

Purpose: perform response spectrum computation to obtain the peak response of a structure.

Card 1	1	2	3	4	5	6	7	8
Variable	MDMIN	MDMAX	FNMIN	FNMAX	RESTRT	MCOMB		
Type	I	I	F	F	I	I		
Default	1		0.0		0	0		

Card 2	1	2	3	4	5	6	7	8
Variable	DAMPF	LCDAMP	LDTYP	DMPMAS	DMPSTF			
Type	F	I	I	F	F			
Default	none	none	0	0.0	0.0			

Card 3 can be repeated if 2 or more input spectra exist (multiple-point response spectrum)

Card 3	1	2	3	4	5	6	7	8
Variable	LCTYP	DOF	LC/TBID	SF	VID	LNID	LNTYP	INFLAG
Type	I	I	I	F	I	I	I	I
Default				1.0				0

VARIABLE	DESCRIPTION
MDMIN	The first mode in modal superposition method (optional).
MDMAX	The last mode in modal superposition method (optional).

FNMIN

The minimum natural frequency in modal superposition method (optional).

FNMAX

The maximum natural frequency in modal superposition method (optional).

RESTRT

Restart option

EQ.0: A new run including modal analysis,

EQ.1: Restart with d3eigv family files created elsewhere.

MCOMB

Method for combination of modes:

EQ.0: SRSS method,

EQ.1: NRC Grouping method,

EQ.2: Complete Quadratic Combination method (CQC),

EQ.3: Double Sum method based on Rosenblueth-Elorduy coefficient,

EQ.4: NRL-SUM method,

EQ.5: Double Sum method based on Gupta-Cordero coefficient,

EQ.6: Double Sum method based on modified Gupta-Cordero coefficient,

EQ.7: Rosenblueth method.

DAMPF

Modal damping ratio, ζ.

LCDAMP

Load Curve ID for defining frequency dependent modal damping ratio ζ.

LDTYP

Type of load curve for LCDAMP

EQ.0: Abscissa value defines frequency,

EQ.1: Abscissa value defines mode number.

DMPMAS

Mass proportional damping constant α, in Rayleigh damping.

DMPSTF

Stiffness proportional damping constant β, in Rayleigh damping.

LCTYP

Load curve type for defining the input spectrum.

EQ.0: base velocity,

EQ.1: base acceleration,

EQ.2: base displacement,

EQ.3: nodal force,

EQ.4: pressure,

EQ.10: base velocity time history,

EQ.11: base acceleration time history,

EQ.12: base displacement time history.

DOF

Applicable degrees-of-freedom for excitation input:

EQ.1: x-translational degree-of-freedom,

EQ.2: y-translational degree-of-freedom,

EQ.3: z-translational degree-of-freedom,

EQ.4: translational movement in direction given by vector VID.

LC/TBID

Load curve or table ID, see *DEFINE_TABLE, defining the response spectrum for frequencies. If the table definition is used a family of curves are defined for discrete critical damping ratios.

SF

Scale factor for the input load spectrum.

VID

Vector ID for DOF values of 4.

LNID

Node ID, or node set ID, or segment set ID where the excitation is applied. If the input load is given as base excitation spectrum, LNID = 0

LNTYP

Set type for LNID:

EQ.1: Node, see *NODE,

EQ.2: Node set, see *SET_NODE,

EQ.3: Segment set, see *SET_SEGMENT,

EQ.4: Part, see *PART,

EQ.5: Part set, see *SET_PART.

INFLAG Frequency interpolation option

EQ.0: Logarithmic interpolation,

EQ.1: Semi-logarithmic interpolation.

EQ.2: Linear interpolation.

Remarks:

1. This command uses modal superposition method to evaluate the maximum response of a structure subjected to input response spectrum load, such as the acceleration spectrum load in earthquake engineering.

2. Modal analysis has to be performed preceding the response spectrum analysis. Thus the keywords *CONTROL_IMPLICIT_GENERAL and *CONTROL_IMPLICIT_EIGENVALUE are expected in the input file.

3. MDMIN, MDMAX, FNMIN and FNMAX should be set appropriately to cover all the natural modes inside the input spectrum.

4. To include stress results, modal stress computation has to be requested in *CONTROL_IMPLICIT_EIGENVALUE (set MSTRES = 1).

5. For base excitation cases, user can choose relative values or absolute values for displacement, velocity and acceleration results output.

6. RESTRT = 1 enables a fast restart run based on d3eigv family files generated in last run or elsewhere. LS-DYNA reads d3eigv family files to get the natural vibration frequencies and mode shapes. If the d3eigv family files are located in a directory other than the working directory, the directory must be specified in *FREQUENCY_DOMAIN_PATH.

7. For Double Sum method (MCOMB = 3), earthquake duration time is given by ENDTIM in the keyword *CONTROL_TERMINATION.

8. Three interpolation options are available for frequency interpolation when reading response spectrum values

   a) When INFLAG = 0 (default), logarithmic interpolation is used, e.g.

   $$\frac{\log y - \log y_1}{\log x - \log x_1} = \frac{\log y_2 - \log y_1}{\log x_2 - \log x_1}$$

   b) When INFLAG = 1, semi-logarithmic interpolation is used, e.g.

   $$\frac{\log y - \log y_1}{x - x_1} = \frac{\log y_2 - \log y_1}{x_2 - x_1}$$

c) When INFLAG = 2, linear interpolation is used, e.g.

$$\frac{y - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1}$$

9. Linear interpolation is used for interpolation with respect to damping ratios.

FREQUENCY_DOMAIN_SSD_*{OPTION}

Available options include:

FATIGUE

ERP

Purpose: Compute steady state dynamic response due to given spectrum of harmonic excitations.

When the FATIGUE option is applied LS-DYNA also calculates the cumulative fatigue damage ratio. When the ERP option is applied LS-DYNA also calculates the Equivalent Radiated Power (ERP) due to vibration.

Card 1	1	2	3	4	5	6	7	8
Variable	MDMIN	MDMAX	FNMIN	FNMAX	RESTMD	RESTDP	LCFLAG	RELATV
Type	I	I	F	F	I	I	I	I
Default	1		0.0		0	0	0	0

Card 2	1	2	3	4	5	6	7	8
Variable	DAMPF	LCDAM	LCTYP	DMPMAS	DMPSTF	DMPFLG		
Type	F	I	I	F	F	I		
Default	0.0	0	0	0.0	0.0	0		

Card 3	1	2	3	4	5	6	7	8
Variable			MEMORY	NERP	STRTYP	NOUT	NOTYP	NOVA
Type			I	I	I	I	I	I
Default			0	0	0	0	0	0

ERP Card. This card is read only when the ERP option is active.

Card 4a	1	2	3	4	5	6	7	8
Variable	R0	C	ERPRLF	ERPREF				
Type	F	F	F	F				
Default	none	none	1.0	0.0				

ERP Part Cards. This card is read NERP times. Since NERP defaults to zero this card is, by default not read, and, furthermore, it is not read *unless* the ERP option is active.

Card 5	1	2	3	4	5	6	7	8
Variable	PID	PTYP						
Type	I	I						
Default	none	0						

Excitation Loads. Repeat Card 4 if multiple excitation loads are present.

Card 4	1	2	3	4	5	6	7	8
Variable	NID	NTYP	DOF	VAD	LC1	LC2	LC3	VID
Type	I	I	I	I	I	I	I	I
Default	none	0	none	none	none	none	0	0

VARIABLE	DESCRIPTION
MDMIN	The first mode in modal superposition method (optional).
MDMAX	The last mode in modal superposition method (optional).
FNMIN	The minimum natural frequency in modal superposition method (optional).

VARIABLE **DESCRIPTION**

FNMAX

The maximum natural frequency in modal superposition method (optional).

RESTMD

Restart option:

EQ.0: A new modal analysis is performed,

EQ.1: Restart with d3eigv.

RESTDP

Restart option:

EQ.0: A new run without dumpssd,

EQ.1: Restart with dumpssd.

LCFLAG

Load Curve definition flag.

EQ.0: load curves are given as amplitude / phase angle,

EQ.1: load curves are given as real / imaginary components.

RELATV

Flag for displacement, velocity and acceleration results:

EQ.0: absolute values are requested,

EQ.1: relative values are requested (for VAD = 2, 3 and 4 only).

DAMPF

Modal damping coefficient, ζ.

LCDAM

Load Curve ID defining mode dependent modal damping coefficient ζ.

LCTYP

Type of load curve defining modal damping coefficient.

EQ.0: Abscissa value defines frequency,

EQ.1: Abscissa value defines mode number.

DMPMAS

Mass proportional damping constant α, in Rayleigh damping.

DMPSTF

Stiffness proportional damping constant β, in Rayleigh damping

DMPFLG

Damping flag:

EQ.0: use modal damping coefficient ζ, defined by DAMPF, or LCDAM, or Rayleigh damping defined by DMPMAS and DMPSTF in this card.

EQ.1: use damping defined by *DAMPING_PART_MASS and *DAMPING_PART_STIFFNESS.

VARIABLE	DESCRIPTION
MEMORY	Memory flag: EQ.0: modal superposition will be performed in-core. This option runs faster. EQ.1: modal superposition will be performed out-of-core. This is needed for some large scale problems that cannot fit in main memory. This method incurs a performance penalty associated with disk speed.
NERP	Number of ERP panels.
STRTYP	Stress used in fatigue analysis: EQ.0: Von Mises stress, EQ.1: Maximum principal stress, EQ.2: Maximum shear stress.
NOUT	Part, part set, segment set, or node set ID for response output (use with acoustic computation). See NOTYP below.
NOTYP	Type of NOUT: EQ.0: part set ID (not implemented), EQ.1: part ID (not implemented), EQ.2: segment set ID, EQ.3: node set ID, EQ.-2: segment set ID which mismatches with acoustic boundary nodes. Mapping of velocity or acceleration to the acoustic boundary nodes is performed.
NOVA	Response output type. EQ.0: velocity, EQ.1: acceleration.
RO	Fluid density.
C	Sound speed of the fluid.
ERPRLF	ERP radiation loss factor.
ERPREF	ERP reference value. This is used to convert the absolute ERP value to ERP in decibels (dB).

VARIABLE	DESCRIPTION
PID	Part, part set, or segment set ID for ERP computation. See PTYP below.
PTYP	Type of PID: EQ.0: part ID, EQ.1: part set ID, EQ.2: segment set ID.
NID	Node, node set, or segment set ID for excitation input. See NTYP below.
NTYP	Type of NID. EQ.0: node ID, EQ.1: node set ID, EQ.2: segment set ID.
DOF	Applicable degrees-of-freedom for excitation input (ignored if VAD = 1). EQ.1: x-translational degree-of-freedom, EQ.2: y-translational degree-of-freedom, EQ.3: z-translational degree-of-freedom, EQ.4: translational movement in direction given by vector VID.
VAD	Excitation input type: EQ.0: nodal force, EQ.1: pressure, EQ.2: base velocity, EQ.3: base acceleration, EQ.4: base displacement, EQ.5: enforced velocity by large mass method (see remark 10), EQ.6: enforced acceleration by large mass method (see remark 10), EQ.7: enforced displacement by large mass method (see remark 10).

VARIABLE	DESCRIPTION
LC1	Load Curve ID defining amplitude (LCFLAG = 0) or real (in-phase) part (LCFLAG = 1) of load as a function of frequency.
LC2	Load Curve ID defining phase angle (LCFLAG = 0) or imaginary (out-phase) part (LCFLAG = 1) of load as a function of frequency.
LC3	Load Curve ID defining load duration for each frequency. This parameter is optional and is only needed for fatigue analysis.
VID	Vector ID for DOF = 4 for excitation input, see *DEFINE_VECTOR.

Remarks:

1. This command computes steady state dynamic response due to harmonic excitation spectrum by modal superposition method.

2. Natural frequencies and mode shapes are needed for running the modal superposition method. Thus, the keyword *CONTROL_IMPLICIT_EIGENVALUE *must* be included in input.

3. MDMIN/MDMAX and FNMIN/FNMAX together determine which modes are used in modal superposition method. The first mode must have a mode number ≥ MDMIN, and frequency ≥ FNMIN; The last mode must have mode number ≤ MDMAX, and frequency ≤ FNMAX. When MDMAX or FNMAX is not given, the last mode in modal superposition method is the last mode available in FILENM.

4. Restart option RESTMD = 1 is used if mode analysis has been done previously. In this case, LS-DYNA skips modal analysis and reads in d3eigv family files generated previously. For RESTMD = 1, always use MDMIN = 1 and MDMAX = number of modes given by modal analysis (can be found from ASCII file eigout, or from d3eigv files using LS-PREPOST).

5. Restart option RESTDP = 1 is used if user wants to add contribution of additional modes to previous SSD results. In this case, LS-DYNA reads in binary dump file dumpssd which contains previous SSD results and adds contribution from new modes. For RESTDP = 1, the new modal analysis (RESTMD = 0) or the d3eigv family files created elsewhere (RESTMD = 1) should exclude the modes used in previous SSD computation. This can be done by setting LFLAG (and RFLAG, if necessary), and setting a nonzero LFTEND (and RHTEND) in *CONTROL_IMPLICIT_EIGENVALUE. The RESTDP option can also be used

if the frequency range for modal analysis is divided into segments and modal analysis is performed for each frequency range separately.

6. Sometimes customers would like to add some acoustic field nodes and run BEM/FEM acoustic computation after SSD. The RESTMD and RESTDP options still work even if the number of nodes may get changed after previous modal analysis, provided that the IDs of the old nodes are not changed.

7. Damping can be prescribed in several ways:

   a) To use a constant modal damping coefficient ζ for all the modes, define DAMPF only. LCDMP, LCTYP, DMPMAS and DMPSTF are ignored.

   b) To use mode dependent modal damping, define a load curve (*DEFINE_-CURVE) and specify that if the abscissa value defines the frequency or mode number by LCTYP. DMPMAS and DMPSTF are ignored.

   c) To use Rayleigh damping, define DMPMAS (α) and DMPSTF (β) and keep DAMPF as 0.0, and keep LCDMP, LCTYP as 0. The damping matrix in Rayleigh damping is defined as $\mathbf{C} = \alpha\mathbf{M} + \beta\mathbf{K}$, where, $\mathbf{C}$, $\mathbf{M}$ and $\mathbf{K}$ are the damping, mass and stiffness matrices respectively.

8. NOUT and NOTYP are used to define the nodes where velocity or acceleration are requested to be written to a binary file "bin_ssd" or other filename defined by "bem=filename" (see keyword *FREQUENCY_DOMAIN_ACOUSTIC_-BEM) in command line. The velocity or acceleration data in this file can be used by BEM or FEM acoustic solver to perform a vibro-acoustic analysis. If structure nodes and acoustic boundary nodes are mismatched, the option NOTYP = -2 can be used. The velocity or acceleration data given at a structure segment set NOUT is mapped to acoustic boundary nodes.

9. For base velocity, base acceleration or base displacement (VAD = 2, 3 or 4) excitations, the parameters NID, NTYP are not used and can be blank. The base velocity, base acceleration and base displacement cases are treated by applying inertia force to the structure.

10. For the cases with enforced motion excitation such as nodal velocity, acceleration, or displacement) the large mass method can be used to compute the SSD results. The excitation input can be given as enforced motion curves (VAD = 5, 6, 7). To use the large mass method, the user need to attach a large mass to the nodes where the enforced motion is applied by using the keyword *ELEMENT_MASS_*{OPTION}*, and report the large mass per node (MPN) in the keyword *CONTROL_FREQUENCY_DOMAIN. For more details, please refer to *CONTROL_FREQUENCY_DOMAIN.

11. Displacement, velocity and acceleration results are output into ASCII file NODOUT_SSD. The nodes to be output to NODOUT_SSD are specified by card *DATABASE_HISTORY_NODE.

12. Stress results are output into ASCII file ELOUT_SSD. The solid, beam, shell and thick shell elements to be output to ELOUT_SSD are specified by the following cards:

    DATABASE_HISTORY_SOLID_*{OPTION}

    DATABASE_HISTORY_BEAM_*{OPTION}

    DATABASE_HISTORY_SHELL_*{OPTION}

    DATABASE_HISTORY_TSHELL_*{OPTION}

13. The phase angle is given in range (-180°, 180°].

14. When the FATIGUE option is present, the cumulative fatigue damage ratio due to the harmonic vibration is computed and saved in binary plot database d3ftg. The *MATERIAL_ADD_FATIGUE keyword is needed to define the S-N fatigue curve for each material.

*HOURGLASS

Purpose: Define hourglass and bulk viscosity properties which are referenced via HGID in the *PART command. Properties specified here, when invoked for a particular part, override those in *CONTROL_HOURGLASS and *CONTROL_BULK_VISCOSITY.

An additional option **TITLE** may be appended to ***HOURGLASS** keywords. If this option is used then an additional line is read for each section in 80a format which can be used to describe the section. At present LS-DYNA does not make use of the title. Inclusion of titles gives greater clarity to input decks.

Card 1	1	2	3	4	5	6	7	8
Variable	HGID	IHQ	QM	IBQ	Q1	Q2	QB, VDC	QW
Type	I/A	I	F	I	F	F	F	F
Default	0		.10		1.5	0.06	QM, 0.	QM
Remark		1,6	2 ,4, 7		3	3	5	5

VARIABLE	DESCRIPTION
HGID	Hourglass ID. A unique number or label must be specified. This ID is referenced by HGID in the *PART command.

VARIABLE **DESCRIPTION**

IHQ

Hourglass control type. For solid elements six options are available. For quadrilateral shell and membrane elements the hourglass control is based on the formulation of Belytschko and Tsay, i.e., options 1-3 are identical, and options 4-5 are identical:

EQ.0: see remark 9,

EQ.1: standard LS-DYNA viscous form,

EQ.2: Flanagan-Belytschko viscous form,

EQ.3: Flanagan-Belytschko viscous form with exact volume integration for solid elements,

EQ.4: Flanagan-Belytschko stiffness form,

EQ.5: Flanagan-Belytschko stiffness form with exact volume integration for solid elements.

EQ.6: Belytschko-Bindeman [1993] assumed strain co-rotational stiffness form for 2D and 3D solid elements only.

EQ.7: Linear total strain form of type 6 hourglass control. (See remark 6 below).

EQ.8: Activates full projection warping stiffness for shell formulations 16 and -16, and is the default for these formulations. A speed penalty of 25% is common for this option.

EQ.9: Puso [2000] enhanced assumed strain stiffness form for 3D hexahedral elements.

EQ.10: Cosserat Point Element (CPE) developed by Jabareen and Rubin [2008] and Jabareen et.al. [2013], see *CONTROL_HOURGLASS.

A discussion of the viscous and stiffness hourglass control for shell elements follows at the end of this section..

VARIABLE **DESCRIPTION**

QM Hourglass coefficient. Values of QM that exceed 0.15 may cause instabilities for brick elements used with forms IHG = 0 to 5 and all the IHG forms applicable to shell elements. The stiffness forms can stiffen the response especially if deformations are large and therefore should be used with care. For the shell and membrane elements QM is taken as the membrane hourglass coefficient, the bending as QB, and warping as QW. These coefficients can be specified independently, but generally, QM = QB = QW, is adequate. For type 6 solid element hourglass control, see remark 4 below. For hourglass type 9, see Remark 8.

IBQ Not used. Bulk viscosity is always on for solids. Bulk viscosity for beams and shells can only be turned on using the variable TYPE in *CONTROL_BULK_VISCOSITY; however, the coefficients can be set using Q1 and Q2 below.

Q1 Quadratic bulk viscosity coefficient.

Q2 Linear bulk viscosity coefficient.

QB Hourglass coefficient for shell bending. The default: QB = QM. (See Remark 4).

VDC Viscous damping coefficient for types 6 and 7 hourglass control.

QW Hourglass coefficient for shell warping. The default: QB = QW.

Remarks:

1. Viscous hourglass control is recommended for problems deforming with high velocities. Stiffness control is often preferable for lower velocities, especially if the number of time steps are large. For solid elements the exact volume integration provides some advantage for highly distorted elements.

2. For automotive crash the stiffness form of the hourglass control with a coefficient of 0.05 is preferred by many users.

3. Bulk viscosity is necessary to propagate shock waves in solid materials. Generally, the default values are okay except in problems where pressures are very high, larger values may be desirable. In low density foams, it may be necessary to reduce the viscosity values since the viscous stress can be significant. It is not advisable to reduce it by more than an order of magnitude.

4. Type 6 hourglass control is for 2D and 3D solid elements only. Based on elastic constants and an assumed strain field, it produces accurate coarse mesh bending results for elastic material when QM = 1.0. For plasticity models with a yield stress tangent modulus that is much smaller than the elastic modulus, a smaller value of QM (0.001 to 0.1) may produce better results. For foam or rubber models, larger values (0.5 to 1.0) may work better. For any material, keep in mind that the stiffness is based on the elastic constants, so if the material softens, a QM value smaller than 1.0 may work better. For anisotropic materials, an average of the elastic constants is used. For fluids modeled with null material, type 6 hourglass control is viscous and is scaled to the viscosity coefficient of the material (see *MAT_NULL).

5. In part, the computational efficiency of the Belytschko-Lin-Tsay and the under integrated Hughes-Liu shell elements are derived from their use of one-point quadrature in the plane of the element. To suppress the hourglass deformation modes that accompany one-point quadrature, hourglass viscous or stiffness based stresses are added to the physical stresses at the local element level. The discussion of the hourglass control that follows pertains to all one point quadrilateral shell and membrane elements in LS-DYNA.

   The hourglass shape vector τ_I is defined as

$$\tau_I = h_I - \left(h_J \hat{x}_{aJ}\right) B_{aI}$$

   where, $\hat{x}_{aJ}$ are the element coordinates in the local system at the Ith element node, B_{aI} is the strain displacement matrix, and hourglass basis vector is:

$$h = \begin{bmatrix} +1 \\ -1 \\ +1 \\ -1 \end{bmatrix}$$

   is the basis vector that generates the deformation mode that is neglected by one-point quadrature. In the above equations and the remainder of this subsection, the Greek subscripts have a range of 2, e.g., $\hat{x}_{aI} = (\hat{x}_{1I}, \hat{x}_{2I}) = \left(\hat{x}_I, \hat{y}_I\right)$.

   The hourglass shape vector then operates on the generalized displacements to produce the generalized hourglass strain rates

$$\dot{q}_\alpha^M = \tau_I \hat{v}_{\alpha I}$$

$$\dot{q}_\alpha^B = \tau_I \hat{\theta}_{\alpha I}$$

$$\dot{q}_3^W = \tau_I \hat{v}_{zI}$$

   where the superscripts M, B, and W denote membrane, bending, and warping modes, respectively. The corresponding hourglass stress rates are then given by

$$\dot{Q}_\alpha^M = \frac{\mathrm{QM} \times EtA}{8} B_{\beta I} B_{\beta I} \dot{q}_\alpha^M$$

$$\dot{Q}_{\alpha}^{B} = \frac{\mathrm{QB} \times Et^{3}A}{192} B_{\beta I} B_{\beta I} \dot{q}_{\alpha}^{B}$$

$$\dot{Q}_{3}^{W} = \frac{\mathrm{QW} \times \kappa G t^{3} A}{12} B_{\beta I} B_{\beta I} \dot{q}_{3}^{B}$$

where t is the shell thickness. The hourglass coefficients: QM, QB, and QW are generally assigned values between 0.05 and 0.10.

Finally, the hourglass stresses which are updated using the time step, Δt, from the stress rates in the usual way, that is,

$$\boldsymbol{Q}^{n+1} = \boldsymbol{Q}^{n} + \Delta t \dot{\mathbf{Q}}$$

and the hourglass resultant forces are then

$$\hat{f}_{\alpha I}^{H} = \tau_I Q_{\alpha}^{M}$$

$$\hat{m}_{\alpha I}^{H} = \tau_I Q_{\alpha}^{B}$$

$$\hat{f}_{3I}^{H} = \tau_I Q_{3}^{W}$$

where the superscript H emphasizes that these are internal force contributions from the hourglass deformations.

6. IHQ = 7 is a linear total strain formulation of the Belytschko-Bindeman [1993] stiffness form for 2D and 3D solid elements. This linear form was developed for visco-elastic material and guarantees that an element will spring back to its initial shape regardless of the severity of deformation.

7. The default value for QM is 0.1 unless superseded by a nonzero value of QH in *CONTROL_HOURGLASS. A nonzero value of QM supersedes QH.

8. Hourglass type 9 is available for hexahedral elements and is based on physical stabilization using an enhanced assumed strain method. In performance it is similar to the Belytschko-Bindeman hourglass formulation (type 6) but gives more accurate results for distorted meshes, e.g., for skewed elements. If QM = 1.0, it produces accurate coarse bending results for elastic materials. The hourglass stiffness is by default based on elastic properties, hence the QM parameter should be reduced to about 0.1 for plastic materials in order not to stiffen the structure during plastic deformation. For materials 3, 18 and 24 there is the option to use a negative value of QM. With this option, the hourglass stiffness is based on the current material properties, i.e., the plastic tangent modulus, and scaled by |QM|.

9. The default value for IHQ, if not defined on *CONTROL_HOUGRGLASS is as follows:

   For shells: viscous type (1 = 2 = 3) for explicit; stiffness type (4=5) for implicit

For solids: type 2 for explicit; type 6 for implicit.

For formulation 1 tshells: type 2.

10. For implicit analysis, hourglass forms 6, 7, 9, and 10 are available for solid elements, and the stiffness form (4 = 5) is available for shells.

11. Tshell formulations 2 and 3 have 2 × 2 in-plane integration and therefore do not use hourglass control.

12. In the case of tshell formulation 1, there are two viscous hourglass types (IHQ = 1,2) and one stiffness type (IHQ > 2).

13. The hourglass type IHQ has no bearing on tshell formulation 5 as this formulation is based on an assumed strain field, similar to formulation 1 solids with hourglass type 6. The hourglass coefficient QM does affect the behavior of tshell formulation 5.

*INCLUDE

Purpose: The keyword *INCLUDE provides a means of reading independent input files containing model data. The file contents are placed directly at the location of the *INCLUDE line.

INCLUDE_*{OPTION}

*INCLUDE_AUTO_OFFSET

INCLUDE_COMPENSATION_*OPTION

*INCLUDE_MULTISCALE_SPOTWELD

*INCLUDE_TRIM

*INCLUDE_UNITCELL

INCLUDE_*{OPTION}

Available options include:

<BLANK>

BINARY

NASTRAN

PATH

PATH_RELATIVE

STAMPED_SET

TRANSFORM

TRANSFORM_BINARY

STAMPED_PART_{*OPTION1*}_{*OPTION2*}_{*OPTION3*}

OPTION1: SET

OPTION2: MATRIX

OPTION3: INVERSE

The BINARY and TRANSFORM_BINARY options specify that the initial stress file, dynain, is written in a binary format. See the keyword *INTERFACE_SPRINGBACK.

The PATH option defines a directory in which to look for the include files. The program always searches the local directory first. If an include file is not found and the filename has no path, the program will search for it in all the directories defined by *INCLUDE_PATH. Multiple paths can defined with one *INCLUDE_PATH definition, i.e.,

```
*INCLUDE_PATH
Directory_path1
Directory_path2
Directory_path3
```

Directory paths are read until the next "*" card is encountered. A directory path can have up to 236 characters (see Remark 3).

The PATH_RELATIVE option is like the PATH option, except all directories are relative to the location of the input file. For example, if "i=/home/test/problems/input.k" is given on the command line, and the input contains

```
*INCLUDE_PATH_RELATIVE
Includes
../includes
```

then the two directories /home/test/problems/includes and /home/test/includes will be searched for include files.

The STAMPED_PART option applies only to thin shell elements and allows the plastic strain and thickness distribution of the stamping simulation to be mapped onto a part in the crash model.

1. When option 1, SET is used, the PID will be part set ID. All the parts included in this set will be considered in this mapping.
2. When option 2, MATRIX is used, translation matrix will be read directly and the orientation nodes will be ignored.
3. When option 3, INVERSE (must be used with MATRIX) is used, the matrix will be reversed first.

When STAMPED_SET is used, the target is a part set ID. Between the stamped part and the crash part, note the following points:

1. The outer boundaries of the parts do not need to match since only the regions of the crash part which overlap the stamped part are initialized.
2. Arbitrary mesh patterns are assumed.
3. Element formulations can change.
4. Three nodes on each part are used to reorient the stamped part for the mapping of the strain and thickness distributions. After reorientation, the three nodes on each part should approximately coincide.
5. The number of in plane integrations points can change.
6. The number of through thickness integration points can change. Full interpolation is used.
7. The node and element ID's between the stamped part and the crash part do not need to be unique.

The TRANSFORM option allows for node, element, and set ID's to be offset and for coordinates and constitutive parameters to be transformed and scaled.

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	C							

If the *INCLUDE command is used <u>without</u> options, multiple filenames can be specified, i.e.,

```
*INCLUDE
Filename1
Filename2
Filename3
```

which are processed sequentially. Filenames are read until the next "*" card is encountered.

Nastran Card. Additional Card for the NASTRAN keyword option.

Card 2	1	2	3	4	5	6	7	8
Variable	BEAMDF	SHELLDF	SOLIDDF					
Type	I	I	I					
Default	2	21	18					

Stamped Part Card 1. Additional Card for STAMPED_PART keyword option.

Card 2	1	2	3	4	5	6	7	8
Variable	PID	THICK	PSTRN	STRAIN	STRESS	INCOUT		RMAX
Type	I	I	I	I	I	I		F
Default	none	0	0	0	0	0		20.0

Stamped Part Card 2a. Additional card for STAMPED_PART option *not* ending in_MATRIX.

Card 3	1	2	3	4	5	6	7	8
Variable	N1S	N2S	N3S	N1C	N2C	N3C	TENSOR	THKSCL
Type	I	I	I	I	I	I	I	F
Default	0	0	0	0	0	0	0	1.0
Remarks	2	2	2	2	2	2	4	

Stamped Part (Matrix) Card 2b. Additional card for STAMPED_PART_MATRIX option.

Card 3	1	2	3	4	5	6	7	8
Variable	R11	R12	R13	XP				
Type	F	F	F	F				
Default	0	0	0	0				
Remarks	2	2	2	2				

Stamped Part (Matrix) Card 3. Additional card for STAMPED_PART_MATRIX option.

Card 4	1	2	3	4	5	6	7	8
Variable	R21	R22	R23	YP				
Type	F	F	F	F				
Default	0	0	0	0				
Remarks	2	2	2	2				

Stamped Part (Matrix) Card 4. Additional card for STAMPED_PART_MATRIX option.

Card 5	1	2	3	4	5	6	7	8
Variable	R31	R32	R33	ZP				
Type	F	F	F	F				
Default	0	0	0	0				
Remarks	2	2	2	2				

Remaining Stamped Part cards are optional.†

Stamped Part Card 6. Optional card for STAMPED_PART (with and without_MATRIX) keyword option.

Card 4	1	2	3	4	5	6	7	8
Variable	ISYM	IAFTER	PERCELE	IORTHO		ISRCOUT		
Type	I	I	F	I		I		

Stamped Part Card 6. Optional card for STAMPED_PART (with and without_MATRIX) keyword option.

Card 5	1	2	3	4	5	6	7	8
Variable	X01	Y01	Z01					
Type	F	F	F					

Stamped Part Card 7. Optional card for STAMPED_PART (with and without_MATRIX) keyword option.

Card 6	1	2	3	4	5	6	7	8
Variable	X02	Y02	Z02	X03	Y03	Z03		
Type	F	F	F	F	F	F		

Transform Card 1. Additional card for TRANSFORM keyword option.

Card 2	1	2	3	4	5	6	7	8
Variable	IDNOFF	IDEOFF	IDPOFF	IDMOFF	IDSOFF	IDFOFF	IDDOFF	
Type	I	I	I	I	I	I	I	

Transform Card 2. Additional card for TRANSFORM keyword option.

Card 3	1	2	3	4	5	6	7	8
Variable	IDROFF		PREFIX	SUFFIX				
Type	I		A	A				

Transform Card 3. Additional card for TRANSFORM keyword option.

Card 4	1	2	3	4	5	6	7	8
Variable	FCTMAS	FCTTIM	FCTLEN	FCTTEM	INCOUT1			
Type	F	F	F	A	I			

Transform Card 4. Additional card for TRANSFORM keyword option.

Card 5	1	2	3	4	5	6	7	8
Variable	TRANID							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
FILENAME	File name of file to be included in this keyword file, 80 characters maximum. If the STAMPED_PART option is active, this is the dynain file containing the results from metal stamping.
BEAMDF	LS-DYNA beam element type. Defaults to type 2.
SHELLDF	LS-DYNA shell element type. Defaults to type 21.
SOLIDDF	LS-DYNA solid element type. Defaults to type 18.
PID	Part ID of crash part for remapping.
THICK	Thickness remap: EQ.0: map thickness EQ.1: do not map thickness EQ.2: average value inside a circle defined by RMAX
PSTRN	Plastic strain remap: EQ.0: map plastic strain EQ.1: do not plastic strain EQ.2: average value inside a circle defined by RMAX
STRAIN	Strain remap: EQ.0: map strains EQ.1: do not map strains

VARIABLE **DESCRIPTION**

STRESS

Stress tensor remap:

EQ.0: map stress tensor and history variables

EQ.1: do not map stress tensor, only history variables

EQ.2: neither map stress tensor nor history variables

EQ.-1: map stress tensor in an internal large format (binary files)

EQ.-3: do not map stress tensor in an internal large format, only history (binary files)

INCOUT

EQ.1: to save the mapped data to a file called dyna.inc, which contains the mapped data for the part that is being mapped. This option is useful to do mapping using INCLUDE_STAMPED_PART and then save the mapped data for future use. When INCOUT is set to 2, the output file is in dynain format and the file name is dynain_xx (xx is the part or part set id); and when INCOUT is set to 3, the output file is in NASTRAN format, and the file name is: nastran_xx.

EQ.2: to save the mapped data for the specified part (PID) to a file called dynain_PID.

EQ.3: to save the mapped data for the specified part (PID) to a file called nastran_PID (in nastran format)

RMAX

Search radius. LS-DYNA remaps history variables from the mesh of the stamped part to the mesh of the crash part with a spatial tolerance of RMAX. If an element in the crash part lies within RMAX of the stamped part, data will be mapped to that element. If set less than 0.001, RMAX automatically assumes the default value of 20.

N1S

First of 3 nodes needed to reorient the stamped part.

N2S

Second of 3 nodes needed to reorient the stamped part.

N3S

Third of 3 nodes needed to reorient the stamped part.

N1C

First of 3 nodes needed to reorient the crash model part.

N2C

Second of 3 nodes needed to reorient the crash model part.

N3C

Third of 3 nodes needed to reorient the crash model part.

VARIABLE	DESCRIPTION
TENSOR	Tensor remap: EQ.0: map tensor data from history variables. (See Remark 4.) EQ.1: do not map tensor data from history variables
THKSCL	Thickness scale factor.
R11, R12, R33	Components of the transformation matrix.
XP, YP, ZP	Translational distance.
ISYM	Symmetric switch EQ.0: no symmetric mapping EQ.1: *yz* plane symmetric mapping EQ.2: *zx* plane symmetric mapping EQ.3: *zx* and *yz* planes symmetric mapping EQ.4: user defined symmetric plane mapping
IAFTER	Mirroring sequence switch EQ.0: generate a symmetric part before transformation EQ.1: generate a symmetric part after transformation
PERCELE	Percentage of elements that should be mapped to proceed (default = 0); otherwise an error termination occurs. See Remark 6.
IORTHO	Location of the material direction cosine in the array of history variables of an orthotropic material. See Remark 5.
ISRCOUT	Optional output of stamped part after transformation(s) EQ.0: no output is written NE.0: keyword output file "srcmsh_<ISRCOUT>" is created
X01, Y01, Z01	First point in the symmetric plane (required if ISYM.NE.0)
X02, Y02, Z02	Second point in the symmetric plane
X03, Y03, Z03	Third point in the symmetric plane
IDNOFF	Offset to node ID.

VARIABLE	DESCRIPTION
IDEOFF	Offset to element ID.
IDPOFF	Offset to part ID, nodal rigid body ID, constrained nodal set ID, Rigidwall ID, and *DATABASE_CROSS_SECTION.
IDMOFF	Offset to material ID and equation of state ID.
IDSOFF	Offset to set ID.
IDFOFF	Offset to function ID, table ID, and curve ID.
IDDOFF	Offset to any ID defined through *DEFINE except the FUNCTION, TABLE, and CURVE options (see IDFOFF).
IDROFF	Used for all offsets except for those listed above.
PREFIX	Prefix added to the beginning of the titles/heads defined in the keywords (like *MAT, *PART, *SECTION, *DEFINE, for examples) of the included file. A dot, “.”, is automatically added between the prefix and the existing title.
SUFFIX	Suffix added to the end of the titles/heads defined in the keywords of the included file. A dot, “.”, is automatically added between the suffix and the existing title.
FCTMAS	Mass transformation factor. For example, FCTMAS = 1000. When the original mass units are in tons and the new unit is kg.
FCTTIM	Time transformation factor. For example, FCTTIM=.001 when the original time units are in milliseconds and the new time unit is seconds.
FCTLEN	Length transformation factor.
FCTTEM	Temperature transformation factor consisting of a four character flag: FtoC (Fahrenheit to Centigrade), CtoF, FtoK, KtoF, KtoC, and CtoK.
INCOUT1	Set to 1 for the creation of a file, DYNA.INC, which contains the transformed data. The data in this file can be used in future include files and should be checked to ensure that all the data was transformed correctly.
TRANID	Transformation ID, if 0 no transformation will be applied. See the input DEFINE_TRANSFORMATION.

Remarks:

1. **Scalability.** To make the input file easy to maintain, this keyword allows the input file to be split into subfiles. Each subfile can again be split into sub-subfiles and so on. This option is beneficial when the input data deck is very large. Consider the following example:

```
*TITLE
full car model
*INCLUDE
carfront.k
*INCLUDE
carback.k
*INCLUDE
occupantcompartment.k
*INCLUDE
dummy.k
*INCLUDE
bag.k
*CONTACT
⋮
*END
```

Note that the command *END terminates the include file.

The carfront.k file can again be subdivided into rightrail.k, leftrail.k, battery.k, wheel-house.k, shotgun.k, etc.. Each *.k file can include nodes, elements, boundary conditions, initial conditions, and so on.

```
*INCLUDE
rightrail.k
*INCLUDE
leftrail.k
*INCLUDE
battery.k
*INCLUDE
wheelhouse.k
*INCLUDE
shotgun.k
⋮
*END
```

2. **Reorienting the Result of a Stamping Simulation for STAMPED_PART option.** When defining *INCLUDE_STAMPED_PART the target mesh must be read in before the include stamped part.

N1S, N2S, N3S, N1C, N2C, and N3C are used for transforming the stamped part to the crash part, such that it is in the same position as the crash part. If the stamped part is in the same position as the crash part then N1S, N2S, N3S, N1C, N2C, N3C can all be set to 0. Note: If these 6 nodes are input as 0, LS-DYNA will not transform the stamped part.

When symmetric mapping is used (ISYM is not zero), the three points should not be in one line.

If ISYM = 0, 1, 2, or 3, only the first point (X01,Y01, Z01) is needed

If ISYM = 4, all the three points are needed

3. **Path Length Limitations.** Filenames and pathnames are limited to 236 characters spread over up to three 80 character lines. When 2 or 3 lines are needed to specify the filename or pathname, end the preceding line with "␣+" (space followed by a plus sign) to signal that a continuation line follows. Note that the "␣+" combination is, itself, part of the 80 character line; hence the maximum number of allowed characters is 78 + 78 + 80 = 236.

4. **Mapping Material Data for Springback for STAMPED_PART option.** Certain material models (notably Material 190) have tensor data stored within the history variables. Within material subroutines this data is typically stored in element local coordinate systems. In order to properly map this information between models it is necessary to have the tensor data present on the *INITIAL_STRESS_SHELL card and have it stored in global coordinates. During mapping the data is then converted into the local coordinate system of the crash mesh. This data can be dumped into the dynain file that is created at termination time if the parameter FTENSR is set to 0 on the *INTERFACE_SPRINGBACK_DYNA3D card. Currently, the only material model that supports mapping of element history tensor data is Material 190.

5. **IORTHO.** If IORTHO is set, correct mapping between non-matching meshes is invoked for the directions of orthotropic materials. A list of appropriate values for several materials is given here:

IORTHO.EQ.1:	materials 23, 122, 157, 234
IORTHO.EQ.3:	materials 22, 33, 36, 133, 189, 233, 243
IORTHO.EQ.4:	material 59
IORTHO.EQ.6:	materials 58, 104, 158
IORTHO.EQ.8:	materials 54, 55
IORTHO.EQ.9:	material 39
IORTHO.EQ.10:	material 82

IORTHO.EQ.13: materials 2, 86, 103

6. **Mapping Mismatch with STAMPED_PART option.** Sometimes during mapping the two meshes (stamp mesh and crash mesh) do not fit exactly and therefore not all elements of the new mesh get results from the old mesh. Information about the total number of crash elements which are / are not mapped is given in the message file. By default (PERCELE=0), the calculation continues even with zero number of mapped elements. With PERCELE>0 the percentage of minimum number of elements can be defined, which have to be mapped. If a percentage less than PERCELE is mapped, calculation stops with an error termination.

7. **NASTRAN Option.** The transformed LS-Dyna deck for *INCLUDE_NASTRAN will be automatically written to file DYNA.INC.

*INCLUDE_AUTO_OFFSET

Purpose: This particular *INCLUDE keyword offsets node and element IDs to avoid duplication during stamping simulations. In stamping simulations the rigid tools often undergo several iterations of modifications. The node or element IDs comprising the new tools sometimes conflict with other parts of the model, which makes it difficult to automate the process simulation. This keyword automatically checks for and fixes duplicate IDs. The *CONTROL_FORMING_MAXID keyword is related.

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	C							

VARIABLE	DESCRIPTION
FILENAME	File name to be included.

Remarks:

This keyword can be used to offset element and node IDs of the tooling. *The keyword will not offset meshes with initial stress and strain information*. As such, sheet blank (including dynain file) should always be included first using the *INCLUDE keyword, followed by *INCLUDE_AUTO_OFFSET to offset tooling mesh IDs which do not have stress and strain information.

Incoming element and node IDs of the tooling mesh files such as the punch, die, and binder, can be overlapped with each other, or overlapped with those on the sheet blank. Multiple *INCLUDE_AUTO_OFFSET can be used to include punch, die, binder separately, if desired. For example, four different components of the tooling, upper die, lower punch, binder and gage pins can be included and their element and node IDs properly offset after those of a gravity-loaded sheet blank:

```
*INCLUDE
gravity.dynain
*INCLUDE_AUTO_OFFSET
upperdie.k
*INCLUDE_AUTO_OFFSET
lowerpunch.k
*INCLUDE_AUTO_OFFSET
binder.k
*INCLUDE_AUTO_OFFSET
pins.k
```

All of the included meshes can have conflicting mesh IDs starting from “1”. Mesh IDs will be offset and reordered in the order of the tool inclusion using *INCLUDE_AU-

TO_OFFSET. Included tool files whose mesh IDs do not overlap with those on either the blank or other tools will not be offset or reordered. In many circumstances this feature allows the user to bypass the metal forming GUI when updating just one or two tooling pieces.

Revision information:

This feature is available in SMP and MPP starting in LS-DYNA Revision 92417. Revision 117818 extends the keyword to beams (used to model draw beads, for example) and solids.

INCLUDE_COMPENSATION_*{OPTION}

Purpose: This group of keywords allow for the inclusion of stamping die geometry information for springback compensation. In addition, trim curves from the target geometry can be included for mapping onto the intermediate compensated tool geometry, which can be used for the next compensation iteration. Furthermore, compensation can be done for a localized tool region. These keywords must be used together with *INTERFACE_COMPENSATION_NEW.

Options available include:

BLANK_BEFORE_SPRINGBACK

BLANK_AFTER_SPRINGBACK

DESIRED_BLANK_SHAPE

COMPENSATED_SHAPE

CURRENT_TOOLS

TRIM_CURVE

CURVE

ORIGINAL_DYNAIN

SPRINGBACK_INPUT

COMPENSATED_SHAPE_NEXT_STEP

SYMMETRIC_LINES

ORIGINAL_RIGID_TOOL

NEW_RIGID_TOOL

ORIGINAL_TOOL

UPDATED_BLANK_SHAPE

UPDATED_RIGID_TOOL

Blank Before Springback Card. Additional card for BLANK_BEFORE_SPRINGBACK keyword option.

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	C							
Default	blank0.tmp							

Blank After Springback Card. Additional card for BLANK_AFTER_SPRINGBACK keyword option.

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	C							
Default	spbk.tmp							

Desired Blank Shape Card. Additional card for DESIRED_BLANK_SHAPE keyword option.

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	C							
Default	reference0.dat							

Compensated Shape Card. Additional card for COMPENSATED_SHAPE keyword option.

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	C							
Default	reference1.dat							

Current Tools Card. Additional card for CURRENT_TOOLS keyword option.

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	C							
Default	rigid.tmp							

Generic Filename Card. Additional Card for TRIM_CURVE, CURVE, ORIGINAL_DYNAIN, SPRINGBACK_INPUT, COMPENSATED_SHAPE_NEXT_STEP, ORIGINAL_RIGID_TOOL, NEW_RIGID_TOOL, ORIGINAL_TOOL, UPDATED_BLANK_SHAPE, and UPDATED_RIGID_TOOL keyword options.

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	C							
Default	See Remarks							

Symmetric Lines Cards. Additional card for SYMMETRIC_LINES keyword option.

Card 1	1	2	3	4	5	6	7	8
Variable	SYMID	SYMXY	X0	Y0				
Type	I	I	F	F				
Default	1	none	0.0	0.0				

VARIABLE	DESCRIPTION

FILENAME

For options below, input the name of the keyword files containing nodes and elements information, with adaptive constraints if exist. Currently all sheet blanks must have the same numbers of nodes and elements.

BLANK_BEFORE_SPRINGBACK,

BLANK_AFTER_SPRINGBACK,

DESIRED_BLANK_SHAPE,

COMPENSATED_SHAPE,

CURRENT_TOOLS,

COMPENSATED_SHAPE_NEXT_STEP

For option ORIGINAL_DYNAIN, input the dynain file name from LS-DYNA simulation (for example, trimmed panel from ITER0 baseline simulation) which contains model information, adaptive constraints, stress and strain tensor information. This keyword is to be used in conjunction with *INTERFACE_COMPENSATION_NEW_ACCELATOR.

For option SPRINGBACK_INPUT, give the file name of springback simulation input deck for the baseline ITER0 simulation. This keyword is to be used in conjunction with *INTERFACE_COMPENSATION_NEW_ACCELATOR.

For option TRIM_CURVE, input the name of the keyword file containing x, y, z coordinates as defined using keyword *DEFINE_CURVE_TRIM_3D (only TCTYPE = 0, or 1 is supported). This option is used to map the trim curve to the new,

VARIABLE **DESCRIPTION**

compensated tooling mesh for next iterative simulation.

For option CURVE, input the name of the keyword file containing x, y, z coordinates of two curves defining the compensation zone, using keywords: *DEFINE_CURVE_COMPENSATION_CONSTRAINT_BEGIN, and, *DEFINE_CURVE_COMPENSATION_CONSTRAINT_END. This option is for compensation of localized tooling areas.

All foregoing keyword options are used together with *INTERFACE_COMPENSATION_NEW.

For options ORIGINAL_RIGID_TOOL and NEW_RIGID_TOOL, input the file names of the keyword file containing meshes of the rigid tools. This option is used to smooth distorted meshes of localized tool surfaces. These keyword options are used together with *INTERFACE_COMPENSATION_NEW_LOCAL_SMOOTH.

For option ORIGINAL_TOOL, input the file name of the original tool (without any compensation) mesh containing nodes and elements information in keyword format. This option allow the use of the original tool mesh, which is of higher quality, in the iterative compensation runs, to minimize the tool surface mesh distortion in the addendum and binder areas of the compensated tool (see Remarks). These keyword options are used together with *INTERFACE_COMPENSATION_NEW.

For options UPDATED_BLANK_SHAPE, and UPDATED_RIGID_TOOL, input the respective mesh information in keyword format. The updated blank shape is the blank formed (or trimmed) shape based on the new tool (die) geometry. These options allow for updating of compensated tool shape for small part shape changes, without the need to go through a full-blown iterative compensation loop again (see Remarks). The options are used together with *INTERFACE_COMPENSATION_NEW_PART_CHANGE, among others.

SYMID ID of the symmetric condition being defined.

VARIABLE	DESCRIPTION
SYMXY	Code defining symmetric boundary conditions: EQ.1: symmetric about y-axis. EQ.2: symmetric about x-axis.
X0, Y0	Coordinates of a point on the symmetric plane.

Default Filenames:

Keyword Option	Default Filename
UPDATED_BLANK_SHAPE	updatedpart.tmp
UPDATED_RIGID_TOOL	newrigid.tmp

About various options:

This group of keywords is used in conjunction with *INTERFACE_COMPENSATION_-NEW, to compensate stamping tool shapes for springback with an iterative method. The method approaches the final target design intent from two opposite directions from iteration to iteration. A typical successful compensation requires about 3 to 4 iterations.

1. **BLANK_BEFORE_SPRINGBACK.** When the option BLANK_BEFORE_-SPRINGBACK is used, the included file is the mesh information in keyword format in the first state (from d3plot) of the springback simulation, or the "dynain" file after trimming (before springback and with no mesh coarsening). The default file name is "blank0.tmp".

2. **BLANK_AFTER_SPRINBACK.** When the option BLANK_AFTER_SPRINBACK is used, the included file is the "dynain" file after springback, or the last state mesh (from d3plot) of the springback. The default file name is "spbk.tmp".

3. **DESIRED_BLANK_SHAPE.** When the option DESIRED_BLANK_SHAPE is used, the included file is the "dynain" file after trimming in the first iteration. This file never changes in all subsequent iterative compensation. The file name default is "reference0.dat".

4. **COMPENSATED_SHAPE.** When the option COMPENSATED_SHAPE is used, the included file for the first iteration, is a "dynain" file, same as in the option DESIRED_BLANK_SHAPE; and for the following compensation iterations, this

file is obtained from the file "disp.tmp" generated as an output file during the previous compensation iteration. The default file name is "reference1.dat".

5. **CURRENT_TOOLS.** When the option CURRENT_TOOLS is used, the included file is the file containing the tool mesh in the keyword format. This is the tool mesh from the last compensation run and used for the current forming simulation. The draw bead nodes have to be included in this file so that they will be modified together with the rigid tools. The default file name is "rigid.tmp", and if the file is named as "rigid0.tmp" the elements of the tools get refined along the outline of the part.

6. **TRIM_CURVE.** When the option TRIM_CURVE is used, trim curves off the current tools are mapped onto the compensated tools for the trimming operation in the next iteration.

   If the trimming simulation uses the IGES format trim curves, a new file "geocur.trm" will be generated at the end of the trimming simulation. The file basically contains XYZ data of the trim curves in keyword *DEFINE_CURVE_TRIM_{OPTIONS}, which is used for the compensation run. Note that the variable TCTYPE in the keyword must be set to "0" (or "1") for the compensation. Length of lines everywhere in the compensated part are calculated according to springback amounts (including the die expansion factors, therefore no die expansion needs to be included in the NC machining of the compensated tooling). These mapped trim curves can be used for die development on the compensated tools and for laser trimming of stamped panels. Procedures outline in keyword manual pages *INTERFACE_BLANKSIZE can be followed to convert in LS-PrePost IGES file of the trim curves to XYZ format (and vice versa) used in this keyword.

   In an example keyword input shown below, the file name for this option is trimcurves.k. The format is in XYZ format, written with LS-PrePost:

```
*DEFINE_CURVE_TRIM_3D
$#    tcid    tctype      tflg      tdir     tctol      toln     nseed
      1116         1         1               0.100         1
$#                cx                  cy                  cz
           178.05170          -326.24771           51.924496
           177.77397          -301.90869           50.288792
           177.29764          -265.39716           48.594341
...
```

7. **CURVE.** When the option CURVE is used, it allows for die face compensation of a local region in a stamping die. This option is used in conjunction with two more keywords defining two enclosed curves that form the compensation zone in position coordinates x, y, z: *DEFINE_CURVE_COMPENSATION_CONSTRAINT_BEGIN, and *DEFINE_CURVE_COMPENSATION_CONSTRAINT_END. Detailed usage of these two keywords is available in the related manual pages.

Full Example:

The following example is for compensation of a localized area, defined by the file curves.k. Trim lines are mapped onto the new compensated rigid tool, with trimcurves.k. Both files which were generated by LS-PrePost 4.0 are in the "XYZ format". A detailed explanation of each keyword is given in the manual pages related to *INTERFACE_COMPENSATION_NEW.

```
*KEYWORD
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+--
*INTERFACE_COMPENSATION_NEW
$   METHOD        SL        SF     ELREF      PSID    UNDRCT    ANGLE NLINEAR
         8    10.000     1.000         0         1         0      0.0       1
*INCLUDE_COMPENSATION_BLANK_BEFORE_SPRINGBACK
blank0.k
*INCLUDE_COMPENSATION_BLANK_AFTER_SPRINGBACK
spbk.k
*INCLUDE_COMPENSATION_DESIRED_BLANK_SHAPE
reference0.k
*INCLUDE_COMPENSATION_COMPENSATED_SHAPE
reference1.k
*INCLUDE_COMPENSATION_CURRENT_TOOLS
tools.k
*INCLUDE_COMPENSATION_TRIM_CURVE
trimcurves.k
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$ For compensation of a localize region only, add the following keyword:
*INCLUDE_COMPENSATION_CURVE
curves.k
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
*SET_PART_LIST
$      PSID
         1
$      PID
         3
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+--
*END
```

The option ORIGINAL_DYNAIN and SPRINGBACK_INPUT are used together with keyword *INTERFACE_COMPENSATION_NEW_ACCELATOR, for a springback compensation with a faster convergence rate and a simplified user interface. For detailed usage, please refer to manual pages under *INTERFACE_COMPENSTION_{*OPTION*}. Here a complete keyword input is provided:

```
*KEYWORD
*INTERFACE_COMPENSATION_NEW_ACCELATOR
$   ISTEPS      TOLX      TOLY      TOLZ    OPTION
         3      0.20      0.20       0.2        1
*INCLUDE_COMPENSATION_ORIGINAL_DYNAIN
./case20trimmed.dynain
*INCLUDE_COMPENSATION_SPRINGBACK_INPUT
./spbk.dyn
*END
```

The option COMPENSATED_SHAPE_NEXT_STEP enables compensation of tools for the next die process. It is used in conjunction with keyword *INTERFACE_COMPENSATION_NEW_MULTI_STEPS, which is discussed in the corresponding manual pages. Here a complete input deck is given below:

```
*KEYWORD
*INTERFACE_COMPENSATION_NEW_MULTI_STEPS
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$   METHOD        SL        SF     ELREF      PSID    UNDRCT     ANGLE   NLINEAR
         8     6.000      1.00         1         1         0         0         1
*INCLUDE_COMPENSATION_DESIRED_BLANK_SHAPE
reference0.tmp
*INCLUDE_COMPENSATION_COMPENSATED_SHAPE_NEXT_STEP
Reference1_flanging.tmp
*INCLUDE_COMPENSATION_CURRENT_TOOLS
rigid.tmp
*SET_PART_LIST
$      PSID
          1
$       PID
          2
*END
```

The option SYMMTRIC_LINES applies to compensation Method 7 and 8, as discussed in *INTERFACE_COMPENSATION_NEW. In a complete keyword input example below, part set ID 1 is being compensated with symmetric boundary condition about X-axis. The symmetric plane passes a point with coordinates of x = 101.5, and y = 0.0.

```
*KEYWORD
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$*INTERFACE_COMPENSATION_NEW
$ Method = 8 changes the binder; Method = 7 binder/P.O.  no changes.
*INTERFACE_COMPENSATION_NEW
$   METHOD        SL        SF     ELREF      PSID    UNDRCT     ANGLE   NLINEAR
         7    10.000     1.000         2         1         1       0.0         1
*INCLUDE_COMPENSATION_BLANK_BEFORE_SPRINGBACK
./state1.k
*INCLUDE_COMPENSATION_BLANK_AFTER_SPRINGBACK
./state2.k
*INCLUDE_COMPENSATION_DESIRED_BLANK_SHAPE
./state1.k
*INCLUDE_COMPENSATION_COMPENSATED_SHAPE
./state1.k
*INCLUDE_COMPENSATION_CURRENT_TOOLS
./currenttools.k
*INCLUDE_COMPENSATION_SYMMETRIC_LINES
$    SYMID     SYMXY        X0        Y0
         1         2     101.5       0.0
$ SYMXY = 2: symmetric about X-axis
*SET_PART_LIST
$     PSID
         1
$      PID
         1
*END
```

The options ORIGINAL_RIGID_TOOL and NEW_RIGID_TOOL are used together with *INTERFACE_COMPENSATION_NEW_LOCAL_SMOOTH, and *SET_NODE_LIST_SMOOTH, to smooth local areas of distorted meshes of a tooling surface. Details can be found in manual pages for *INTERFACE_COMPENSATION_NEW_LOCAL_SMOOTH.

The option ORIGINAL_TOOL is used to obtain a smoother mesh for the addendum and binder region for the current compensation, using the original tool mesh (of better quality) instead of the last compensated tool mesh (maybe distorted). This reduces the accumulative error in mesh extrapolation outside of the trim lines. Details can be found in manual pages for *INTERFACE_COMPENSATION_NEW.

The options UPDATED_BLANK_SHAPE, and UPDATED_RIGID_TOOL calculate a new compensated tool shape according to the updated blank shape, thus eliminating the need to go through a full-blown iterative compensation loop again. Note that these options are intended only for small part changes that do not substantially affect the amount of springback. More details can be found in manual pages for *INTERFACE_-COMPENSATION_NEW_PART_CHANGE.

Revision information:

The option TRIM_CURVE is available starting in Revision 60398. The options ORIGINAL_DYNAIN, and SPRINGBACK_INPUT are available starting in Revision 61264. The option COMPENSATED_SHAPE_NEXT_STEP is available starting in Revision 61406. The option CURVE is available starting in Revision 62038. The option SYMMETRIC_LINES is available starting in Revision 63618 (updated in Rev. 83711). The options of ORIGINAL_RIGID_TOOL and NEW_RIGID_TOOL are available starting in Revision 73850. The option ORIGINAL_TOOL is available starting in Revision 82701. The options UPDATED_BLANK_SHAPE, and UPDATED_RIGID_TOOL are available starting in Revision 82698.

.

*INCLUDE_MULTISCALE_SPOTWELD

Purpose: To define a type of MULTISCALE spot weld to be used for coupling and for modeling of spot weld failure.

Card 1	1	2	3	4	5	6	7	8
Variable	TYPE							
Type	I							
Default	none							

Card 2	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	C							
Default	none							

VARIABLE	DESCRIPTION
TYPE	TYPE for this multiscale spot weld. This type is used in the keyword *DEFINE_SPOTWELD_MULTISCALE. Any unique integer will do.
FILENAME	Name of file from which to read the spot weld definition.

Remarks:

This capability is available *only* in the MPP version of LS-DYNA.

With the multiscale spot weld feature heuristic spot weld models are replaced with zoomed-in geometrically and constitutively correct continuum models, which, in turn are coupled to the large-scale calculation without reducing the time step. In some respects, multiscale models are similar to the "hex spot weld assemblies," capability but more general in terms of their geometry. Because the spot weld models are run in a separate process, they can run at a much smaller time step without slowing down the rest of the simulation. A brief outline of their use looks like this:

- The user creates one (or more) detailed models of their spot welds, and includes these definitions into their model using the keyword *INCLUDE_MULTISCALE_SPOTWELD
- The user indicates which beam (or hex assembly) spot welds should be coupled to these models via the keyword *DEFINE_SPOTWELD_MULTISCALE
- When MPP-DYNA is started, a special (MPI dependent) invocation is required in order to run in a "multiple program" mode. Effectively, two separate instances of MPP-DYNA are started together, one to run the full model and a separate instance to run the spot welds.
- As the master process runs, each cycle it communicates to the slave process deformation information for the area surrounding each coupled spot weld. The slave process imposes this deformation on the detailed spot welds, computes a failure flag for each, and communicates this back to the master process.
- The coupled spot welds in the master process have their failure determined solely by these failure flags.

The file referred to on the *INCLUDE_MULTISCALE_SPOTWELD card should contain one generic instance of a detailed spot weld. For each coupled spot weld in the main model, a specific instance of this spot weld will be generated which is translated, rotated, and scaled to match the spot weld to which it is coupled. In this way, many spot welds can be coupled with only a single *INCLUDE_MULTISCALE_SPOTWELD. The included file should contain everything required to define the spot weld, such as *MAT and *PART definitions, any required *DEFINE_CURVEs, etc., as well as *NODE and *ELEMENT definitions. In order for the translation and scaling to work properly there are some assumptions made about the spot weld model:

- It consists entirely of solid elements.
- The z-axis is aligned with the coupled spot weld in the main model, with $z = 0$ and $z = 1$ at the two ends of the spot weld.
- The cross sectional area of the spot weld in the xy plane is equal to 1.
- That portion of the "top" and "bottom" of the spot weld that are coupled are identified using a single *SET_NODE_LIST card.
- One *BOUNDARY_COUPLED card referencing the *SET_NODE_LIST of the boundary nodes is required. It must specify a coupling type of 2 and a coupling program of 1.
- The spot weld model does not support *INCLUDE cards.

Failure of the fine model is determined topologically. Any element of the spot weld having all four nodes of one of its faces belonging to the *SET_NODE_LIST of tied nodes is classified as a "tied" element. The "tied" elements are partitioned into two disjoint sets: the "top," and "bottom". When there is no longer a complete path from any "top" to any "bottom" element (where a "path" passes through non-failed elements that share a common face), then the spot weld has failed. Note that this places some restrictions on the *SET_NODE_LIST and element geometry, namely that some "tied" elements exist, and the set of "tied" elements consists of exactly two disjoint subsets.

The specifics of launching a multi-program MPI program are installation dependent. But the idea behind running a coupled model is that you want to run one set of MPI ranks as if you were running a normal MPP-DYNA job, such as:

```
mpirun –np 4 mppdyna i=input.k memory=200m p=pfile
```

and a second set with just the command line argument "slave" (no input file):

```
mpirun –np 4 mppdyna slave memory=100m p=pfile
```

The main instance knows to look for the slave (because of the presence of the *INCLUDE_MULTISCALE_SPOTWELD card), and will run the main model. The "slave" instance will run all the detailed spot weld models. Due to the nature of the coupling, the main model cannot progress when the detailed spot welds are being processed, nor can the detailed spot welds run while the main model is being computed. From a processor efficiency standpoint, it therefore makes sense to run as many slave processes as master processes, and run them on the same CPUs, so that each processing core has one slave and one master process running on it. But you don't have to – the processes are independent and you can have any number of either.

*INCLUDE_TRIM

Purpose: This keyword is developed to reduce memory requirements and CPU time (as compared with *INCLUDE) during trimming in sheet metal forming. This keyword is intended to be used together with the *DEFINE_CURVE_TRIM_OPTION and *ELEMENT_TRIM keywords.

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	C							

VARIABLE	DESCRIPTION
FILENAME	File name of the part to be trimmed.

General remarks:

The name of the file to be trimmed should be included in a usual LS-DYNA input file for trimming, as is in *INCLUDE. Model information, stress and strain tensors should be all in one dynain file generated from LS-DYNA simulation. For example, a drawn panel from a previous simulation can be included in a current trim input file as follows,

```
*INCLUDE_TRIM
Drawnpanel.dynain
*ELEMENT_TRIM
             ⋮
*DEFINE_CURVE_TRIM_3D
             ⋮
*CONTROL_ADAPTIVE_CURVE
             ⋮
```

This feature has been developed in conjunction with the *Ford Motor Company Research and Advanced Engineering Laboratory*, and is implemented in LS-PrePost as of version 4.0 under the metal forming application under eZ-Setup (http://ftp.lstc.com/anonymous/outgoing/lsprepost/4.0/metalforming/). Compared with *INCLUDE, this keyword draws much less computer memory and runs much faster.

Furthermore, in case where to-be-trimmed sheet blank has no stress and strain information (no *INITIAL_STRESS_SHELL, and *INITIAL_STRAIN_SHELL cards present in the sheet blank keyword/dynain file), the *bare* *INCLUDE keyword *must* be used.

Application example:

Referring to the table below (parts courtesy of the *Ford Motor Company*), compared with simply the keyword *INCLUDE, this keyword reduces memory requirement for trimming by more than 50%. Levels of CPU time reductions vary, in some cases more than 50%.

Performance Improvements

		Roof	Hood Inr	B-Plr	Fender	BSA Otr	Door Otr	Wheel House (2 in 1)	Boxside Otr
#Element		410810	1021171	351007	189936	380988	315556	261702	1908369
CPU	old	7m26s	10m20s	3m11s	2m6s	5m45s	4m27s	2m52s	27m31s
	new	4m	9m18s	2m56s	1m22s	4m54s	3m35s	2m30s	13m59s
Memory (MW)	old	282	616	221	119	233	217	157	1150
	new	112	383	117	50	130	114	75	539

Revision information:

This feature is available in LS-DYNA Revision 62207 or later releases, where the output of strain tensors for the shells is included. Prior Revisions do not include strain tensors for the shells.

*INCLUDE_UNITCELL

Purpose: This card creates a unit cell model with periodic boundary conditions using *CONSTRAINED_MULTIPLE_GLOBAL.

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	C							
Default	none							

Card 2	1	2	3	4	5	6	7	8
Variable	INPT	OUPT	NEDOF					
Type	I	I	I					
Default	0	0	0					
Remarks	1		2					

Card 3	1	2	3	4	5	6	7	8
Variable	DX	DY	DZ	NEX	NEY	NEZ	NNPE	
Type	F	F	F	I	I	I	I	
Default	1.0	1.0	1.0	1	1	1	8	

Card 4	1	2	3	4	5	6	7	8
Variable	NOFF	EOFF	PNM					
Type	I	I	I					
Default	none	none	none					

Card 5	1	2	3	4	5	6	7	8
Variable	CNX	CNY	CNZ					
Type	I	I	I					
Default	none	none	none					

Node ID Cards. Input is terminated at the next keyword ("*") card

Card 6	1	2	3	4	5	6	7	8
Variable	ECNX	ECNY	ECNZ					
Type	I	I	I					
Default	none	none	none					

VARIABLE	DESCRIPTION
FILENAME	Name of the file containing the information of unit cell.
INPT	Type of input EQ.0: read *NODE information from the include file and add periodic boundary conditions to the include file. EQ.1: create a unit cell mesh with periodic boundary conditions, and output to the include file.

VARIABLE	DESCRIPTION
OUPT	Type of output EQ.1: create a new main keyword file where the keyword *INCLUDE_UNITCELL is replaced by *INCLUDE with the include file name.
NEDOF	Number of extra nodal degrees of freedom (DOFs) for user-defined element. In the current implementation, the limit of NEDOF is 15.
DX	Defines the x-dimension of unit cell.
DY	Defines the y-dimension of unit cell.
DZ	Defines the z-dimension of unit cell.
NEX	Defines number of elements along x-direction.
NEY	Defines number of elements along y-direction.
NEZ	Defines number of elements along z-direction.
NNPE	Defines number of nodes per element. The current implementation supports only 4-node tetrahedron or 8-node hexahedron elements.
NOFF	Defines offset of nodal IDs.
EOFF	Defines offset of elemental IDs.
PNM	Defines part ID.
CNX	Defines nodal ID of the 1st control point for the constraint in x direction.
CNY	Defines nodal ID of the 2nd control point for the constraint in y direction.
CNZ	Defines nodal ID of the 3rd control point for the constraint in z direction.
ECNX	Defines nodal ID of extra control point for the constraint in x direction of 3 extra nodal DOFs.
ECNY	Defines nodal ID of extra control point for the constraint in y direction of 3 extra nodal DOFs.

VARIABLE	DESCRIPTION
ECNZ	Defines nodal ID of extra control point for the constraint in z direction of 3 extra nodal DOFs.

Remarks:

1. **Include File Field.** If INPT=0, the geometry and discretization information of unit cell are from the include file. In this case, the parameters in cards 3 and 4 are ignored.

2. **Extra Degrees of Freedom.** The extra degrees of freedom (DOFs) specified by NEDOF>0 are represented by extra nodes with regular x, y and z DOFs. When NEDOF=7, for example, the following chart shows the mapping from the extra DOFs to the regular ones of extra nodes:

Extra Node #	Extra DOFs	Regular DOFs
1	1	x
	2	y
	3	z
2	4	x
	5	y
	6	z
3	7	x

In this case, 3 control points for x, y, and z directions, respectively, need to be defined for each extra node.

*INITIAL

The keyword *INITIAL provides a way of initializing velocities and detonation points. The keyword control cards in this section are defined in alphabetical order:

*INITIAL_AIRBAG_PARTICLE_POSITION

*INITIAL_ALE_MAPPING

*INITIAL_AXIAL_FORCE_BEAM

*INITIAL_CONTACT_WEAR

*INITIAL_DETONATION

*INITIAL_EOS_ALE

*INITIAL_FATIGUE_DAMAGE_RATIO

*INITIAL_FIELD_SOLID

*INITIAL_FOAM_REFERENCE_GEOMETRY

*INITIAL_GAS_MIXTURE

*INITIAL_HYDROSTATIC_ALE

*INITIAL_IMPULSE_MINE

INITIAL_INTERNAL_DOF_SOLID_*{OPTION}

*INITIAL_LAG_MAPPING

*INITIAL_MOMENTUM

*INITIAL_PWP_DEPTH

INITIAL_STRAIN_SHELL_*{OPTION}

INITIAL_STRAIN_SOLID_*{OPTION}

*INITIAL_STRAIN_TSHELL

*INITIAL_STRESS_BEAM

*INITIAL_STRESS_DEPTH

*INITIAL_STRESS_SECTION

INITIAL_STRESS_SHELL_*{OPTION}

*INITIAL_STRESS_SOLID

*INITIAL_STRESS_SPH

*INITIAL_STRESS_TSHELL

INITIAL_TEMPERATURE_*{OPTION}

*INITIAL_VEHICLE_KINEMATICS

There are two alternative sets of keywords for setting initial velocities. Cards from one set cannot be combined with cards from the other. **Standard velocity cards:**

*INITIAL_VELOCITY

*INITIAL_VELOCITY_NODE

*INITIAL_VELOCITY_RIGID_BODY

Alternative initial velocity cards supporting initial rotational about arbitrary axes and start times. **Alternative velocity cards:**

*INITIAL_VELOCITY_GENERATION

*INITIAL_VELOCITY_GENERATION_START_TIME

INITIAL_VOID_*{OPTION}

*INITIAL_VOLUME_FRACTION

*INITIAL_VOLUME_FRACTION_GEOMETRY

*INITIAL_AIRBAG_PARTICLE_POSITION

Purpose: This card initializes the position of CPM initial air particle to the location specified.

Card 1	1	2	3	4	5	6	7	8
Variable	Bag_ID							
Type	I							
Default	none							

Particle Cards. The *i*th card specifies the location of the *i*th particle. LS-DYNA expects one card for each particle, if fewer cards are supplied the coordinates will be reused and particles may share the same location at the beginning of the simulation.

Card	1	2	3	4	5	6	7	8
Variable		X	Y	Z				
Type	8x	F	F	F				
Default								

VARIABLE	DESCRIPTION
Bag_ID	Airbag ID defined in *AIRBAG_PARTICLE_ID card
X	*x* coordinate
Y	*y* coordinate
Z	*z* coordinate

*INITIAL_ALE_MAPPING

Purpose: This card initializes the current ALE run with data from the last cycle of a previous ALE run. Data are read from a mapping file specified by "map=" on the command line (see Remark 6). To map data histories (not just the last cycle) to a region of selected elements see *BOUNDARY_ALE_MAPPING.

The following transitions are allowed:

1D → 2D
1D → 3D

2D → 2D
2D → 3D

3D → 3D
3D → 2D

Card 1	1	2	3	4	5	6	7	8
Variable	PID	TYP	AMMSID					
Type	I	I	I					
Default	none	none	none					

Card 2	1	2	3	4	5	6	7	8
Variable	X0	Y0	Z0	VECID	ANGLE			
Type	F	F	F	I	F			
Default	0.0	0.0	0.0	none	none			

VARIABLE **DESCRIPTION**

PID Part ID or part set ID.

TYP Type of "PID" (see Remark 1):

EQ.0: part set ID (PSID).

EQ.1: part ID (PID).

AMMSID Set ID of ALE multi-material groups defined in *SET_MULTI-MATERIAL_GROUP. See Remark 1.

XO Origin position in global x-direction. See Remarks 2 and 5.

VARIABLE	DESCRIPTION
YO	Origin position in global *y*-direction. See Remarks 2 and 5.
ZO	Origin position in global *z*-direction. See Remarks 2 and 5.
VECID	ID of the symmetric axis defined by *DEFINE_VECTOR. See Remarks 3 and 5.
ANGLE	Angle of rotation in degrees around an axis defined by *DEFINE_VECTOR for the 3D to 3D mapping. See Remark 4.

Remarks:

1. **Mapping of Ale Multi-Material Groups.** The routines of this card need to know which mesh will be initialized with the mapping data, and more specifically, which multi-material groups. The first two fields, PID, and TYP, define the mesh. The third field, AMMSID, refers to a multi-material group list ID; see the *SET_MULTI-MATERIAL_GROUP_LIST card. The group list AMMSID should have as many elements as there are groups in the previous calculation (see *ALE_MULTI-MATERIAL_GROUP).

   Example: If the previous model has 3 groups, the current one has 5 groups and the following mapping is wanted.

   Group 1 from the previous run → Group 3 in the current run
   Group 2 from the previous run → Group 5 in the current run
   Group 3 from the previous run → Group 4 in the current run

   The *SET_MULTI-MATERIAL_GROUP_LIST card should be set as follows:

```
*SET_MULTI-MATERIAL_GROUP_LIST
300
3,5,4
```

   In special cases, a group can be replaced by another. If the group 4 in the previous example should be replaced by the group 3, the keyword setup would be modified to have -3 instead of 4. The minus sign is a way for the code to know that the replacing group (-3 replaces 4) is a complement of the group 3:

```
*SET_MULTI-MATERIAL_GROUP_LIST
300
3,5,-3
```

2. **Coordinate System Origin.** The location to which the data is mapped is controlled by the origin of the coordinate system (XO, YO, ZO).

3. **Symmetry Axis.** For a mapping file created by a previous asymmetric model, the symmetric axis orientation in the current model is specified by VECID. For a mapping file created by a 3D or 1D spherical model, the vector VECID is read but ignored. For a 3D to 3D mapping the vector is used if the parameter ANGLE is defined (see Remark 4).

4. **Rotating 3D Data Onto a 3D calculation.** For a mapping from a previous 3D run to a current 3D model the previous 3D data will be rotated about the vector, VECID, through an angle specified in the ANGLE field.

5. **Plain Strain, and 3D to 2D.** The definitions of X0, Y0, Z0 and VECID change in the case of the following mappings:

   a) plain strain 2D (ELFORM = 13 in *SECTION_ALE2D) to plain strain 2D

   b) plain strain 2D to 3D

   c) 3D to 2D

   While, VECID still defines the y-axis in the 2D domain, the 3 first parameters in *DEFINE_VECTOR, additionally, define the location of the origin. The 3 last parameters defines a position along the y-axis. For this case when 2D data is used in a 3D calculation the point X0, Y0, Z0 together with the vector, VECID, define the plane.

6. **Mapping File.** Including the command line argument "map=" will invoke the creation of a mapping file. When the keyword INITIAL_ALE_MAPPING is not in the input deck, but the argument "map=" is present on the command line, the ALE data from the last cycle is written in the mapping file. This file contains the following nodal and element data:

   - nodal coordinates (last step)
   - nodal velocities
   - part ids
   - element connectivities
   - element centers
   - densities
   - volume fractions
   - stresses
   - plastic strains
   - internal energies
   - bulk viscosities

- relative volumes

Chained Mappings. To chain mapping operations so that LS-DYNA both reads and writes a mapping file the command line argument "map1=" is necessary. If the keyword INITIAL_ALE_MAPPING is in the input deck and the prompt "map=" is in the command line, the ALE data is read from the mapping file defined by "map=" to initialize the run. Data from the last cycle are written in the mapping file defined by "map1=".

*INITIAL_AXIAL_FORCE_BEAM

Purpose: Initialize the axial force resultants in beam elements that are used to model bolts. This option works with *MAT_SPOTWELD with beam type 9, a Hughes-Liu type beam.

Card 1	1	2	3	4	5	6	7	8
Variable	BSID	LCID	SCALE	KBEND				
Type	I	I	F	I				
Default	none	none	1.0	0				

VARIABLE **DESCRIPTION**

BSID Beam set ID.

LCID Load curve ID defining preload force versus time. When the load curve ends or goes to zero, the initialization is assumed to be completed. See Remark 2 below.

SCALE Scale factor on load curve.

KBEND Bending stiffness flag

EQ.0: Bending stiffness is negligible since all integration points are assigned the same axial stress

EQ.1: Bending stiffness is retained by keeping the axial stress gradient

Remarks:

1. **Damping.** To achieve convergence during explicit dynamic relaxation, the application of the damping options is very important. If contact is active, contact damping is recommended with a value between 10-20 percent. Additional damping, via the option DAMPING_PART_STIFFNESS also speeds convergence where a coefficient of 0.10 is effective. If damping is not used, convergence may not be possible.

2. **Ramping.** When defining the load curve, LCID, a ramp starting at the origin should be used to increase the force to the desired value. The time duration of the ramp should produce a quasistatic response. When the end of the load

curve is reached, or when the value of the load decreases from its maximum value, the initialization stops. If the load curve begins at the desired force value, i.e., no ramp, convergence will take much longer, since the impulsive like load created by the initial force can excite nearly every frequency in the structural system where force is initialized.

*INITIAL_CONTACT_WEAR

Purpose: Initialize contact wear for simulation of wear processes, define as many cards as necessary.

Card 1	1	2	3	4	5	6	7	8
Variable	CID	NID	WDEPTH	NX	NY	NZ	ISEQ	NCYC
Type	I	I	F	F	F	F	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE **DESCRIPTION**

CID Contact Interface ID.

NID Node ID.

WDEPTH Wear depth, in units of length.

NX, NY, NZ Direction vector for wear, internally normalized.

ISEQ Simulation sequence number for the entire process.

NCYC The wear on this card will be processed NCYC times to modify the worn geometry. This is to say that *one* LS-DYNA simulation is used to predict the wear for *NCYC* repetitions of the process, in order to save simulation time. This number should be chosen with care, a negative number means that LS-DYNA will not apply this card, see remarks below.

Remarks:

This is a card that is not supposed to be manually inserted, but is automatically generated by LS-DYNA when simulating wear processes, see *CONTACT_ADD_-WEAR and parameters NCYC on *INTERFACE_SPRINGBACK_LSDYNA and SPR/MPR on *CONTACT. A sequence of identical simulations, except for perturbation of the geometry of certain components due to wear, is undertaken. For a given contact interface and node ID, the corresponding node is perturbed by the wear depth in the direction of wear. If the cycle number NCYC is negative, this means that the geometry has been already processed in LS-PrePost and the card is ignored by LS-DYNA, and if a node appears multiple times the wear from the individual sequences is accumulated.

*INITIAL_CRASHFRONT

Purpose: To define initial crashfront node set for materials supporting crashfront.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	STYPE						
Type	I	I						
Default	none	0						

VARIABLE	DESCRIPTION
SID	Crash front node set ID for initial crashfront nodes.
STYPE	ID type of SID: EQ.0: segment set ID, EQ.1: shell element set ID, EQ.2: part set ID, EQ.3: part ID, EQ.4: node set ID.

Remarks:

Material models 17, 54, 55, 58, 169, 261, and 262 reduce material strength in crashfront elements. This keyword defines the initial crashfront nodes, and all elements connected to these nodes are initialized as crashfront elements with reduced strength.

*INITIAL_DETONATION

Purpose: Define points to initiate the location of high explosive detonations in part ID's which use *MAT_HIGH_EXPLOSIVE_BURN (*MAT_008). Also see *CONTROL_EXPLOSIVE_SHADOW. If no *INITIAL_DETONATION is defined, detonation occurs in all the high explosive elements at time = 0.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	X	Y	Z	LT			
Type	I	F	F	F	F			
Default	all HE	0.	0.	0.	0.			

Accoustic Boundary Card. Additional card for PID = -1.

Card 2	1	2	3	4	5	6	7	8
Variable	PEAK	DECAY	XS	YS	ZS	NID		
Type	F	F	F	F	F	I		
Remark	1	1						

VARIABLE	DESCRIPTION
PID	Part ID of the high explosive to be lit, except in the case where the high explosive is modeled using an ALE formulation, in which case PID is the part ID of the mesh where the high explosive material to be lit initially resides. However, two other options are available: EQ.-1: an acoustic boundary, also, *BOUNDARY_USA_SURFACE, EQ.0: all high explosive materials are considered.
X	x-coordinate of detonation point, see Figure 23-1.
Y	y-coordinate of detonation point.

VARIABLE	DESCRIPTION
Z	z-coordinate of detonation point.
LT	Lighting time for detonation point. This time is ignored for an acoustic boundary.
PEAK	Peak pressure, p_0, of incident pressure pulse, see remark below.
DECAY	Decay constant, τ
XS	x-coordinate of standoff point, see Figure 23-1.
YS	y-coordinate of standoff point
ZS	z-coordinate of standoff point
NID	Reference node ID near structure

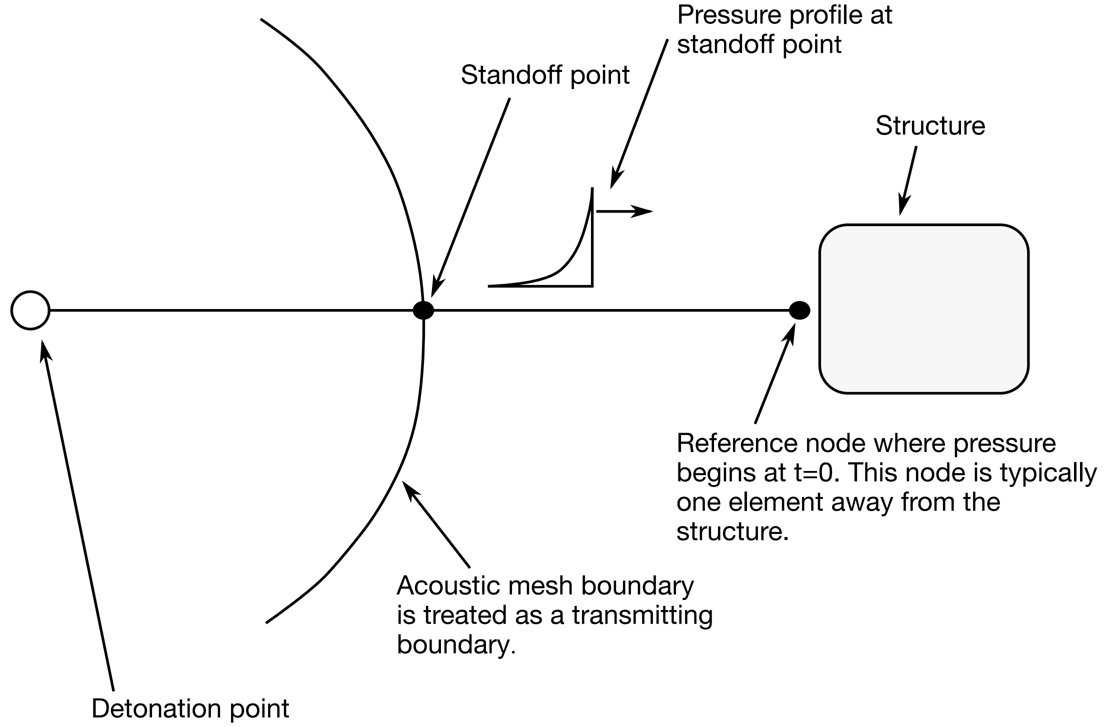

Pressure profile at standoff point
Standoff point
Structure
Reference node where pressure begins at t=0. This node is typically one element away from the structure.
Acoustic mesh boundary is treated as a transmitting boundary.
Detonation point

Figure 23-1. Initialization of the initial pressures due to an explosive disturbance is performed in the acoustic media. LS-DYNA automatically determines the acoustic mesh boundary and applies the pressure time history to the boundary. This option is only applicable to the acoustic element formulation, see *SECTION_SOLID.

Remarks:

For solid elements (not acoustic) two options are available. If the control card option, *CONTROL_EXPLOSIVE_SHADOW, is not used the lighting time for an explosive element is computed using the distance from the center of the element to the nearest detonation point, L_d; the detonation velocity, D; and the lighting time for the detonator, t_d:

$$t_L = t_d + \frac{L_d}{D}.$$

The detonation velocity for this default option is taken from the element whose lighting time is computed and does not account for the possibilities that the detonation wave may travel through other explosives with different detonation velocities or that the line of sight may pass outside of the explosive material.

If the control card option, *CONTROL_EXPLOSIVE_SHADOW, is defined, the lighting time is based on the shortest distance through the explosive material. If inert obstacles exist within the explosive material, the lighting time will account for the extra time required for the detonation wave to travel around the obstacles. The lighting times also automatically accounts for variations in the detonation velocity if different explosives are used. No additional input is required for this option but care must be taken when setting up the input. This option works for two and three-dimensional solid elements. It is recommended that for best results:

1. Keep the explosive mesh as uniform as possible with elements of roughly the same dimensions.

2. Inert obstacle such as wave shapers within the explosive must be somewhat larger than the characteristic element dimension for the automatic tracking to function properly. Generally, a factor of two should suffice. The characteristic element dimension is found by checking all explosive elements for the largest diagonal.

3. The detonation points should be either within or on the boundary of the explosive. Offset points may fail to initiate the explosive. When LT is nonzero, the detonation point is fixed to the explosive material at t = 0 and moves as the explosive material moves prior to detonation.

4. Check the computed lighting times in the post processor LS-PREPOST. The lighting times may be displayed at time = 0., state 1, by plotting component 7 (a component normally reserved for plastic strain) for the explosive material. The lighting times are stored as negative numbers. The negative lighting time is replaced by the burn fraction when the element ignites.

Line detonations may be approximated by using a sufficient number of detonation points to define the line. Too many detonation points may result in significant initialization cost.

The pressure versus time curve for the acoustic option is defined by:

$$p(t) = p_o e^{-\frac{t}{\tau}}.$$

*INITIAL_EOS_ALE

Purpose: This card initializes the pressure in ALE elements that have materials with *EOS.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	TYP	MMG	E0	V0	P0		
Type	I	I	I	F	F	F		
Default	none	none	none	0.0	0.0	0.0		

VARIABLE	DESCRIPTION
ID	Part ID or part set ID or element set ID.
TYP	Type of "ID": EQ.0: part set ID. EQ.1: part ID. EQ.2: element set ID (*SET_BEAM in 1D, *SET_SHELL in 2D, *SET_SOLID in 3D).
MMG	Specifies the multi-material group. GT.0: ALE multi-material group. LT.0: Set ID of ALE multi-material groups defined in *SET_-MULTI-MATERIAL_GROUP.
E0	Initial internal energy per reference volume unit (as defined in *EOS). See Remark 1.
V0	Initial relative volume (as defined in *EOS). See Remark 1.
P0	Initial pressure. See Remark 2.

Remarks:

1. **Initialization with Volume and Energy.** For most *EOS, E0 and V0 should be used to initialize the pressure. If only the internal energy is initialized, V0 should be 1.0 (If V0 = 0.0, E0 will not be applied).

2. **Initial Pressure with Derived Volume and Energy.** For *EOS_001, *EOS_004 and *EOS_006, the initial pressure P0 can be input directly. An iterative method will compute the initial internal energy and relative volume. This approach is applied if E0 = 0.0 and V0 = 0.0.

INITIAL_FATIGUE_DAMAGE_RATIO_*{OPTION}

Available options include:

<BLANK>

BINARY

Purpose: This card sets initial damage ratio for fatigue analysis. The initial damage ratio may come from the previous loading cases. The initial damage ratio can be defined by user directly, or can be extracted from existing binary database like D3FTG (using the option BINARY).

Card 1 for no option, <BLANK>.

Card 1	1	2	3	4	5	6	7	8
Variable	PID/SID	PTYP	DRATIO					
Type	I	I	F					
Default	none	0	0.0					

Card 1 for option BINARY.

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	C							
Default	d3ftg							

VARIABLE	DESCRIPTION
PID/SID	Part ID or part set ID for which the initial damage ratio is defined.
PTYP	Type of PID/SID: EQ.0: part ID. EQ.1: part set ID.

VARIABLE	DESCRIPTION
DRATIO	Initial damage ratio.
FILENAME	Path and name of existing binary database for fatigue information.

Remarks:

1. The card works for both time domain fatigue and frequency domain fatigue problems.
2. Card 1 can be repeated if the model has initial damage ratio coming from multiple loading cases.

*INITIAL_FIELD_SOLID

Purpose: This keyword is a simplified version of *INITIAL_STRESS_SOLID which can be used with hyperelastic materials. The keyword is used for history variable input. Data is usually in the form of the eigenvalues of diffusion tensor data. These are expressed in the global coordinate system. The input deck takes the following parameters:

> **NOTE:** As of LS-DYNA R5 in all contexts, other than *MAT_TISSUE_DISPERSED, this keyword is deprecated (and disabled). For all other materials this keyword has been superceded by *INITIAL_STRESS_SOLID.

Include as many pairs of cards 1 and 2 as necessary. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	EID	NINT	NHISV					
Type	I	I	I					
Default	none	none	0					

Card 2	1	2	3	4	5	6	7	8
Variable	FLD1	FLD2	FLD3	FLD4	FLD5	FLD6	FLD7	FLD8
Type	F	F	F	F	F	F	F	F
Default								

VARIABLE	DESCRIPTION
EID	Element ID
NINT	Number of integration points (should correspond to the solid element formulation).

VARIABLE	DESCRIPTION
NHISV	Number of field variables. If NHISV exceeds the number of integration point field variables required by the constitutive model, only the number required is output; therefore, if in doubt, set NHISV to a large number.
FLDn	Data for the nth field (history) variable. NOTE that *MAT_TISSUE_DISPERSED only use FLD1 to FLD3 since NHISV = 3.

Remarks:

Add as many cards as necessary. The keyword input ends when next keyword appears (next *). For example for two elements it can look as:

```
*INITIAL_FIELD_SOLID
$EID      NINT    NHISV
    1        1        3
$FLD1     FLD2     FLD3
  0.1      0.8      0.1
$EID      NINT    NHISV
    2        1        3
$FLD1     FLD2     FLD3
  0.3      0.2      0.5
```

INITIAL_FOAM_REFERENCE_GEOMETRY_*OPTION

Available options include:

RAMP

Purpose: The reference configuration allows stresses to be initialized (via REF in *MAT) in the following hyperelastic material models: 2, 5, 7, 21, 23, 27, 31, 38, 57, 73, 77, 83, 132, 179, 181, 183, and 189. Supported solid elements are the constant stress hexahedron (#1), the fully integrated S/R hexahedron (#2), the tetrahedron (#10), and the pentahedron (#15).

To use this option, the geometry of the foam material is defined in a deformed configuration. The stresses in the low density foam then depend only on the deformation gradient matrix F_{ij}:

$$F_{ij} = \frac{\partial x_i}{\partial X_j}$$

where x_i is the deformed configuration and X_j is the undeformed configuration. By using this option, dynamic relaxation can be avoided once a deformed configuration is obtained usually on the first run of a particular problem.

Optional RAMP Card. Additional optional card for the option of RAMP.

Card 1	1	2	3	4	5	6	7	8
Variable	NDTRRG							
Type	I							
Default	0							

Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8	9	10
Variable	NID	X		Y		Z				
Type	I	F		F		F				
Default	none	0.		0.		0.				
Remarks										

VARIABLE	DESCRIPTION
NDTRRG	Number of time steps taken for an element to restore its reference geometry. Definition of NDTRRG allows an element to ramp up to its reference shape in NDTRRG time steps. Currently ls-dyna uses only one NDTRRG and applies it to all foam materials with reference geometries. If more than one NDTRRG is defined, the latter defined one will replace the previously define one.
NID	Node number
X	*x* coordinate in reference configuration
Y	*y* coordinate in reference configuration
Z	*z* coordinate in reference configuration

*INITIAL_GAS_MIXTURE

Purpose: This command is used to specify (a) which ALE multi-material groups may be present inside an ALE mesh set at time zero, and (b) the corresponding reference gas temperature and density which define the initial thermodynamic state of the gases. The order of the species in the gas mixture corresponds to the order of different gas species defined in the associated *MAT_GAS_MIXTURE card. This card must be used together with a *MAT_GAS_MIXTURE (or equivalently, a *MAT_ALE_GAS_MIXTURE) card.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	STYPE	MMGID	TEMP				
Type	I	I	I	F				
Default	none	0	none	none				

Card 2	1	2	3	4	5	6	7	8
Variable	R01	R02	R03	R04	R05	R06	R07	R08
Type	F	F	F	F	F	F	F	F
Default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

VARIABLE	DESCRIPTION
SID	Set ID for initialization. This SID defines the ALE mesh within which certain ALE multi-material group(s) may be present at $t = 0$.
STYPE	Set type for the SID above: EQ.0: SID is a part set ID EQ.1: SID is a part ID
MMGID	ALE Multi-material group ID of the material that may be present at t = 0 in the ALE mesh set defined by SID.

VARIABLE **DESCRIPTION**

TEMP Initial static temperature of the gas species occupying the ALE mesh. Note that all species in the mixture are assumed to be in thermal equilibrium (having the same T).

RO1-RO8 Initial densities of the ALE material(s) which may be occupying some region (or all) of the aforementioned ALE mesh, for up to eight different gas species. The order of the density input corresponds to the order of the materials defined in associated *MAT_GAS_MIXTURE card.

Remarks:

1. Please see the example under the *MAT_GAS_MIXTURE card definition for an application of the *INITIAL_GAS_MIXTURE card.

2. The temperature is assumed to be the initial temperature which together with the gas density, will define the initial pressure of the gas species via the perfect gas law,

$$P|_{t=0} = \rho|_{t=0}(C_P - C_V)T|_{t=0}.$$

   The user should manually check the initial pressure for consistency.

3. Given an ALE mesh, this mesh may initially be occupied by one or more ALE multi-material groups (AMMG). For example, a background ALE mesh (H1) containing AMMG 1 may be partially filled with AMMG 2 via the volume filling command *INITIAL_VOLUME_FRACTION_GEOMETRY. Then there are 2 AMMGs to be initialized for this mesh H1. The commands look like the following.

```
$----------------------------------------------------------------------------
$ One card is defined for each AMMG that will occupy some elements of a mesh
set
*INITIAL_GAS_MIXTURE
$      SID     STYPE     MMGID        T0
         1         1         1    298.15
$     RHO1      RHO2      RHO3      RHO4      RHO5      RHO6      RHO7
RHO8
    1.0E-9
*INITIAL_GAS_MIXTURE
$      SID     STYPE     MMGID        T0
         1         1         2    298.15
$     RHO1      RHO2      RHO3      RHO4      RHO5      RHO6      RHO7
RHO8
    1.2E-9
$----------------------------------------------------------------------------
```

*INITIAL_HYDROSTATIC_ALE

Purpose: When an ALE model contains one or more regular (not reservoir-type) ALE parts (ELFORM = 11 and AET = 0), this command may be used to initialize the hydrostatic pressure field in the regular ALE domain due to gravity. The *LOAD_-BODY_(OPTION) keyword must be defined.

Card 1	1	2	3	4	5	6	7	8
Variable	ALESID	STYPE	VECID	GRAV	PBASE			
Type	I	I	I	I	I			
Default	none	0	none	0	0			

Multi-material Layers Group Cards. Repeat card 2 as many times as the number of AMMG layers present in the model.

Card 2	1	2	3	4	5	6	7	8
Variable	NID	MMGBLO						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
ALESID	ALESID is a set ID defining the ALE domain/mesh whose hydrostatic pressure field due to gravity is being initialized by this keyword. See Remark 2 and 4.
STYPE	ALESID set type. See Remark 4. EQ.0: Part set ID (PSID), EQ.1: Part ID (PID), EQ.2: Solid set ID (SSID).
VECID	Vector ID of a vector defining the direction of gravity.

VARIABLE **DESCRIPTION**

GRAV Magnitude of the Gravitational acceleration. For example, in metric units the value is usually set to 9.80665 m/s^2.

PBASE Nominal or reference pressure at the top surface of all fluid layers. By convention, the gravity direction points from the top layer to the bottom layer. Each fluid layer must be represented by an ALE multi-material group ID (AMMGID or MMG). See Remark 1.

NID Node ID defining the top of an ALE fluid (AMMG) layer.

MMGBLO AMMG ID of the fluid layer immediately below this NID. Each node is defined in association with one AMMG layer below it. See Remark 3.

Remarks:

1. **Pressure in Multi-Layer Fluids.** For models using multi-layer ALE Fluids the pressure at the top surface of the top fluid layer is set to PBASE and the hydrostatic pressure is computed as following

$$P = P_{\text{base}} + \sum_{i=1}^{N_{\text{layers}}} \rho_i g h_i \ .$$

2. **Limitations on Element Formulation.** The keyword applies only to the regular ALE parts with ELFORM = 11 and AET = 0 on the *SECTION_SOLID and *SECTION_ALE2D cards (not reservoir-type). This keyword cannot be used to initialize reservoir-type ALE parts (AET = 4). Also, ramping functions are not supported, so the loading is done in one step at $t = 0$. For initializing reservoir-type ALE domain, please review the *ALE_AMBIENT_HYDROSTATIC keyword.

3. **Limitation on EOS Model.** The keyword only supports *EOS_GRUNEISEN and *EOS_LINEAR_POLYNOMIAL, but only inthe following two cases,

$$c_3 = c_4 = c_5 = c_6 = 0, \qquad E_0 = 0$$

$$c_4 = c_5 > 0, \qquad c_1 = c_2 = c_3 = c_6 = 0, \qquad V_0 = 0.$$

4. **Structured ALE usage.** When used with structured ALE, PART and PART set options might not make too much sense. This is because all elements inside a structured ALE mesh are assigned to one single PART ID. If we want to prescribe initial hydrostatic pressure for all the elements inside the structured mesh, we can certainly use that PART ID. But if we only want to do that to some elements, we have to generate a solid set which contains those structured

ALE elements. It is done by using the *SET_SOLID_GENERAL keyword with SALECPT option. And then use STTYPE=2 (solid element set ID) option.

Example:

Model Summary: Consider a model consisting of 2 ALE parts, air on top of water.
H1 = AMMG1 = Air part above.
H2 = AMMG2 = Water part below.

```
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
$ (non-ambient) ALE materials (fluids) listed from top to bottom:
$
$ NID AT TOP OF A LAYER SURFACE          ALE MATERIAL LAYER BELOW THIS NODE
$ TOP OF 1st LAYER -------> 1722         ----------------------------------------
$                                        Air above   = PID 1 = H1 = AMMG1 (AET=0)
$ TOP OF 2nd LAYER -------> 1712         ----------------------------------------
$                                        Water below = PID 2 = H2 = AMMG2 (AET=0)
$ BOTTOM ------------------------------------------------------------------------
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
*INITIAL_HYDROSTATIC_ALE
$   ALESID     STYPE     VECID      GRAV     PBASE
        12         0        11   9.80665  101325.0
$      NID    MMGBLO
      1722         1
      1712         2
*SET_PART_LIST
        12
         1         2
*ALE_MULTI-MATERIAL_GROUP
         1         1
         2         1
*DEFINE_VECTOR
$      VID        XT        YT        ZT        XH        YH        ZH       CID
        11       0.0       1.0       0.0       0.0       0.0       0.0
*DEFINE_CURVE
         9
               0.000               0.000
               0.001               1.000
              10.000               1.000
*LOAD_BODY_Y
$     LCID        SF    LCIDDR        XC        YC        ZC
         9   9.80665         0       0.0       0.0       0.0
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
```

*INITIAL_IMPULSE_MINE

Purpose: Apply initial velocities to the nodes of a 3D structure due to the impulse imparted by the detonation of a buried land mine. This feature is based on the empirical model developed by [Tremblay 1998].

Card 1	1	2	3	4	5	6	7	8
Variable	SSID	M	RHOS	DEPTH	AREA	SCALE	not used	UNIT
Type	I	F	F	F	F	F		I
Default	none	0.0	0.0	0.0	0.0	1.0		1
Remarks	1	2						

Either set a heading or delete this row.

Card 1	1	2	3	4	5	6	7	8
Variable	X	Y	Z	NIDMC	GVID	TBIRTH	PSID	SEARCH
Type	F	F	F	I	I	F	I	F
Default	0.0	0.0	0.0	0	none	0.0	0	0.0

VARIABLE	DESCRIPTION
SSID	Segment set ID (see *SET_SEGMENT and Remark 1)
MTNT	Equivalent mass of TNT (See Remark 2).
RHOS	Density of overburden soil.
DEPTH	Burial depth from the ground surface to the center of the mine. This value must be a positive.
AREA	Cross sectional area of the mine.
SCALE	Scale factor for the impulse.

VARIABLE	DESCRIPTION
UNIT	Unit system. This must match the units used by finite element model. EQ.1: inch, dozen slugs (i.e., lbf × s^2/in), second, PSI (default) EQ.2: meter, kilogram, second, Pascal EQ.3: centimeter, gram, microsecond, megabar EQ.4: millimeter, kilogram, millisecond, GPa EQ.5: millimeter, metric ton, second, MPa EQ.6: millimeter, gram, millisecond, MPa
X, Y, Z	x-, y-, and z- coordinates of mine center.
NIDMC	Optional node ID representing the mine center (see *NODE). If defined then X, Y and Z are ignored.
GVID	Vector ID representing the vertically upward direction, i.e., normal to the ground surface (see *DEFINE_VECTOR).
TBIRTH	Birth time. Impulse is activated at this time.
PSID	Part set ID identifying the parts affected by the mine (see *SET_PART). If zero it defaults to the part comprised by the nodes of the segment set.
SEARCH	Limit the search depth into the structure. Initial nodal velocity is distributed from the segment to a depth equal to the SEARCH value. The value must be positive. If set to zero the search depth is unlimited and extends through the part(s) identified by PSID.

Remarks:

1. Segment normals should nominally point toward the mine.

2. The segments should belong to 3D thin shell, solid, or thick shell elements. This feature cannot be applied to 2D geometries.

3. Several methods can be used to approximate the equivalent mass of TNT for a given explosive. One method involves scaling the mass by the ratio of the squares of the Chapman-Jouguet detonation velocities given by the relationship.

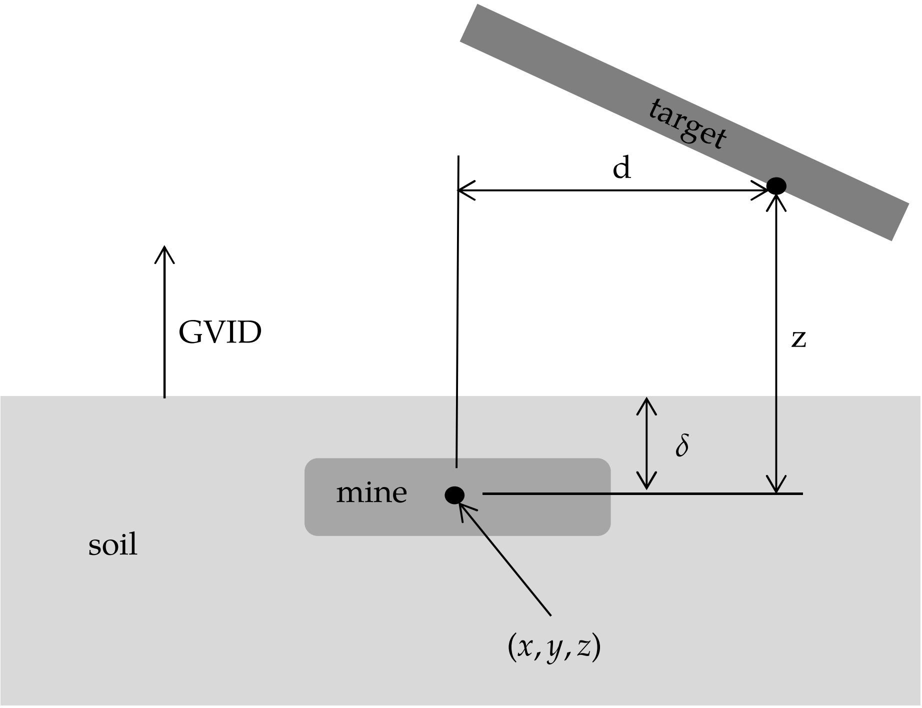


Figure 23-2. Schematic of the buried mine parameters.

$$M_{\mathrm{TNT}} = M\frac{(\mathrm{DCJ})^2}{(\mathrm{DCJ})^2_{\mathrm{TNT}}}$$

where M_{TNT} is the equivalent TNT mass and $(\mathrm{DCJ})^2_{\mathrm{TNT}}$is the Chapman-Jouguet detonation velocity of TNT. M and DCJ are, respectively, the mass and C-J velocity of the explosive under investigation. "Standard" TNT is considered to be cast with a density of 1.57gm/cm^3 and $(\mathrm{DCJ})_{\mathrm{TNT}} = 0.693$ cm/μsec.

4. This implementation assumes the energy release (heat of detonation) for 1 kilogram of TNT is 4.516 MJ.

5. Prediction of the impulse relies on an empirical approach which involves fitting curves to experimental results. The upper error bound is 1.8 times the predicted value and the lower is predicted value divided by 1.8. Thus, if the predicted impulse is 10 kN-seconds then the solution space ranges from 5.6 kN-sec to 18 kN-sec.

6. The computed impulse is valid when the following criteria are met.

$$0.106 \le \frac{\delta}{z} \le 1$$

$$6.35 \le \frac{E/A}{\rho c^2 z} \le 150$$

$$0.154 \le \frac{\sqrt{A}}{z} \le 4.48$$

$$0 \le \frac{d}{z} \le 19.3$$

where,

δ = the distance from the mine center to the ground surface (DEPTH)

z = the vertical distance from the mine center to the target point

E = the energy release of the explosive

A = the cross-sectional area of the mine (AREA)

ρ = the soil density (RHOS)

c = the wave speed in the soil

d = the lateral distance from the mine center to the target point.

See Figure 23-2.

References:

Tremblay, J.E., “Impulse on Blast Deflectors from a Landmine Explosion,” DRDC Valcartier, DREV-TM-9814, (1998).

INITIAL_INTERNAL_DOF_SOLID_*OPTION

Available Options Include:

TYPE3

TYPE4

Purpose: Initialize the internal degrees of freedom for solid element types 3 and 4.

Card 1	1	2	3	4	5	6	7	8
Variable	LID							
Type	I							
Default	none							

Value Cards. Include 12 cards TYPE3 and 6 cards for TYPE4.

Card	1	2	3	4	5	6	7	8
Variable	VALX	VALY	VALZ					
Type	F	F	F					
Default	none	none	none					

VARIABLE	DESCRIPTION
LID	Element ID.
VALX	*x* component of internal degree of freedom.
VALY	*y* component of internal degree of freedom.
VALZ	*z* component of internal degree of freedom.

Remarks:

1. The internal degrees of freedom are specified in terms of the displacements of the corresponding mid-side nodes of the 20 node hex and the 10 node tetrahedron that are the basis of the type 3 and 4 solid elements, respectively.

INITIAL_LAG_MAPPING_*{OPTION}

The available options are

<BLANK>

WRITE

Purpose: This card initializes a 3D Lagrangian calculation with data from the last cycle of a preceding 2D or 3D Lagrangian calculation.

In its *INITIAL_LAG_MAPPING form (<BLANK> option), this keyword causes data to be read in from a *mapping file*; while, with the WRITE active this card is used to set which parts are written to the mapping file. The mapping file's filename is specified using the "lagmap=" command line argument (see Remarks 1 and 2).

Card 1	1	2	3	4	5	6	7	8
Variable	SETID							
Type	I							
Default	none							

Card 2. Additional card for <BLANK> keyword option.

Card 2	1	2	3	4	5	6	7	8
Variable	XP	YP	ZP	VECID	ANGLE	NELANGL		
Type	F	F	F	I	F	I		
Default	0.0	0.0	0.0	none	none	0		

VARIABLE	DESCRIPTION
SETID	part set ID. See Remarks 3 and 4.
XP	*x*-position of a point belonging to the plane from which the 3D mesh is generated (only for a 2D to 3D mapping). See Remark 5.

VARIABLE **DESCRIPTION**

YP y-position of a point belonging to the plane from which the 3D mesh is generated (only for a 2D to 3D mapping). See Remark 5.

ZP z-position of a point belonging to the plane from which the 3D mesh is generated (only for a 2D to 3D mapping). See Remark 5.

VECID ID of the rotation axis (symmetric axis for a 2D to 3D mapping) defined by *DEFINE_VECTOR. See Remark 5.

ANGLE Angle of rotation around an axis defined by *DEFINE_VECTOR. See Remark 5.

NELANGL Number of elements to create in the azimuthal direction for ANGLE (only for a 2D to 3D mapping). See Remark 5.

Remarks:

1. **The Mapping File as Output.** In the absence of a *INITIAL_LAG_MAPPING card, adding a "lagmap=" argument to the command line will cause LS-DYNA to write a mapping file. This file contains the following nodal and element data:
   - nodal coordinates (initial and last steps)
   - nodal velocities
   - nodal temperatures (if *CONTROL_THERMAL_SOLVER is used)
   - part ids
   - element connectivities
   - element centers
   - densities
   - volume fractions
   - stresses
   - plastic strains
   - internal energies
   - bulk viscosities
   - relative volumes

2. **The Mapping File as Input.** If the keyword INITIAL_LAG_MAPPING is in the input deck and the "lagmap=" argument is in the command line, then La-

grangian data is read from the mapping file defined by "lagmap=" to initialize the run.

3. **Part Sets (Write).** The part set, SETID, defines which parts are involved in the mapping. The WRITE option can be used to write data in the mapping file for ONLY the parts specified by the set. If the keyword INITIAL_LAG_MAPPING_WRITE is not included in the input deck then ALL Lagrangian parts are written in the mapping file during the last cycle. Similarly, for reading

4. **Part Sets (Read).** The mapping initializes the data for every node and element defined by SETID within the domain swept by the 2D mesh or the region initially occupied by the previous 3D mesh. For nodes and elements outside of SETID it has no effect.

5. **Embedding.** The first point in the definition of the rotation axis VECID specifies the origin location for the previous run in the current 3D space. The 2D to 3D mapping depends on whether or not a 3D mesh has already been defined. The 3D to 3D mapping needs a pre-existing mesh.

   a) *No Mesh Case for a 2D to 3D mapping*. If there is no 3D mesh (no solid and shell with parts in SETID), the point (XP, YP, ZP) together with the symmetry axis (VECID) are used to generate a mesh. The point defines the plane in which the 2D is embedded. The 3D mesh is generated by rotating the 2D mesh around the axis. The point (XP, YP, ZP) must not be on the symmetry axis. ANGLE defines the angle of rotation in degrees. The rotation is counterclockwise when viewed from the axis head. NELANGL is the number of elements to generate in the azimuthal direction.

   b) *Pre-existing Mesh Case for a 2D to 3D mapping.* If there is a 3D mesh (solids or shells with parts in SETID), the nodes should be within the domain swept by the initial positions of the 2D mesh. Then, the nodes are mapped to new locations based on the last mesh positions of the previous run.

   c) *Pre-existing Mesh Case for a 3D to 3D mapping.* If there is a 3D mesh (solids or shells with parts in SETID), the nodes should be in the region initially occupied by the previous 3D mesh. Then, the nodes are mapped to new locations based on the last mesh positions of the previous run. VECID is an axis of rotation of the pre-existing mesh if ANGLE is defined. If this latter is not defined, the first point in VECID is still the previous origin location and it can be used to translate the pre-existing mesh

*INITIAL_MOMENTUM

Purpose: Define initial momentum to be deposited in solid elements. This option is to crudely simulate an impulsive type of loading.

Card	1	2	3	4	5	6	7	8
Variable	EID	MX	MY	MZ	DEPT			
Type	I	F	F	F	F			
Default	none	0.	0.	0.	0,			

VARIABLE	DESCRIPTION
EID	Element ID
MX	Initial *x*-momentum
MY	Initial *y*-momentum
MZ	Initial *z*-momentum
DEPT	Deposition time

INITIAL_PWP_DEPTH_*{OPTION}

The available options include:

<BLANK>

SET

Purpose: Initialize pore water pressure in solid elements where a non-hydrostatic profile is required.

Card 1	1	2	3	4	5	6	7	8
Variable	PID/PSID	LC						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
PID	Part ID or Part Set ID for the_SET option
LC	Curve of pore water pressure head (length units) vs *z*-coordinate

Remarks:

This feature overrides the automatically calculated hydrostatic pressure profile. The points in the curve must be ordered with the most negative z-coordinate first – this order looks "upside-down" on the page.

If a part has pore fluid but no *INITIAL_PWP_DEPTH is defined, the default initial pressure profile is hydrostatic.

INITIAL_STRAIN_SHELL_*{OPTION}

The available options include:

<BLANK>

SET

Purpose: Initialize strain tensor for shell element. This option is primarily for multi-stage metal forming operations where the accumulated strain is of interest.

These strain tensors are defined at the inner and outer integration points and are used for post-processing only. There is no interpolation with this option and the strains are defined in the global Cartesian coordinate system. The *DATABASE_EXTENT_BINARY flag STRFLG must be set to unity for this option to work. When OPTION is blank, users have the option to define strains at all integration points by providing nonzero NPLANE, NTHICK and setting INTOUT flag of *DATABASE_EXTENT_BINARY to either "STRAIN" or "ALL".

Card Sets. Define as many shell elements in this section as desired, *one set of cards per element*. The input is assumed to terminate when a new keyword is detected.

Card 1	1	2	3	4	5	6	7	8
Variable	EID	NPLANE	NTHICK					
Type	I	I	I					
Default	none	none	none					

When NPLANE and NTHICK are defined, include NPLANE × NTHICK cards below. For each through thickness point define NPLANE points. NPLANE should be either 1 or 4 corresponding to either 1 or 4 Gauss integration points. If four integration points are specified, they should be ordered such that their in plane parametric coordinates are at:

$$\left(-\frac{\sqrt{3}}{3},-\frac{\sqrt{3}}{3}\right),\quad \left(\frac{\sqrt{3}}{3},-\frac{\sqrt{3}}{3}\right),\quad \left(\frac{\sqrt{3}}{3},\frac{\sqrt{3}}{3}\right),\quad \left(-\frac{\sqrt{3}}{3},\frac{\sqrt{3}}{3}\right)$$

respectively.

Strain Cards. When NPLANE and NTHICK are not defined or when the **SET** option is used, define two cards below, one for the inner integration point and the other for the outer integration point, respectively.

Card 2	1	2	3	4	5	6	7	8
Variable	EPSxx	EPSyy	EPSzz	EPSxy	EPSyz	EPSzx	T	
Type	F	F	F	F	F	F	F	
Default	none	none	none	0			0.	

VARIABLE	DESCRIPTION
EID	Element ID or shell element set ID when the **SET** option is used.
NPLANE	Number of in-plane integration points being output (not read when the **SET** option is used).
NTHICK	Number of integration points through the thickness (not read when the **SET** option is used).
EPS*ij*	Define the *ij* strain component. The strains are defined in the GLOBAL Cartesian system.
T	Parametric coordinate of through thickness integration point between -1 and 1 inclusive.

*INITIAL_STRAIN_SOLID_{OPTION}

The availables options include:

<BLANK>

SET

Purpose: Initialize strain tensor at element center. This option can be used for multi-stage metal forming operations where the accumulated strain is of interest.

These strain tensors are defined at the element center and are used for post-processing only. The strains are defined in the global Cartesian coordinate system. The *DATABASE_EXTENT_BINARY flag STRFLG must be set to unity for this option to work. This capability is not available for the cohesive element since it is based on displacements, not strains.

Card Sets. Define as many solid elements in this section as desired: *one pair of cards per element*. The input is assumed to terminate when a new keyword is detected.

Element ID Cards.

Card 1	1	2	3	4	5	6	7	8
Variable	EID							
Type	I							
Default	none							

Strain Cards.

Card 2	1	2	3	4	5	6	7	8
Variable	EPSxx	EPSyy	EPSzz	EPSxy	EPSyz	EPSzx		
Type	F	F	F	F	F	F		

VARIABLE	DESCRIPTION
EID	Element ID or solid element set ID when the **SET** option is used.
EPS*ij*	Define the *ij* strain component. The strains are defined in the GLOBAL cartesian system.

*INITIAL_STRAIN_TSHELL

Purpose: Initialize the strain tensors for thick shell elements..

Strain tensors are defined at the inner and outer integration points and are used for post-processing only. Strain tensors are defined in the global Cartesian coordinate system. The STRFLG flag on *DATABASE_EXTENT_BINARY must be set to unity for this option to work. Initialize as many elements as needed.

Card Sets. For each element, include a set of cards 1, 2, and 3, where card 2 is for the inner layer and card 3 is for the outer layer. The input is assumed to terminate when a new keyword is detected

Card 1	1	2	3	4	5	6	7	8
Variable	EID							
Type	I							
Default	none							

Strain Cards. Card 2 is the strain at the inner layer. Card 3 is the strain at the outer layer.

Cards 2, 3	1	2	3	4	5	6	7	8
Variable	EPSxx	EPSyy	EPSzz	EPSxy	EPSyz	EPSzx		
Type	F	F	F	F	F	F		
Default	0	0	0	0	0	0	.	

VARIABLE	DESCRIPTION
EID	Element ID.
EPS*ij*	Define the *ij* strain component. The strains are defined in the GLOBAL Cartesian system.

*INITIAL_STRESS_BEAM

Purpose: Initialize stresses, plastic strains and history variables for Hughes-Liu beam elements, or the axial force, moment resultants and history variables for Belytschko-Schwer beam elements.

Card Sets. Define as many beams in this section as desired. Each set consists of one Card 1 and several additional cards depending on variables NPTS, LARGE, NHISV, and NAXES. The input terminates when a new keyword ("*") card is detected.

Card 1	1	2	3	4	5	6	7	8
Variable	EID	RULE	NPTS	LOCAL	LARGE	NHISV	NAXES	
Type	I	I	I	I	I	I	I	
Default	none	none	none	0	0	0	0	

Belytschko-Schwer Card for LARGE = 0. Additional card for the Belytschko-Schwer beam.

Card 2	1	2	3	4	5	6	7	8
Variable	F11	T11	M12	M13	M22	M23	PARM	
Type	F	F	F	F	F	F	F	

Belytschko-Schwer Cards for LARGE = 1. Additional cards for the Belytschko-Schwer beam. Include as many cards as necessary to collect NHISV (see Card 1) history variables.

Card 2	1	2	3	4	5	6	7	8	9	10
Variable	F11		T11		M12		M13		M22	
Type	F		F		F		F		F	

.

Card 3	1	2	3	4	5	6	7	8	9	10
Variable	M23		PARM		HISV1		HISV2		HISV3	
Type	F		F		F		F		F	

Hughes-Liu Cards for LARGE = 0. Additional cards for the Hughes-Liu beam. Include NPTS additional cards, one per integration point.

Card 2	1	2	3	4	5	6	7	8
Variable	SIG11	SIG22	SIG33	SIG12	SIG23	SIG31	EPS	
Type	F	F	F	F	F	F	F	

Hughes-Liu Cards for LARGE = 1. Additional cards for the Hughes-Liu beam. Include NPTS additional card sets, one per integration point. Include as many cards in one card set as necessary to collect NHISV (see Card 1) history variables.

Card 2	1	2	3	4	5	6	7	8	9	10
Variable	SIG11		SIG22		SIG33		SIG12		SIG23	
Type	F		F		F		F		F	

.

Card 3	1	2	3	4	5	6	7	8	9	10
Variable	SIG31		EPS		HISV1		HISV2		HISV3	
Type	F		F		F		F		F	

Optional Local Axes Cards for NAXES = 12. Additional cards for definition of local axes values. These 12 values are internally used by LS-DYNA for the mapping between local beam element system and global coordinate system. They are automatically written to the dynain file if *INTERFACE_SPRINGBACK_LSDYNA or *CONTROL_-STAGED_CONSTRUCTION is used.

Card 4	1	2	3	4	5	6	7	8	9	10
Variable	AX1		AX2		AX3		AX4		AX5	
Type	F		F		F		F		F	

.

Card 5	1	2	3	4	5	6	7	8	9	10
Variable	AX6		AX7		AX8		AX9		AX10	
Type	F		F		F		F		F	

.

Card 6	1	2	3	4	5	6	7	8	9	10
Variable	AX11		AX12							
Type	F		F							

VARIABLE **DESCRIPTION**

EID Element ID

RULE Integration rule type number:

EQ.1.0: 1 × 1 Gauss quadrature,

EQ.2.0: 2 × 2 Gauss quadrature (default beam),

EQ.3.0: 3 × 3 Gauss quadrature,

EQ.4.0: 3 × 3 Lobatto quadrature,

EQ.5.0: 4 × 4 Gauss quadrature.

NPTS Number of integration points. For the Belytschko-Schwer resultant beam element, NPTS = 1.

VARIABLE	DESCRIPTION
LOCAL	Coordinate system for stresses: EQ.0: Stress components are defined in the global coordinate system. EQ.1: Stress components are defined in the local beam system. In the local system components SIG22, SIG33, and SIG23 are set to 0.0.
LARGE	Format size: EQ.0: off, EQ.1: on. Each field is twice as long for higher precision.
NHISV	Number of additional history variables. Only available for LARGE = 1.
NAXES	Number of variables giving beam local axes (0 or 12)
F11	Axial force resultant along local beam axis 1
T11	Torsional moment resultant about local beam axis 1
M12	Moment resultant at node 1 about local beam axis 2
M13	Moment resultant at node 1 about local beam axis 3
M22	Moment resultant at node 2 about local beam axis 2
M23	Moment resultant at node 2 about local beam axis 3
PARM	Generally not used.
SIG*ij*	Define the *ij* stress component
EPS	Effective plastic strain
HISV*n*	Define the *n*th history variable
AX*n*	The *n*th local axes value

INITIAL_STRESS_DEPTH_*{OPTION}

Available options include:

<BLANK>

SET

Purpose: Initialize solid element stresses where stress is a function of depth.

Card 1	1	2	3	4	5	6	7	8
Variable	PID/PSID	RO_G	ZDATUM	KFACT	LC	LCH	LCK0	
Type	I	F	F	F	I	I	I	
Default	none	none	none	0.0	none	none	none	

VARIABLE	DESCRIPTION
PID/PSID	Part ID or Part Set ID for the SET option
RO_G	Stress per unit elevation above datum, which is usually ρ_g = density × gravity.
ZDATUM	*z*-coordinate of datum
KFACT	*x* and *y*-stresses = KFACT × *z*-stress
LC	Optional curve of stress vs z-coordinate (ZDATUM is ignored with this option)
LCH	Optional curve of horizontal stress versus z-coordinate (KFACT is ignored with this option)
LVK0	Optional curve of K0 (ratio of horizontal_stress/vertical_stress) versus *z*-coordinate. KFACT and LCH are ignored with this option. The *x*-axis of the curve is the *z*-coordinate, the *y*-axis is K0.

Remarks:

With this keyword stress is calculated according to,

$$\sigma_z = \mathrm{RO_G} \times (\mathrm{Z_{element}} - \mathrm{ZDATUM}).$$

To generate compressive stresses, the datum should be above the highest element. For instance, this is usually at the surface of the soil in geotechnics simulations. If the curve is present, it overrides RO_G and ZDATUM. Note that the points in the curve should be ordered with most negative *z*-coordinate first.

First, select how the vertical stress as a function of *z*-coordinate will be defined (either RO_G and ZDATUM, or LC). Next, select how the horizontal stress will be defined (either a constant factor KFACT times the vertical stress; or a factor that varies with *z*-coordinate times the vertical stress using LCK0; or a curve of horizontal stress versus depth LCH).

If pore water is present, the stresses input here are effective (soil skeleton stresses only). The pore water pressures will automatically be initialized to hydrostatic, or by *INITIAL_PWP_DEPTH or *BOUNDARY_PWP_TABLE if those cards are present.

For a 2D problem (axisymmetric or plane strain), replace *z* in above documentation with *y*.

*INITIAL_STRESS_SECTION

Purpose: Initialize the stress in solid elements that are included in a section definition (*DATABASE_CROSS_SECTION_option) to create a preload. The stress component in the direction normal to the cross-section plane is prescribed according to a curve. This option works with a subset of materials that are incrementally updated including the elastic, viscoelastic, and elastoplastic materials. Rubbers, foams, and materials that are combined with equations-of-state cannot be initialized by this approach, except as noted in Remark #3.

Card 1	1	2	3	4	5	6	7	8
Variable	ISSID	CSID	LCID	PSID	VID	IZSHEAR		
Type	I	I	I	I	I	I		
Default	none	none	none	none	none	0		

VARIABLE	DESCRIPTION
ISSID	Section stress initialization ID.
CSID	Cross-section ID. See *DATABASE_CROSS_SECTION.
LCID	Load curve ID defining preload stress versus time. When the load curve ends or goes to zero, the initialization is assumed to be completed. See Remark 2.
PSID	Part set ID.
VID	Vector ID defining the direction normal to the cross section. This vector must be defined if *DATABASE_CROSS_SECTION_SET is used to define the cross section. If the cross section is defined using the PLANE option, the normal used in the definition of the plane is used if VID is left undefined.
IZSHEAR	Shear stress flag: EQ.0: Shear stresses are prescribed as zero during the time the curve is acting to prescribe normal stress. EQ.1: Shear stresses are allowed to develop during the time the curve is acting to prescribe normal stress.

Remarks:

1. To achieve convergence during explicit dynamic relaxation, the application of the damping options is very important. If contact is active, contact damping is recommended with a value between 10-20 percent. Additional damping, via the option DAMPING_PART_STIFFNESS also speeds convergence where a coefficient of 0.10 is effective. If damping is not used, convergence may not be possible.

2. When defining the load curve, LCID, a ramp starting at the origin should be used to increase the stress to the desired value. The time duration of the ramp should produce a quasi-static response. When the end of the load curve is reached, or when the value of the load decreases from its maximum value, the initialization stops. If the load curve begins at the desired stress value, i.e., no ramp, convergence will take much longer, since the impulsive like load created by the initial stress can excite nearly every frequency in the structural system where stress is initialized.

3. This option currently applies only to materials that are incrementally updated. Hyperelastic materials and materials that require an equation-of-state are not currently supported. However, materials 57, 73, and 83 can be initialized with this approach.

4. Solid elements types 1, 2, 3, 4, 9, 10, 13, 15, 16, 17, and 18 are supported. ALE elements are not supported.

INITIAL_STRESS_SHELL_*{OPTION}

Available options include:

<BLANK>

SET

Purpose: Initialize stresses, history variables, and the effective plastic strain for shell elements. Materials that do not use an incremental formulation for the stress update may not be initializable with this card.

Card Sets per Element. Define as many shell elements or shell element sets in this section as desired. The input is assumed to terminate when a new keyword ("*") card is detected.

Element Card.

Card 1	1	2	3	4	5	6	7	8
Variable	EID/SID	NPLANE	NTHICK	NHISV	NTENSR	LARGE	NTHINT	NTHHSV
Type	I	I	I	I	I	I	I	I
Default	none	none	none	0	0	0	0	0

Ordering of Integration Points.

For each through thickness point define NPLANE points. NPLANE should be either 1 or 4 corresponding to either 1 or 4 Gauss integration points. If four integration points are specified, they should be ordered such that their in plane parametric coordinates are at:

$$\left(-\frac{\sqrt{3}}{3},-\frac{\sqrt{3}}{3}\right),\quad \left(\frac{\sqrt{3}}{3},-\frac{\sqrt{3}}{3}\right),\quad \left(\frac{\sqrt{3}}{3},\frac{\sqrt{3}}{3}\right),\quad \left(-\frac{\sqrt{3}}{3},\frac{\sqrt{3}}{3}\right),$$

respectively. It is not necessary for the location of the through thickness integration points to match those used in the elements which are initialized. The data will be interpolated by LS DYNA.

Solid Mechanics Data Card for LARGE = 0.

The following set of cards: "Stress Card" through "Tensor Cards." Should be included NPLANE × NTHICK times (one set for each integration point).

Stress Card. Additional Card for LARGE = 0.

Card 2	1	2	3	4	5	6	7	8
Variable	T	SIGXX	SIGYY	SIGZZ	SIGXY	SIGYZ	SIGZX	EPS
Type	F	F	F	F	F	F	F	F

History Variable Cards. Additional Cards for LARGE = 0. Include as many cards as necessary to collect NHISV (see Card 1) history variables.

Card 3	1	2	3	4	5	6	7	8
Variable	HISV1	HISV2	HISV3	HISV4	HISV5	HISV6	HISV7	HISV8
Type	F	F	F	F	F	F	F	F

Tensor Cards. Additional card for LARGE = 0. Include as many cards as necessary to collect NTENSR (see Card 1) entries. Tensor cards contain only 6 entries per card.

Card 4	1	2	3	4	5	6	7	8
Variable	TENXX	TENYY	TENZZ	TENXY	TENYZ	TENZX		
Type	F	F	F	F	F	F		

Solid Mechanics Data Card for LARGE = 1.

The following set of cards: "Stress Card 1" through "Tensor Cards." Should be included NPLANE × NTHICK times (one set for each integration point).

Stress Card 1. Additional Card for LARGE = 1.

Card 2	1	2	3	4	5	6	7	8	9	10
Variable	T		SIGXX		SIGYY		SIGZZ		SIGXY	
Type	F		F		F		F		F	

Stress Card 2. Additional Card for LARGE = 1.

Card 3	1	2	3	4	5	6	7	8	9	10
Variable	SIGYZ		SIGZX		EPS					
Type	F		F		F					

History Variable Cards. Additional Cards for LARGE = 1. Include as many cards as necessary to collect NHISV (see Card 1) history variables.

Card 4	1	2	3	4	5	6	7	8	9	10
Variable	HISV1		HISV2		HISV3		HISV4		HISV5	
Type	F		F		F		F		F	

Tensor Cards. Include as many pairs of Cards 5 and 6 as necessary to collect NTENSR entries. Note that Cards 5 and 6 must allows appear as pairs, and that Card 6 may include at most one value, as indicated below.

Card 5	1	2	3	4	5	6	7	8	9	10
Variable	TENXX		TENYY		TENZZ		TENXY		TENYZ	
Type	F		F		F		F		F	

Card 6	1	2	3	4	5	6	7	8	9	10
Variable	TENZX									
Type	F									

Thermal Data Cards for LARGE = 1.

For each element, thermal data cards come after the *entire* set of mechanical data cards. For each of the NTHINT thermal integration points, include the following set of cards.

Thermal Time History Cards. Additional cards for LARGE = 1. Include as many cards as needed to collect all the of NTHHSV time history variables per thermal integration point.

Card 7	1	2	3	4	5
Variable	THHSV1	THHSV2	THHSV3	THHSV4	THHSV5
Type	F	F	F	F	F

VARIABLE	DESCRIPTION
EID/SID	Element ID or shell set ID, see *SET_SHELL_…
NPLANE	Number of in plane integration points being output.
NTHICK	Number of integration points through the thickness.
NHISV	Number of additional history variables.
NTENSR	Number of components of tensor data taken from the element history variables stored.
LARGE	Format size. See cards above. EQ.0: off EQ.1: on
T	Parametric coordinate of through thickness integration point between -1 and 1 inclusive.
SIG*ij*	Define the *ij* stress component. The stresses are defined in the GLOBAL cartesian system.
EPS	Effective plastic strain.
HISV*n*	Define the *n*th history variable.
TEN*ij*	Define the *ij*th component of the tensor taken from the history variables. The tensor is defined in the GLOBAL Cartesian system. Define enough lines to provide a total of NTENSOR components, stored six components per line. This applies to material 190 only.
NTHINT	Number of thermal integration points.

VARIABLE	DESCRIPTION
NTHHSV	Number of thermal history variables per thermal integration point.
THHSV*n*	*n*th history variable at the thermal integration point.

INITIAL_STRESS_SOLID_*{OPTION}

Available options include:

<BLANK>

SET

Purpose: Initialize stresses and plastic strains for solid elements. This command is not applicable to hyperelastic materials or any material model based on a Total Lagrangian formulation. Furthermore, for *MAT_014 and any material that requires an equation-of-state (*EOS), the specified initial stresses are adjusted to be in accordance with the initial pressure calculated from the equation of state.

Card Sets per Element. For this keyword, each data card set consists of an element or element set card and all of its corresponding data cards, both thermal and mechanical. For LARGE = 1, this can involve several (even tens of) cards per set. Include cards for as many solid elements or solid element sets as desired. The input is assumed to terminate when a new keyword ("*") card is detected.

Card 1	1	2	3	4	5	6	7	8
Variable	EID/SID	NINT	NHISV	LARGE	IVEFLG	IALEGP	NTHINT	NTHHSV
Type	I	I	I	I	I	I	I	I
Default	none	none	0	0	0	0	0	0

Ordering of Integration Points.

NINT may be 1, 8, or 14 for hexahedral solid elements, depending on the element formulation. If eight Gauss integration points are specified, they should be ordered such that their parametric coordinates are located at:

$$\left(-\frac{\sqrt{3}}{3},-\frac{\sqrt{3}}{3},-\frac{\sqrt{3}}{3}\right),\left(\frac{\sqrt{3}}{3},-\frac{\sqrt{3}}{3},-\frac{\sqrt{3}}{3}\right),\left(\frac{\sqrt{3}}{3},\frac{\sqrt{3}}{3},-\frac{\sqrt{3}}{3}\right),\left(-\frac{\sqrt{3}}{3},\frac{\sqrt{3}}{3},-\frac{\sqrt{3}}{3}\right),$$

$$\left(-\frac{\sqrt{3}}{3},-\frac{\sqrt{3}}{3},\frac{\sqrt{3}}{3}\right),\left(\frac{\sqrt{3}}{3},-\frac{\sqrt{3}}{3},\frac{\sqrt{3}}{3}\right),\left(\frac{\sqrt{3}}{3},\frac{\sqrt{3}}{3},\frac{\sqrt{3}}{3}\right),\left(-\frac{\sqrt{3}}{3},\frac{\sqrt{3}}{3},\frac{\sqrt{3}}{3}\right),$$

respectively. If eight points are defined for 1 point LS-DYNA solid elements, the average value will be taken.

NINT may be 1, 4, or 5 for tetrahedral solid elements, depending on the element formulation and NIPTETS in *CONTROL_SOLID. NINT may be 1 or 2 for pentahedral solid elements, depending on the element formulation.

Solid Mechanics Data Card for LARGE = 0.

Stress Card. Additional Card for LARGE = 0. This card should be included NINT times (one for each integration point).

Card 2	1	2	3	4	5	6	7	8
Variable	SIGXX	SIGYY	SIGZZ	SIGXY	SIGYZ	SIGZX	EPS	
Type	F	F	F	F	F	F	F	

Mechanical Data Cards for LARGE = 1.

The following set of cards "Stress Card 1" through "Additional History Cards." Should be included NINT times (one set for each integration point).

Stress Card 1. Additional cards for LARGE = 1.

Card 2	1	2	3	4	5	6	7	8	9	10
Variable	SIGXX		SIGYY		SIGZZ		SIGXY		SIGYZ	
Type	F		F		F		F		F	

Stress Card 2. Additional cards for LARGE = 1.

Card 3	1	2	3	4	5	6	7	8	9	10
Variable	SIGZX		EPS		HISV1		HISV2		HISV3	
Type	F		F		F		F		F	

Additional History Cards. Additional cards for LARGE = 1. If NHISV > 3 define as many additional cards as necessary. NOTE: the value of IVEFLG (see Card 1) can affect the number of history variables on these cards.

Card 4	1	2	3	4	5	6	7	8	9	10
Variable	HISV1		HISV2		HISV3		HISV4		HISV5	
Type	F		F		F		F		F	

Thermal Data Cards for LARGE = 1.

For each element, thermal data cards come after the *entire* set of mechanical data cards. For each of the NTHINT thermal integration points, include the following set of cards.

Thermal Time History Cards. Additional cards for LARGE = 1. Include as many cards as needed to capture all the of NTHHSV time history variables per thermal integration point.

Card 5	1	2	3	4	5	6	7	8	9	10
Variable	THHSV1		THHSV2		THHSV3		THHSV4		THHSV5	
Type	F		F		F		F		F	

VARIABLE	DESCRIPTION
EID/SID	Element ID or solid set ID, see *SET_SOLID_...
NINT	Number of integration points (should correspond to the solid element formulation).
NHISV	Number of additional history variables, which is typically equal to the number of history variables stored at the integration point + IVEFLG.
LARGE	Format size, if zero, NHISV must also be set to zero (this is the format used by LS-DYNA versions 970 and earlier) and, if set to 1, a larger format is used and NHISV is used.

VARIABLE	DESCRIPTION
IVEFLG	Initial Volume/energy flag (**only used in large format**) EQ.0: last history variable is used as normal, EQ.1: last history variable is used as the initial volume of the element. One additional history variable is required if IVFLG = 1 EQ.2: last two history variables are used to define the initial volume and the internal energy per unit initial volume. Two additional history variables must be allocated, see NHISV above, if IVFLG = 2. If the initial volume is set to zero, the actual element volume is used.
IALEGP	The ALE multi-material group (AMMG) ID; only if the element is of ALE multi-material formulation (ELEFORM = 11). In this case, each AMMG has its own sets of stress and history variables so we must specify to which AMMG the stress data are assigned. For mixed elements, multiple cards are needed to complete the stress initialization in this element as each AMMG needs to have its own set of stress data. EQ.0: Assuming the element is fully filled by the AMMG that the element part belongs to. Please refer to *ALE_MULTI-MATERIAL_GROUP card. EQ.n: Assigning the stress to nth AMMG in that element.
SIG*ij*	Define the *ij*th stress component. Stresses are defined in the GLOBAL Cartesian system.
EPS	Effective plastic strain
HISV*i*	Define $(NHSV + IVEFLG)$ history variables.
NTHINT	Number of thermal integration points
NTHHSV	Number of thermal history variables per thermal integration point
THHSV*n*	*n*th thermal time history variable

Remarks:

1. The elastic material model for cohesive elements is a total Lagrangian formulation, and the initial stress will therefore be ignored for it.

*INITIAL_STRESS_SPH

Purpose: Initialize stresses and plastic strains for SPH elements. This command is not applicable to hyperelastic materials or any material model based on a Total Lagrangian formulation. For *MAT_005, *MAT_014, and any material that requires an equation-of-state (*EOS), the initialized stresses are deviatoric stresses, not total stresses.

Element Cards. Define as many SPH elements in this section as desired. The input is assumed to terminate when a new keyword is detected.

Card 1	1	2	3	4	5	6	7	8
Variable	EID	SIGXX	SIGYY	SIGZZ	SIGXY	SIGYZ	SIGZX	EPS
Type	I	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
EID	SPH particle ID
SIG*ij*	Define the *ij*th stress component. Stresses are defined in the GLOBAL Cartesian system.
EPS	Effective plastic strain.

*INITIAL_STRESS_TSHELL

Purpose: Initialize stresses and plastic strains for thick shell elements.

Card Sets per Element. Define as many thick shell elements in this section as desired. The input is assumed to terminate when a new keyword is detected.

Card 1	1	2	3	4	5	6	7	8
Variable	EID	NPLANE	NTHICK	NHISV	LARGE			
Type	I	I	I	I	I			
Default	none	none	none	0	0			

Ordering of Integration Points.

For each through thickness point define NPLANE points. NPLANE should be either 1 or 4 corresponding to either 1 or 4 Gauss integration points. If four integration points are specified, they should be ordered such that their in plane parametric coordinates are at:

$$\left(-\frac{\sqrt{3}}{3},-\frac{\sqrt{3}}{3}\right),\quad \left(\frac{\sqrt{3}}{3},-\frac{\sqrt{3}}{3}\right),\quad \left(\frac{\sqrt{3}}{3},\frac{\sqrt{3}}{3}\right),\quad \left(-\frac{\sqrt{3}}{3},\frac{\sqrt{3}}{3}\right)$$

respectively. It is not necessary for the location of the through thickness integration points to match those used in the elements which are initialized. The data will be interpolated by LS DYNA.

Data Card for LARGE = 0.

The following set of cards “Stress Card” and “History Cards.” Should be included NPLANE × NTHICK times (one set for each integration point).

Stress Card. Additional card for LARGE = 0.

Card 2	1	2	3	4	5	6	7	8
Variable	T	SIGXX	SIGYY	SIGZZ	SIGXY	SIGYZ	SIGZX	EPS
Type	F	F	F	F	F	F	F	F

History Cards. Additional Card for LARGE = 0. Include as many History Cards as needed to define all NHIST history variables.

Optional	1	2	3	4	5	6	7	8
Variable	HISV1	HSIV2	HSIV3	HSIV4	HSIV5	HSIV6	HSIV7	HSIV8
Type	F	F	F	F	F	F	F	F

Data Card for LARGE = 1.

The following set of cards "Stress Cards" and "History Cards." Should be included NPLANE × NTHICK times (one set for each integration point).

Stress Card 1. Additional card for LARGE = 1.

Card 2	1	2	3	4	5	6	7	8	9	10
Variable	T		SIGXX		SIGYY		SIGZZ		SIGXY	
Type	F		F		F		F		F	

Stress Card 2. Additional card for LARGE = 3.

Card 3	1	2	3	4	5	6	7	8	9	10
Variable	SIGYZ		SIGZX		EPS					
Type	F		F		F					

History Cards. Additional Card for LARGE = 1. Include as many History Cards as needed to define all NHIST history variables.

Optional	1	2	3	4	5	6	7	8	9	10
Variable	HISV1		HISV2		HISV3		HISV4		HISV5	
Type	F		F		F		F		F	

VARIABLE	DESCRIPTION
EID	Element ID
NPLANE	Number of in plane integration points.
NTHICK	Number of integration points through the thickness.
T	Parametric coordinate of through thickness integration point between -1 and 1 inclusive.
NHISV	Number of additional history variables.
LARGE	Format size. See keywords above. EQ.0: off EQ.1: on
SIG*ij*	Define the *ij* stress component. The stresses are defined in the GLOBAL Cartesian system.
EPS	Effective plastic strain

INITIAL_TEMPERATURE_*OPTION

Available options include:

NODE

SET

Purpose: Define initial nodal point temperatures using nodal set ID's or node numbers. These initial temperatures are used in a thermal only analysis or a coupled thermal/structural analysis. See also *CONTROL_THERMAL_SOLVER, *CONTROL_THERMAL_TIMESTEP, and CONTROL_THERMAL_NONLINEAR.

For thermal loading in a structural only analysis, see *LOAD_THERMAL_*OPTION*.

Node/Node set Cards. Include one card for each node or node set. This input ends at the next keyword ("*") keyword.

Card 1	1	2	3	4	5	6	7	8
Variable	NSID/NID	TEMP	LOC					
Type	I	I	I					
Default	none	0.	0					
Remark	1							

VARIABLE	DESCRIPTION
NSID/NID	Nodal set ID or nodal point ID, see also *SET_NODES: EQ.0: all nodes are included (set option only).
TEMP	Temperature at node or node set.
LOC	For a thick thermal shell, the temperature will be applied to the surface identified by LOC, See parameter, THSHEL, on the *CONTROL_SHELL keyword. EQ.-1: lower surface of thermal shell element EQ.0: middle surface of thermal shell element EQ.1: upper surface of thermal shell element

Remarks:

1. This keyword can be used to define initial nodal point temperatures for SPH particles by using nodal set ID's or node numbers from SPH particles.

*INITIAL_VEHICLE_KINEMATICS

Purpose: Define initial kinematical information for a vehicle. In its initial orientation, the vehicle's yaw, pitch, and roll axes must be aligned with the global axes. Successive simple rotations are taken about these body fixed axes.

Card 1	1	2	3	4	5	6	7	8
Variable	GRAV	PSID	XO	YO	ZO	XF	YF	ZF
Type	I	I	F	F	F	F	F	F
Default	none	none	0.	0.	0.	0.	0.	0.

Card 2	1	2	3	4	5	6	7	8
Variable	VX	VY	VZ	AAXIS	BAXIS	CAXIS		
Type	F	F	F	I	I	I		
Default	0.	0.	0.	0	0	0		

Card 3	1	2	3	4	5	6	7	8
Variable	AANG	BANG	CANG	WA	WB	WC		
Type	F	F	F	F	F	F		
Default	0.	0.	0.	0.	0.	0.		

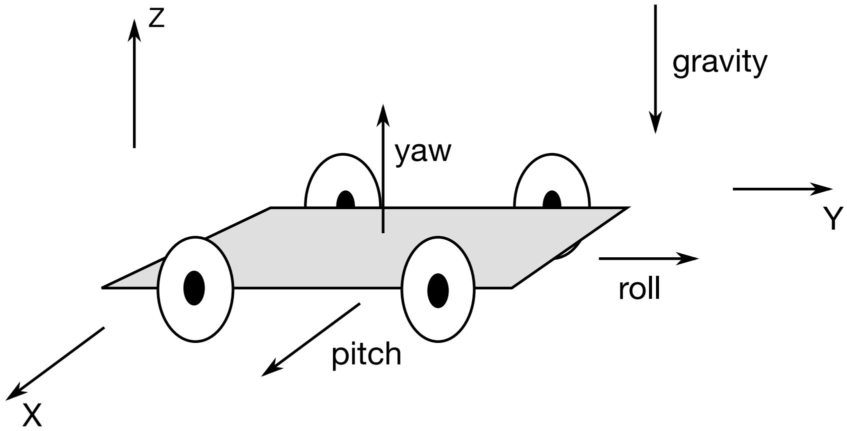


Figure 23-3. The vehicle pictured is to be oriented with a successive rotation sequence about the yaw, pitch, and roll axes, respectively. Accordingly, AAXIS = 3, BAXIS = 1, and CAXIS = 2. The direction of gravity is given by GRAV = -3.

VARIABLE	DESCRIPTION
GRAV	Gravity direction code. EQ.1: Global $+x$ direction. EQ.-1: Global $-x$ direction. EQ.2: Global $+y$ direction. EQ.-2: Global $-y$ direction. EQ.3: Global $+z$ direction. EQ.-3: Global $-z$ direction. Note: this must be the same for all vehicles present in the model.
PSID	Part set ID.
XO	x-coordinate of initial position of mass center.
YO	y-coordinate of initial position of mass center.
ZO	z-coordinate of initial position of mass center.
XF	x-coordinate of final position of mass center.
YF	y-coordinate of final position of mass center.
ZF	z-coordinate of final position of mass center.
VX	global x-component of mass center velocity.

VARIABLE	DESCRIPTION
VY	global y-component of mass center velocity.
VZ	global z-component of mass center velocity.
AAXIS	First rotation axis code. EQ.1: Initially aligned with global x-axis. EQ.2: Initially aligned with global y-axis. EQ.3: Initially aligned with global z-axis.
BAXIS	Second rotation axis code.
CAXIS	Third rotation axis code.
AANG	Rotation angle about the first rotation axis (degrees).
BANG	Rotation angle about the second rotation axis (degrees).
CANG	Rotation angle about the third rotation axis (degrees).
WA	Angular velocity component for the x body-fixed axis (radian/second).
WB	Angular velocity component for the y body-fixed axis (radian/second).
WC	Angular velocity component for the z body-fixed axis (radian/second).

*INITIAL_VELOCITY

Purpose: Define initial nodal point velocities using nodal set ID's. This may also be used for sets in which some nodes have other velocities. See NSIDEX below.

Card 1	1	2	3	4	5	6	7	8
Variable	NSID	NSIDEX	BOXID	IRIGID	ICID			
Type	I	I	I	I	I			
Default	none	0	0	0	0			
Remark	1							

Card 2	1	2	3	4	5	6	7	8
Variable	VX	VY	VZ	VXR	VYR	VZR		
Type	F	F	F	F	F	F		
Default	0.	0.	0.	0.	0.	0.		

Exempted Node Card. Additional card for NSIDEX > 0.

Card 3	1	2	3	4	5	6	7	8
Variable	VXE	VYE	VZE	VXRE	VYRE	VZRE		
Type	F	F	F	F	F	F		
Default	0.	0.	0.	0.	0.	0.		

VARIABLE	DESCRIPTION
NSID	Nodal set ID, see *SET_NODES, containing nodes for initial velocity: If NSID = 0 the initial velocity is applied to all nodes.

VARIABLE **DESCRIPTION**

NSIDEX Nodal set ID, see *SET_NODES, containing nodes that are exempted from the imposed velocities and may have other initial velocities.

BOXID All nodes in box which belong to NSID are initialized. Nodes outside the box are not initialized. Exempted nodes are initialized to velocities defined by VXE, VYE, and VZE below regardless of their location relative to the box.

IRIGID Option to overwrite rigid body velocities defined on *PART_INERTIA and *CONSTRAINED_NODAL_RIGID_BODY_INERTIA cards.

GE.1: part set ID, containing ID of parts to overwrite. Center of gravity of part must lie within box BOXID. If BOXID is not defined then all parts defined in the set are overwritten.

EQ.-1: Overwrite velocities for all *PART_INERTIA's and *CONSTRAINED_NODAL_RIGID_BODY_INERTIA's with a center of gravity within box BOXID. If BOXID is not defined then all are overwritten.

EQ.-2: Overwrite velocities for all *PART_INERTIA's and *CONSTRAINED_NODAL_RIGID_BODY_INERTIA's.

ICID Local coordinate system ID. The initial velocity is specified in the local coordinate system if ICID is greater than zero. Furthermore, if ICID is greater than zero, *INCLUDE_TRANSFORM does not rotate the initial velocity values specificied by VX, VY, …, VZRE.

VX Initial translational velocity in x-direction

VY Initial translational velocity in y-direction

VZ Initial translational velocity in z-direction

VXR Initial rotational velocity about the x-axis

VYR Initial rotational velocity about the y-axis

VZR Initial rotational velocity about the z-axis

VXE Initial velocity in x-direction of exempted nodes

VYE Initial velocity in y-direction of exempted nodes

VARIABLE	DESCRIPTION
VZE	Initial velocity in *z*-direction of exempted nodes
VXRE	Initial rotational velocity in *x*-direction of exempted nodes
VYRE	Initial rotational velocity in *y*-direction of exempted nodes
VZRE	Initial rotational velocity in *z*-direction of exempted nodes

Remarks:

1. This generation input must not be used with *INITIAL_VELOCITY_GENERATION keyword.

2. If a node is initialized on more than one input card set, then the last set input will determine its velocity. However, if the nodal velocity is also specified on a *INITIAL_VELOCITY_NODE card, then the velocity specification on this card will be used.

3. Unless the option IRIGID is specified rigid bodies, initial velocities given in *PART_INERTIA will overwrite generated initial velocities. The IRIGID option will cause the rigid body velocities specified on the *PART_INERTIA input to be overwritten. To directly specify the motion of a rigid body without using the keyword, *PART_INERTIA, which also requires the definition of the mass properties, use the keyword option, *INITIAL_VELOCITY_RIGID_BODY.

4. Nodes which belong to rigid bodies must have motion consistent with the translational and rotational velocity of the center of gravity (c.g.) of the rigid body. During initialization the rigid body translational and rotational rigid body momentum's are computed based on the prescribed nodal velocity field. From this rigid body momentum, the translational and rotational velocities of the nodal points are computed and reset to the new values. These new values may or may not be the same as the values prescribed for the nodes that make up the rigid body. Sometimes this occurs in single precision due to numerical round-off. If a problem like this occurs specify the velocity using the keyword: *INITIAL_VELOCITY_RIGID_BODY.

5. Mid-side nodes generated by *ELEMENT_SOLID_TET4TO10 will not be initialized since the node numbers are not known a priori to the user. Instead use *INITIAL_VELOCITY_GENERATION if you intend to initialize the velocities of the mid-side nodes.

*INITIAL_VELOCITY_NODE

Purpose: Define initial nodal point velocities for a node.

Card	1	2	3	4	5	6	7	8
Variable	NID	VX	VY	VZ	VXR	VYR	VZR	ICID
Type	I	F	F	F	F	F	F	I
Default	none	0.	0.	0.	0.	0.	0.	0

VARIABLE	DESCRIPTION
NID	Node ID
VX	Initial translational velocity in x-direction
VY	Initial translational velocity in y-direction
VZ	Initial translational velocity in z-direction
VXR	Initial rotational velocity about the x-axis
VYR	Initial rotational velocity about the y-axis
VZR	Initial rotational velocity about the z-axis
ICID	Local coordinate system ID. The specified velocities are in the local system if ICID is greater than zero. Furthermore, if ICID is greater than zero, *INCLUDE_TRANSFORM does not rotate the initial velocity values specificied by VX, VY, …, VZR.

See Remarks on *INITIAL_VELOCITY card.

*INITIAL_VELOCITY_RIGID_BODY

Purpose: Define the initial translational and rotational velocities at the center of gravity (c.g.) for a rigid body or a nodal rigid body. This input overrides all other velocity input for the rigid body and the nodes which define the rigid body.

Card	1	2	3	4	5	6	7	8
Variable	PID	VX	VY	VZ	VXR	VYR	VZR	ICID
Type	I	F	F	F	F	F	F	I
Default	none	0.	0.	0.	0.	0.	0.	0

VARIABLE	DESCRIPTION
PID	Part ID of the rigid body or the nodal rigid body.
VX	Initial translational velocity at the c.g. in global *x*-direction.
VY	Initial translational velocity at the c.g. in global *y*-direction.
VZ	Initial translational velocity at the c.g. in global *z*-direction.
VXR	Initial rotational velocity at the c.g. about the global *x*-axis.
VYR	Initial rotational velocity at the c.g. about the global *y*-axis.
VZR	Initial rotational velocity at the c.g. about the global *z*-axis.
ICID	Local coordinate system ID. The specified velocities are in the local system if ICID is greater than zero. Furthermore, if ICID is greater than zero, *INCLUDE_TRANSFORM does not transform the initial velocity values specificied by VX, VY, …, VZR.

See remarks 3 and 4 on the *INITIAL_VELOCITY input description.

*INITIAL_VELOCITY_GENERATION

Purpose: Define initial velocities for rotating and translating bodies.

NOTE: Rigid body velocities cannot be reinitialized after dynamic relaxation by setting PHASE=1 since rigid body velocities are always restored to the values that existed prior to dynamic relaxation. Reinitialization of velocities after dynamic relaxation is only available for nodal points of deformable bodies; therefore, if rigid bodies are present in the part set ID, this input should be defined twice, once for PHASE=0 and again for PHASE=1.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	STYP	OMEGA	VX	VY	VZ	IVATN	ICID
Type	I	I	F	F	F	F	I	I
Default	none	none	0.	0.	0.	0.	0	0

Card 2	1	2	3	4	5	6	7	8
Variable	XC	YC	ZC	NX	NY	NZ	PHASE	IRIGID
Type	F	F	F	F	F	F	I	I
Default	0.	0.	0.	0.	0.	0.	0	0

VARIABLE	DESCRIPTION
ID	Part ID, part set ID, or node set ID. If zero, STYP is ignored, and all velocities are set. WARNING if IVATN = 0: If a part ID of a rigid body is specified only the nodes that belong to elements of the rigid body are initialized. Nodes defined under the keyword. *CONSTRAINED_EXTRA_NODES are not initialized. Set IVATN = 1 to initialize velocities of slaved nodes and parts.

VARIABLE DESCRIPTION

STYP

Set type. See Remark 5.

EQ.1: part set ID, see *SET_PART,

EQ.2: part ID, see *PART,

EQ.3: node set ID, see *SET_NODE.

OMEGA

Angular velocity about the rotational axis.

VX

Initial translational velocity in x-direction (see ICID below).

VY

Initial translational velocity in y-direction (see ICID below).

VZ

Initial translational velocity in z-direction (see ICID below).

IVATN

Flag for setting the initial velocities of slave nodes and parts:

EQ.0: slaved parts are ignored.

EQ.1: slaved parts and slaved nodes of the master parts will be assigned initial velocities like the master part.

ICID

Local coordinate system ID. The specified translational velocities (VX, VY, VZ) and the direction cosines of the rotation axis (NX, NY, NZ) are in the global system if ICID = 0 and are in the local system if ICID is defined. Therefore, if ICID is defined, *INCLUDE_TRANSFORM does not transform (VX, VY, VZ) and (NX, NY, NZ).

XC

Global x-coordinate on rotational axis.

YC

Global y-coordinate on rotational axis.

ZC

Global z-coordinate on rotational axis.

NX

x-direction cosine. If set to -999, NY and NZ are interpreted as the 1st and 2nd nodes defining the rotational axis, in which case the coordinates of node NY are used as XC, YC, ZC. If ICID is defined, the direction cosine, (NX, NY, NZ), is projected along coordinate system ICID to yield the direction cosines of the rotation axis only if NX ≠ -999.

NY

y-direction cosine or the 1st node of the rotational axis when NX = -999.

VARIABLE **DESCRIPTION**

NZ

z-direction cosine or the 2nd node of the rotational axis when NX = -999.

PHASE

Flag specifying phase of the analysis the velocities apply to:

EQ.0: Velocities are applied immediately,

EQ.1: Velocities are applied after reaching the start time, STIME, which is after dynamic relaxation, if active, is completed. See the keyword: *INITIAL_VELOCITY_GENERATION_START_TIME. STIME defaults to zero.

IRIGID

Controls hierarchy of initial velocities set with *INITIAL_VELOCITY_GENERATION versus those set with *PART_INERTIA / *CONSTRAINED_NODAL_RIGID_BODY_INERTIA when the commands conflict.

EQ.0: *PART_INERTIA / *CONSTRAINED_NODAL_RIGID_BODY_INERTIA controls initial velocities.

EQ.1: *INITIAL_VELOCITY_GENERATION controls initial velocities. This option does not apply if STYP = 3.

Remarks:

1. **Exclusions.** This generation input must not be used with *INITIAL_VELOCITY or *INITIAL_VELOCITY_NODE options.

2. **Order Dependence.** The velocities are initialized in the order the *INITIAL_VELOCITY_GENERATION input is defined. Later input via the *INITIAL_VELOCITY_GENERATION keyword may overwrite the velocities previously set.

3. **Consistency for Rigid Body Nodes.** Nodes which belong to rigid bodies must have motion consistent with the translational and rotational velocity of the rigid body. During initialization the rigid body translational and rotational rigid body momentum's are computed based on the prescribed nodal velocities. From this rigid body motion the velocities of the nodal points are computed and reset to the new values. These new values may or may not be the same as the values prescribed for the node.

4. **SPH.** SPH elements can be initialized using the STYP=3 option only.

5. **Constrained Nodal Rigid Bodies.** Part IDs of *CONSTRAINED_NODAL_RIGID_BODYs that do not include the INERTIA option are not recognized by

the code in the case of STYP=1 or 2. Use STYP=3 (a node set ID) when initializing velocity of such nodal rigid bodies.

*INITIAL_VELOCITY_GENERATION_START_TIME

Purpose: Define a time to initialize velocities after time zero. Time zero starts after dynamic relaxation if used for initialization.

Card 1	1	2	3	4	5	6	7	8
Variable	STIME							
Type	F							
Default	0.0							

VARIABLE	DESCRIPTION
STIME	Start time.

Remarks:

1. Only one *INITIAL_VELOCITY_GENERATION_START_TIME can be specified. Multiple start times are not allowed.

2. All *INITIAL_VELOCITY_GENERATION commands adhere to the start time provided the requirement is met that at least one of those commands has PHASE set to 1.

3. When *INITIAL_VELOCITY_GENERATION_START_TIME is active, nodes that are not part of the initial velocity generation definitions will be re-initialized with velocities as they were at $t = 0$.

INITIAL_VOID_*OPTION

Available options include:

PART

SET

Purpose: Define initial voided part set ID's or part numbers. This command can be used only when ELFORM = 12 in *SECTION_SOLID. Void materials cannot be created during the calculation. Fluid elements which are evacuated, e.g., by a projectile moving through the fluid, during the calculation are approximated as fluid elements with very low densities. The constitutive properties of fluid materials used as voids must be identical to those of the materials which will fill the voided elements during the calculation. Mixing of two fluids with different properties is not permitted with this option.

Card 1	1	2	3	4	5	6	7	8
Variable	PSID/PID							
Type	I							
Default	none							
Remark	1							

VARIABLE	DESCRIPTION
PSID/PID	Part set ID or part ID, see also *SET_PART:

Remarks:

This void option and multiple materials per element, see *ALE_MULTI-MATERIAL_-GROUP are incompatible and cannot be used together in the same run.

INITIAL_VOLUME_FRACTION_*OPTION

Available option:

NALEGP

Purpose: Define initial volume fractions of different materials in multi-material ALE elements. Without NALEGP option, the keyword allows up to 7 ALE multi-material groups. The NALEGP option adds in an additional card immediately after the keyword to let users input the number of ALE multi-material groups to be read in for each element.

Format without NALEGP option:

Card 1	1	2	3	4	5	6	7	8
Variable	EID	VF1	VF2	VF3	VF4	VF5	VF6	VF7
Type	I	F	F	F	F	F	F	F
Default	none	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Card 1 repeated multiple times. Each card for an ALE element volume fraction information.

Format with NALEGP option:

NALEGP Card. Additional card for the NALEGP keyword option.

Optional	1	
Variable	NALEGP	
Type	I	

Card 1	1	2	3	4	5	6	7	8
Variable	EID	VF1	VF2	VF3	VF4	VF5	VF6	VF7
Type	I	F	F	F	F	F	F	F
Default	none	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Card 1a	1	2	3	4	5	6	7	8
Variable	VF8	VF9	VF10	VF11	VF12	VF13	VF14	VF15
Type	I	F	F	F	F	F	F	F
Default	none	0.0	0.0	0.0	0.0	0.0	0.0	0.0

…

Card 1z	1	2	3	4	5	6	7	8
Variable	VF(NGP-2)	VF(NGP-1)	VF(NGP)					
Type	I	F	F					
Default	none	0.0	0.0					

Card 1 is continued by additional cards until all the NALEGP volume fractions for this element are listed. Card 1 and its continuation cards are repeated until all ALE elements are finished.

VARIABLE	DESCRIPTION
EID	Element ID.
VF1	Volume fraction of multi-material group 1, AMMGID = 1.
VF2	Volume fraction of multi-material group 2. Only needed in simulations with 3 material groups. Otherwise VF2 = 1 – VF1.

VARIABLE	DESCRIPTION
VF3	Volume fraction of multi-material group 3, AMMGID = 3.
VF4	Volume fraction of multi-material group 4, AMMGID = 4.
VF5	Volume fraction of multi-material group 5, AMMGID = 5.
VF6	Volume fraction of multi-material group 6, AMMGID = 6.
VF7	Volume fraction of multi-material group 7, AMMGID = 7.
VF(N)	Volume fraction of multi-material group N, AMMGID = N.

*INITIAL_VOLUME_FRACTION_GEOMETRY

Purpose: This is a volume-filling command for defining the volume fractions of various ALE multi-material groups (AMMG) that initially occupy various spatial regions in an ALE mesh. This applies only to ELFORMs 11 and 12 in *SECTION_SOLID and ALEFORM 11 in *SECTION_ALE2D. For ELFORM 12, AMMGID 2 is void. See Remark 2.

Background ALE Mesh Card. Defines the background ALE mesh set & an AMMGID that initially fills it.

Card 1	1	2	3	4	5	6	7	8
Variable	FMSID	FMIDTYP	BAMMG	NTRACE				
Type	I	I	I	I				
Default	none	0	0	3				

VARIABLE	DESCRIPTION
FMSID	A background ALE (fluid) mesh SID to be initialized or filled with various AMMG's. This set ID refers to one or more ALE parts.
FMIDTYP	ALE mesh set ID type: EQ.0: FMSID is an ALE part set ID (PSID). EQ.1: FMSID is an ALE part ID (PID).
BAMMG	The background fluid group ID or ALE Multi-Material group ID (AMMGID) that initially fills all ALE mesh region defined by FMSID.
NTRACE	Number of sampling points for volume filling detection. Typically NTRACE ranges from 3 to maybe 10 (or more). The higher it is, the finer the ALE element is divided so that small gaps between 2 Lagrangian shells may be filled in. See Remark 4.

Pairs of Container Cards.

For each container include one "Container Card" (Card a) and one geometry card (Card b*i*). Include as many pairs as desired. This input ends at the next keyword ("*") card.

Contained Card. Defines the container type and the AMMGID that fills the region defined by the container type.

Card a	1	2	3	4	5	6	7	8
Variable	CNTTYP	FILLOPT	FAMMG	VX	VY	VZ		
Type	I	I	I	F	F	F		
Default	none	0	none	0	0	0		

VARIABLE	DESCRIPTION
CNTTYP	A "container" defines a Lagrangian surface boundary of a spatial region, inside (or outside) of which, an AMMG would fill up. CNTTYP defines the container geometry type of this surface boundary (or shell structure). EQ.1: The container geometry is defined by a part ID (PID) or a part set ID (PSID), where the parts should be defined by shell elements (see *PART or *SET_PART). EQ.2: The container geometry is defined by a segment set (SGSID). EQ.3: The container geometry is defined by a plane: a point and a normal vector. EQ.4: The container geometry is defined by a conical surface: 2 end points and 2 corresponding radii (in 2D see Remark 6). EQ.5: The container geometry is defined by a cuboid or rectangular box: 2 opposing end points, minimum to maximum coordinates. EQ.6: The container geometry is defined by a sphere: 1 center point, and a radius.

VARIABLE **DESCRIPTION**

FILLOPT

A flag to indicate which side of the container surface the AMMG is supposed to fill. The "head" side of a container surface/segment is defined as the side pointed to by the heads of the normal vectors of the segments ("tail" side refers to opposite direction to "head"). See Remark 5.

EQ.0: The "head" side of the geometry defined above will be filled with fluid (default).

EQ.1: The "tail" side of the geometry defined above will be filled with fluid.

FAMMG

This defines the fluid group ID or ALE Multi-Material group ID (AMMGID) which will fill up the interior (or exterior) of the space defined by the "container". *The order of AMMGIDs is determined by the order in which they are listed under *ALE_MULTI-MATERIAL_GROUP card.* For example, the first data card under the *ALE_MULTI-MATERIAL_GROUP keyword defines the multi-material group with ID (AMMGUD) 1, the second data card defined AMMGID = 2 and so on.

LT.0: | FAMMG | is a *SET_MULTI-MATERIAL_GROUP_LIST id listing pairs of group IDs. For each pair, the 2nd group replaces the first one in the "container".

VX

Initial velocity in the global x-direction for this AMMGID.

VY

Initial velocity in the global y-direction for this AMMGID.

VZ

Initial velocity in the global z-direction for this AMMGID.

Part/Part Set Container Card. Additional card for CNTTYP = 1.

Card b1	1	2	3	4	5	6	7	8
Variable	SID	STYPE	NORMDIR	XOFFST				
Type	I	I	I	F				
Default	none	0	0	0.0				
Remark			obsolete					

VARIABLE	DESCRIPTION
SID	A Set ID pointing to a part ID (PID) or part set ID (PSID) of the Lagrangian shell element structure defining the "container" geometry to be filled (see *PART or *SET_PART).
SSTYPE	Set ID type: EQ.0: Container SID is a Lagrangian part set ID (PSID). EQ.1: Container SID is a Lagrangian part ID (PID).
NORMDIR	Obsolete (see Remark 5).
XOFFST	Absolute length unit for offsetting the fluid interface from the nominal fluid interface LS-DYNA would otherwise define by default. This parameter only applies to GEOTYPE = 1 (4th column) and GEOTYPE = 2 (3rd column). This is applicable to cases in which high pressure fluid is contained within a container. The offset allows LS-DYNA time to prevent leakage. In general, this may be set to roughly 5-10% of the ALE elm width. It may be important only for when ILEAK is turned ON to give the code time to "catch" the leakage. If ILEAK is not ON, this may not be necessary.

Segment Set Container Card. Additional card for CNTTYP = 2.

Card b2	1	2	3	4	5	6	7	8
Variable	SGSID	NORMDIR	XOFFST					
Type	I	I	F					
Default	none	0	0.0					
Remark		obsolete						

VARIABLE	DESCRIPTION
SGSID	Segment Set ID defining the “container”, see *SET_SEGMENT.
NORMDIR	Obsolete (see Remark 5).
XOFFST	Absolute length unit for offsetting the fluid interface from the nominal fluid interface LSDYNA would otherwise define by default. This parameter only applies to GEOTYPE = 1 (4th column) and GEOTYPE = 2 (3rd column). This is applicable to cases in which high pressure fluid is contained within a container. The offset allows LS-DYNA time to prevent leakage. In general, this may be set to roughly 5-10% of the ALE elm width. It may be important only for when ILEAK is turned ON to give the code time to "catch" the leakage. If ILEAK is not ON, this may not be necessary.

Plane Card. Additional card for CNTTYP = 3.

Card b3	1	2	3	4	5	6	7	8
Variable	X0	Y0	Z0	XCOS	YCOS	ZCOS		
Type	F	F	F	F	F	F		
Default	none	none	none	none	none	none		

VARIABLE	DESCRIPTION
X0, Y0, Z0	x, y and z coordinate of a spatial point on the plane.
XCOS, YCOS, ZCOS	x, y and z direction cosines of the plane normal vector. The filling will occur on the side pointed to by the plane normal vector (or “head” side).

Cylinder/Cone Container Card. Additional Card for CNTTYP = 4 (see Remark 6 for 2D).

Card b4	1	2	3	4	5	6	7	8
Variable	X0	Y0	Z0	X1	Y1	Z1	R1	R2
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
X0, Y0, Z0	x, y and z coordinate of the center of the 1st base of the cone.
X1, Y1, Z1	x, y and z coordinate of the center of the 2nd base of the cone.
R1	Radius of the 1st base of the cone
R2	Radius of the 2nd base of the cone

Rectangular Box Container Card. Additional Card for CNTTYP = 5.

Card b5	1	2	3	4	5	6	7	8
Variable	X0	Y0	Z0	X1	Y1	Z1	LCSID	
Type	F	F	F	F	F	F	I	
Default	none	none	none	none	none	none	none	

VARIABLE	DESCRIPTION
X0, Y0, Z0	Minimum x, y and z coordinates of the box.

VARIABLE	DESCRIPTION
X1, Y1, Z1	Maximum x, y and z coordinates of the box.
LCSID	Local coordinate system ID, if defined, the box is aligned with the local coordinate system instead of global coordinate system. Please see *DEFINE_COORDINATE_ for details.

Sphere Container Card. Additional card for CNTTYP = 6.

Card b6	1	2	3	4	5	6	7	8
Variable	X0	Y0	Z0	R0				
Type	F	F	F	F				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
X0, Y0, Z0	x, y and z coordinate of the center of the sphere.
R0	Radius of the sphere

Remarks:

1. **Structure of Data Cards.** After card 1 defining the basic mesh filled by certain fluid group (AMMGID), each “filling action” will require 2 additional lines of input (cards a, and b#, where # is the CNTTYP value). At the minimum there will be 3 cards required for this command (1, a, and b#) for 1 “filling action”.

   There can be one or more “filling actions” prescribed for each instance of this command. The “filling actions” take place in the prescribed order and the effects are cumulative. Later filling actions will over-write the previous ones. Therefore any complex filling logic will require some planning. For example, the following card sequence, with 2 “filing actions”, is allowable:

   ```
   *INITIAL_VOLIME_FRACTION_GEOMTRY

   [Card 1]

   [Card a, CNTTYP = 1]

   [Card b1]
   ```

[Card a, CNTTYP = 3]

[Card b3]

This sequence of cards prescribes a system of background ALE mesh with 2 "filing actions" to be executed. The 1st is a filling of a CNTTYP = 1, and the 2nd of CNTTYP = 3.

"Card a" is required for all container geometry types (CNTTYP). "Card b*i*" defines the container actual geometry and corresponds to each of the CNTTYP choice.

2. **Group IDs for ELFORM 12.** If ELFORM=12, the single-material-and-void element forumation, is used in *SECTION_SOLID, then the non-void material is defaults to AMMG = 1 and the void to AMMG = 2. These multi-material groups are implied even though no *ALE_MULTI-MATERIAL_GROUP card is required.

3. **Using Shells to Divide Space.** A simple ALE background mesh (for example, a cuboid mesh) can be constructed enveloping some Lagrangian shell structure (or container). The ALE region inside this Lagrangian shell container may be filled with one multi-material group (AMMG1), and the outside region with another (AMMG2). This approach simplifies the mesh generation requirements for ALE material parts with complex geometries.

4. **NTRACE.** Default is NTRACE=3 in which case the total number is

$$(2 \times \mathrm{NTRACE} + 1)^3 = 7^3$$

This means an ALE element is subdivided into 7 × 7 × 7 regions. Each is to be filled in with the appropriate AMMG. An example of this application would be the filling of initial gas between multiple layers of Lagrangian airbag shell elements sharing the same ALE element.

5. **Interior/Exterior Fill Setting.** To set which side of a container is to be filled: (1) define the shell (or segment) container with inward normal vectorsl; then (2) set the FILLOPT field on "Card a" to 0, corresponding to the head of the normal, for the interior, and to 1, corresponding to the tail of the normal, for the exterior.

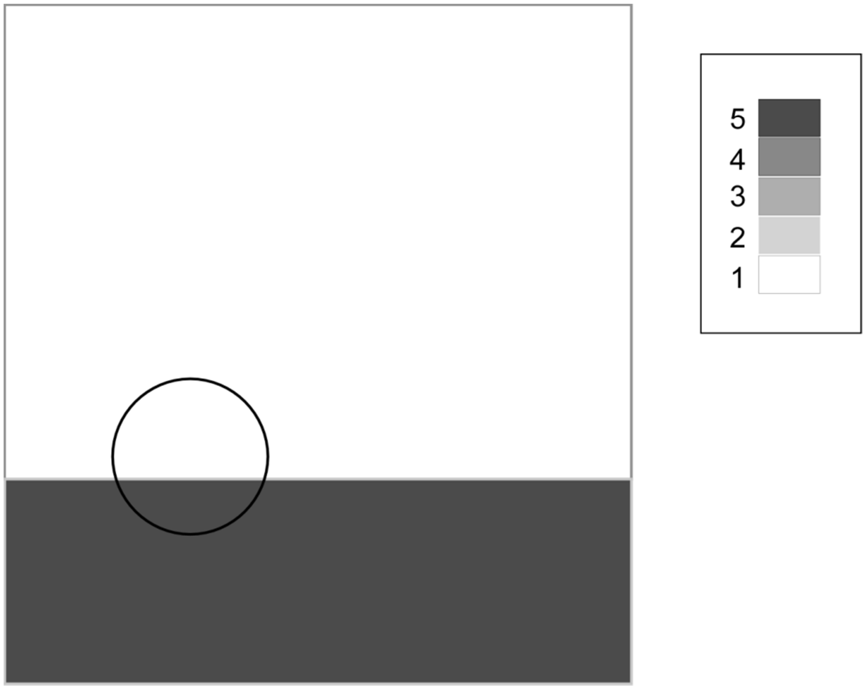
6. **Two Dimensional Geometry.** If the ALE model is 2D (*SECTION_ALE2D instead of *SECTION_SOLID), CNTTYP=4 defines a quadrangle. In this case the fields which, in the 3D case define a cone, are interpreted as the corner coordinates of a clockwise defined (inward normal) quadrangle having the vertices: (X1, Y1), (X2, Y2), (X3, Y3), and (X4, Y4). The CNTYPE = 4 input fields X0, Y0, Z0, X1, Y1, Z1, R1, and R2 becomes X1, Y1, X2, Y2, X3, Y3, X4, and Y4 respectively. CNTTYP=6 should be used to fill a circle.

Example:

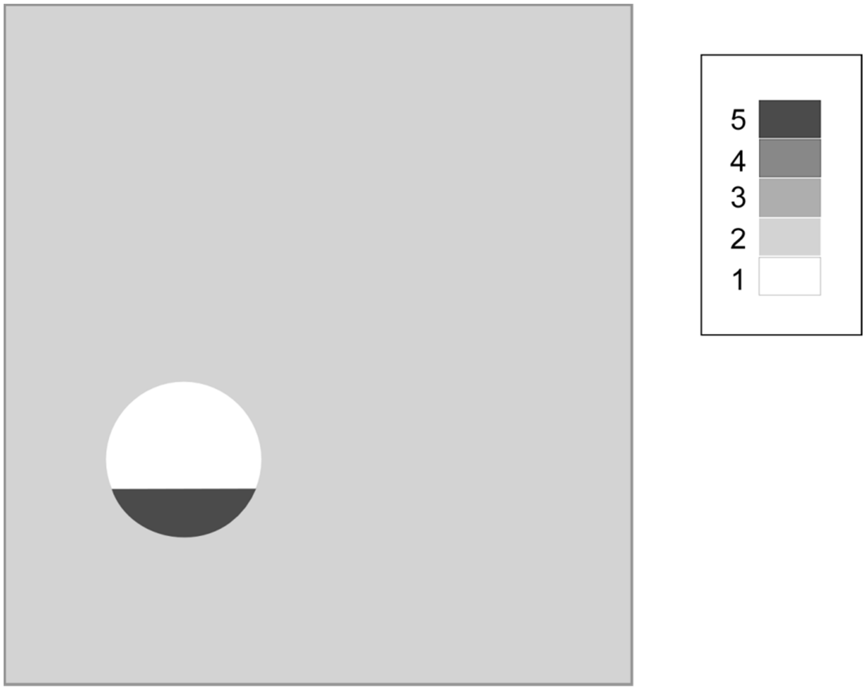
Consider a simple ALE model with ALE parts H1-H5 (5 AMMGs possible) and 1 Lagrangian shell (container) part S6. Only parts H1 and S6 initially have their meshes defined. We will perform 4 "filling actions". The volume filling results after each step will be shown below to clarify the concept used. The input for the volume filling looks like this.

```
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
$ H1 = AMMG 1 = fluid 1 initially occupying whole ALE mesh= background mesh
$ H5 = AMMG 5 = fluid 5 fills below a plane = filling action 1 = CNTTYP=3
$ H2 = AMMG 2 = fluid 2 fills outside S5    = filling action 2 = CNTTYP=1
$ H3 = AMMG 3 = fluid 3 fills inside a cone = filling action 3 = CNTTYP=4
$ H4 = AMMG 4 = fluid 4 fills inside a box  = filling action 4 = CNTTYP=5
$ S6 =          Lagrangian shell container
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
*ALE_MULTI-MATERIAL_GROUP
         1         1
         2         1
         3         1
         4         1
         5         1
*INITIAL_VOLUME_FRACTION_GEOMETRY
$ The 1st card fills the whole pid H1 with AMMG 1=background ALE mesh
$    FMSID   FMIDTYP     BAMMG     <=== card 1: background fluid
         1         1         1
$ filling action 1 = AMMG 5 fill all elms below a plane
$ CNTTYPE   FILLOPT FILAMMGID     <=== card a  : container: CNTTYPE=3=plane
         3         0         5
$   X0, Y0,   Z0,    NX,  NY, NZ   <=== card b-3: details on container =plane
  25.0,20.0, 0.0,   0.0,-1.0,0.0
$ filling action 2: AMMG 2 fills OUTSIDE (FILLOPT=1) shell S6 (inward normals);
$ CNTTYPE   FILLOPT     FAMMG  <== card a  : container #1; FILLOPT=1=fill tail
         1         1         2
$    SETID   SETTYPE   NORMDIR  <== card b-1: details on container #1
         6         1         0
$ filling action 3 = AMMG 3 fill all elms inside a CONICAL region
$ CNTTYPE   FILLOPT     FAMMG  CNTTYP = 4 = Container = conical region
         4         0         3
$       X1        Y1        Z1        X2        Y2        Z2        R1        R2
      25.0      75.0       0.0      25.0      75.0       1.0       8.0       8.0
$ filling action 4 = AMMG 4 fill all elms inside a BOX region
$ CNTTYPE   FILLOPT  FFLUIDID                 : CNTTYP=5 = "BOX"
         5         0         4
$     XMIN      YMIN      ZMIN      XMAX      YMAX      ZMAX
      65.0      35.0       0.0      85.0      65.0       1.0
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
```

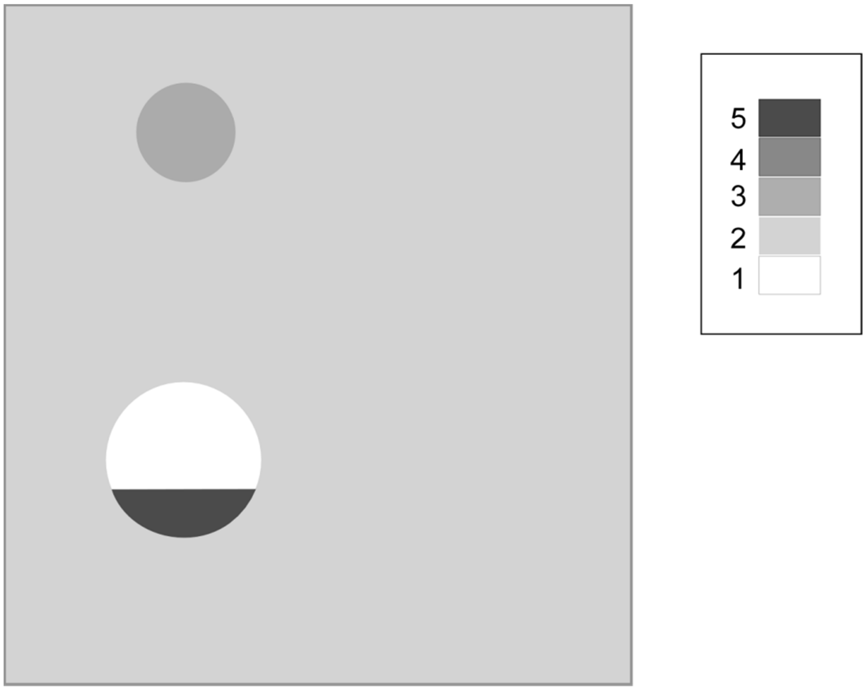
Before the 1st "filling action" the whole ALE mesh of part H1 is filled with AMMG 1 (white). After the 1st "filling action", AMMG 5 fills below the specified plane.



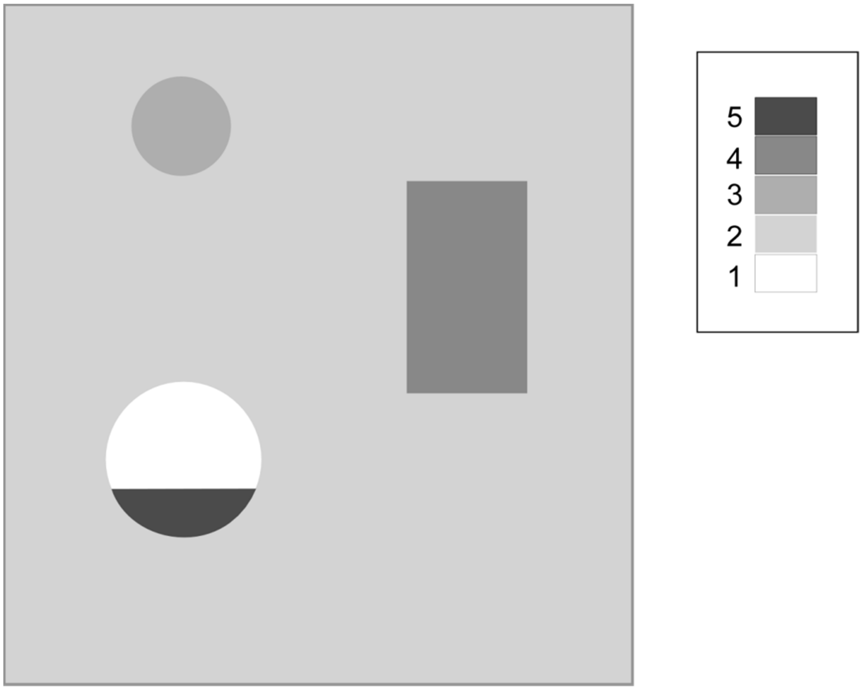
After the 1st and 2nd "filling actions", it fills outside the shell (S6) with AMMG 2.



After the 1st, 2nd and 3rd "filling actions", it fills in the analytical sphere with AMMG 3.



After the 1st, 2nd, 3rd and 4th "filling actions", it fills in the analytical box with AMMG 4.



*INTEGRATION

In this section the user defined integration rules for beam and shell elements are specified. IRID refers to integration rule identification number on *SECTION_BEAM and *SECTION_SHELL cards respectively. Quadrature rules in the *SECTION_SHELL and *SECTION_BEAM cards need to be specified as a negative number. The absolute value of the negative number refers to user defined integration rule number. Positive rule numbers refer to the built in quadrature rules within LS-DYNA. The keyword cards in this section are:

*INTEGRATION_BEAM

*INTEGRATION_SHELL

*INTEGRATION_BEAM

Purpose: To support user defined through the thickness integration rules for the beam element.

Card 1	1	2	3	4	5	6	7	8
Variable	IRID	NIP	RA	ICST	K			
Type	I	I	F	I	I			
Default	none	0	0.0	0	0			

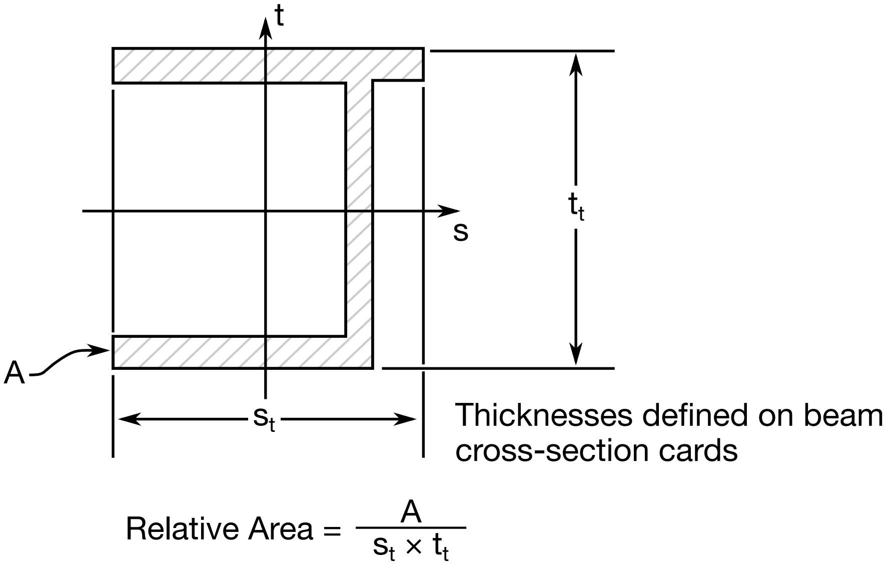
Standard Cross-Section Card. Additional card for ICST > 0.

Card	1	2	3	4	5	6	7	8
Variable	D1	D2	D3	D4	SREF	TREF	D5	D6
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	0.0	0.0	none	none

Quadrature Cards. Include NIP additional cards below for NIP ≠ 0.

Card	1	2	3	4	5	6	7	8
Variable	S	T	WF	PID				
Type	F	F	F	I				

VARIABLE	DESCRIPTION
IRID	Integration rule ID. IRID refers to IRID on *SECTION_BEAM card.
NIP	Number of integration points, see also ICST.



t
s
t_t
A
s_t
Thicknesses defined on beam cross-section cards
Relative Area = $\frac{A}{s_t \times t_t}$

Figure 24-1. Definition of relative area for user defined integration rule.

VARIABLE	DESCRIPTION
RA	Relative area of cross section, i.e., the actual cross-sectional area divided by the area defined by the product of the specified thickness in the s direction and the thickness in the t direction. See also ICST below and Figure 24-1.
ICST	Standard cross section type, ICST. If this type is nonzero then NIP and the relative area above should be input as zero. See shapes in Figure 24-2 following Remarks.

EQ.01: I-Shape	EQ.12: Cross
EQ.02: Channel	EQ.13: H-Shape
EQ.03: L-shape	EQ.14: T-Shape 2
EQ.04: T-shape	EQ.15: I-Shape 3
EQ.05: Tubular box	EQ.16: Channel 2
EQ.06: Z-Shape	EQ.17: Channel 3
EQ.07: Trapezoidal	EQ.18: T-Shape 3
EQ.08: Circular	EQ.19: Box-Shape 2
EQ.09: Tubular	EQ.20: Hexagon
EQ.10: I-Shape 2	EQ.21: Hat-Shape
EQ.11: Solid Box	EQ.22: Hat-Shape 2

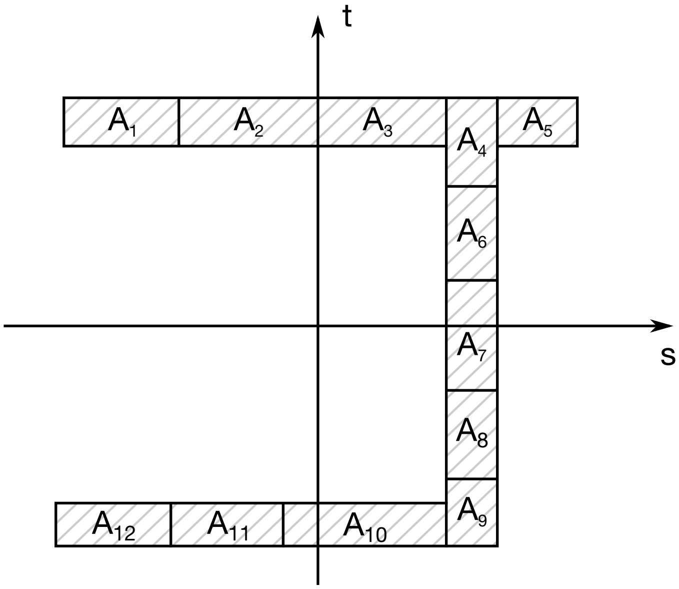


Figure 24-2. Definition of integration points for user defined integration rule.

VARIABLE	DESCRIPTION
K	Integration refinement parameter for standard cross section types. Select an integer ≥ 0. See Figure below.
D1-D6	Cross-section dimensions. See Figure below.
SREF	s_{ref}, location of reference surface normal to s, for the Hughes-Liu beam only. This option is only useful if the beam is connected to a shell or another beam on its outer surface. Overrides NSLOC in *SECTION_BEAM even if SREF = 0.
TREF	t_{ref}, location of reference surface normal to t, for the Hughes-Liu beam only. This option is only useful if the beam is connected to a shell or another beam on its outer surface. Overrides NTLOC in *SECTION_BEAM even if TREF = 0.
S	Normalized s coordinate of integration point, $-1 \leq s \leq 1$.
T	Normalized t coordinate of integration point, $-1 \leq t \leq 1$.
WF	Weighting factor, A_{ri} , i.e., the area associated with the integration point divided by actual cross sectional area $A_{ri} = {}^{A_i}/_{A}$, see Figure 24-1.

VARIABLE	DESCRIPTION
PID	Optional PID, used to identify material properties for this integration point. If zero, the "master" PID (referenced on *ELEMENT) will be used. This feature will be available in release 3 of version 971.

Remarks:

The input for standard beam section types is defined below. In following figures the dimensions are shown on the left and the location of the integration points are shown on the right. If a quantity is not defined in the sketch, then it should be set to zero in the input. The input quantities include:

D1 - D6 = Dimensions of section
k = Integration refinement parameter (an integer GE. 0)
s_{ref} = location of reference surface normal to s, Hughes-Liu beam only
t_{ref} = location of reference surface normal to t, Hughes-Liu beam only

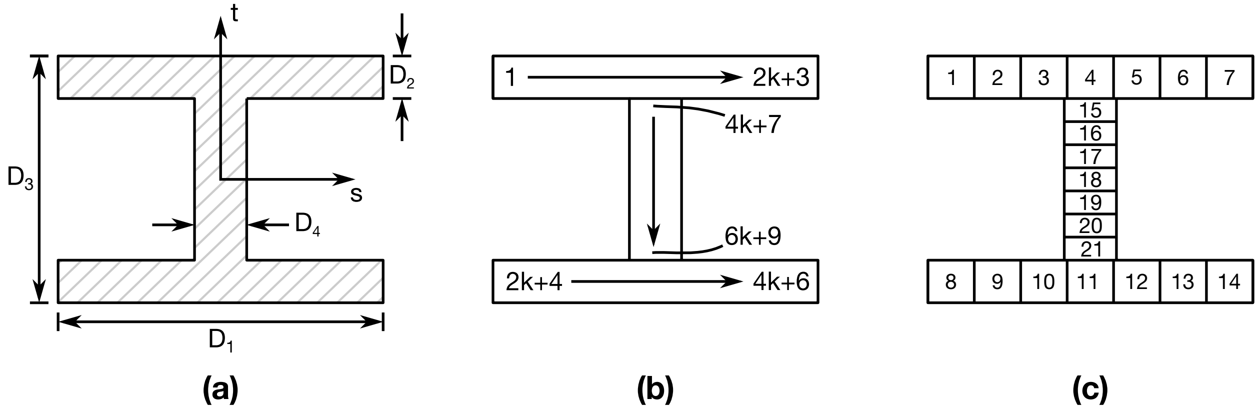


Figure 24-3. Type 1: I-Shape. (a) Cross section geometry. (b) Integration point numbering. (c) Example for k = 2.

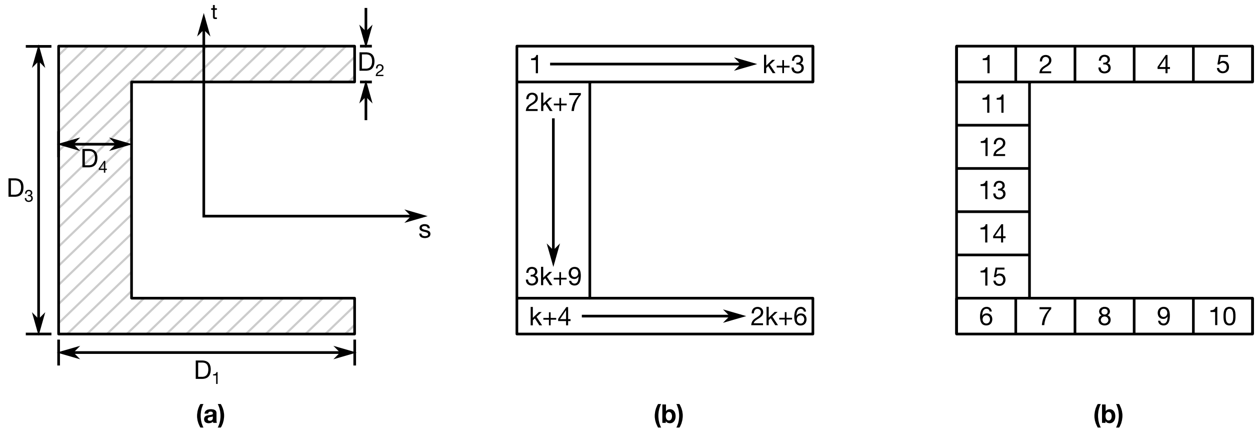


Figure 24-4. Type 2: Channel. (a) Cross section geometry. (b) Integration point numbering. (c) Example for k = 2.

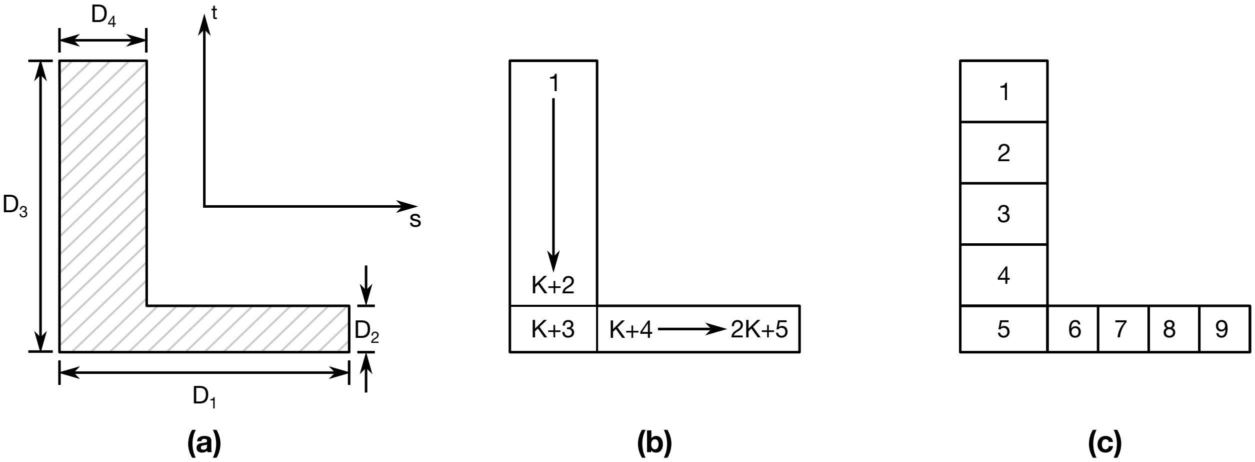


Figure 24-5. Type 3: L-shape. (a) Cross section geometry. (b) Integration point numbering. (c) Example for k = 2.

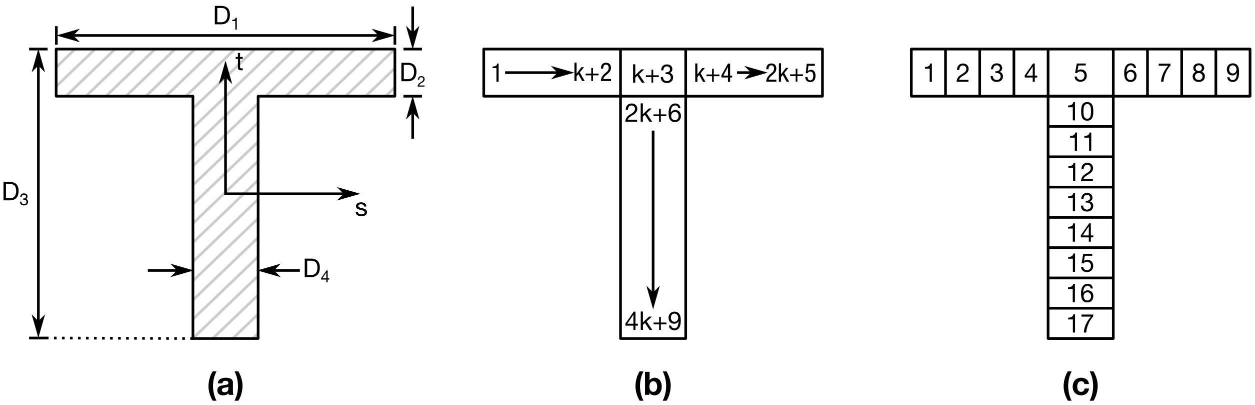


Figure 24-6. Type 4: T-shape. (a) Cross section geometry. (b) Integration point numbering. (c) Example for k = 2.

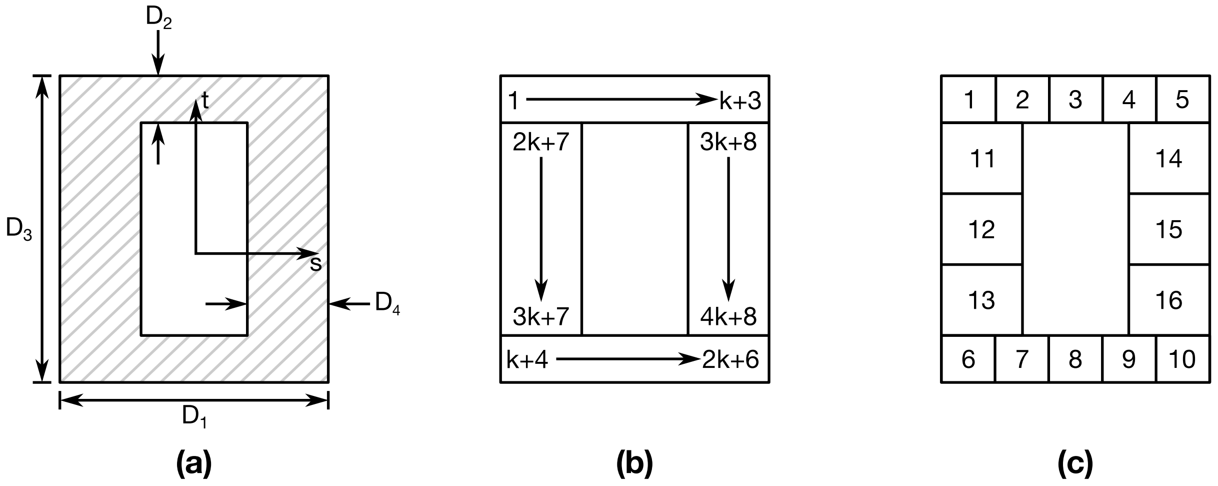


Figure 24-7. Type 5: Box-shape. (a) Cross section geometry. (b) Integration point numbering. (c) Example for k = 2.

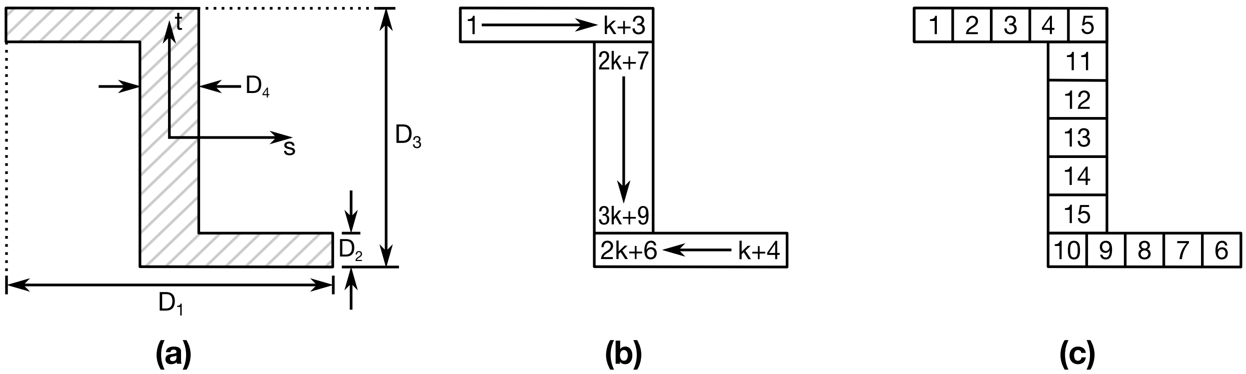


Figure 24-8. Type 6: Z-shape. (a) Cross section geometry. (b) Integration point numbering. (c) Example for k = 2.

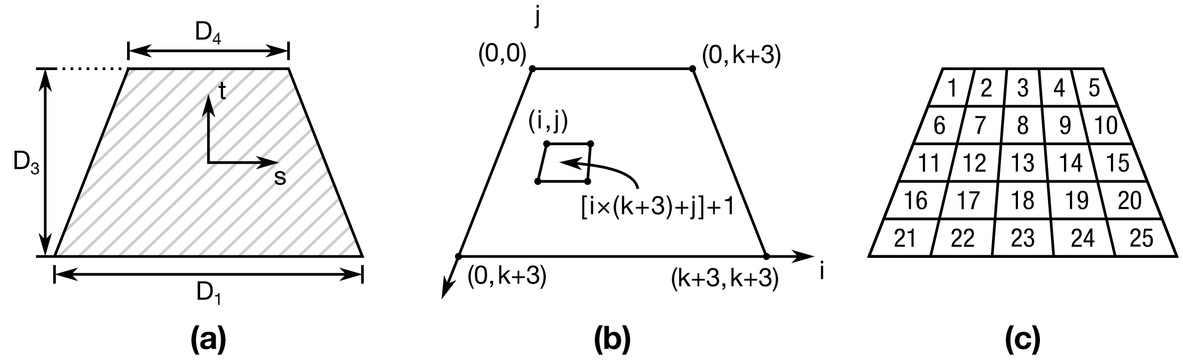


Figure 24-9. Type 7: Trapezoidal. (a) Cross section geometry. (b) Integration point numbering. (c) Example for k = 2.

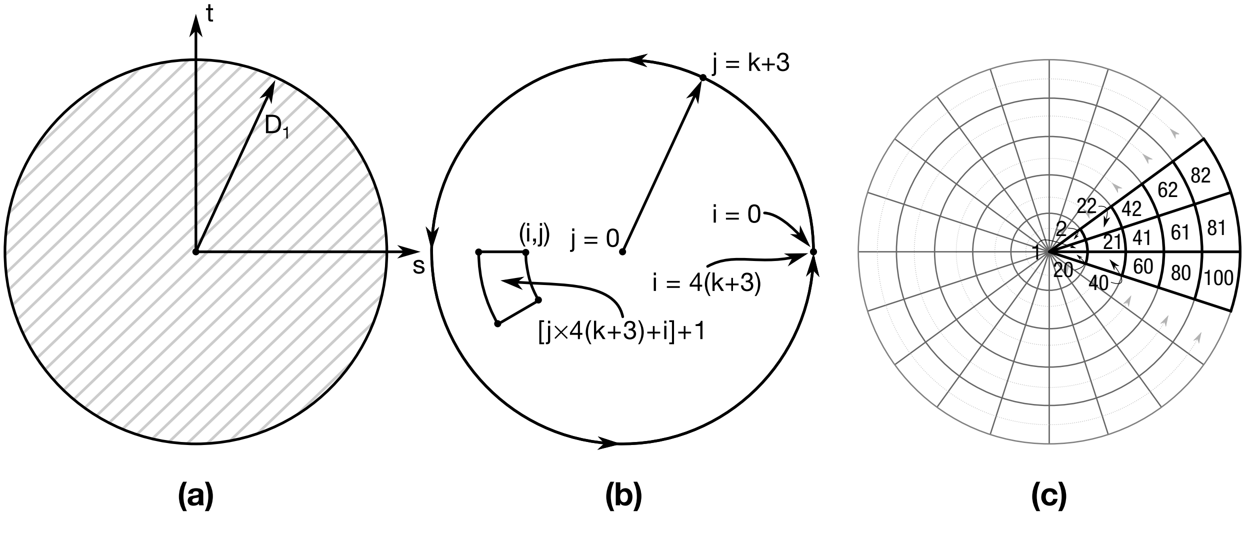


Figure 24-10. Type 8: Circular. (a) Cross section geometry. (b) Integration point numbering. (c) Example for k = 2.

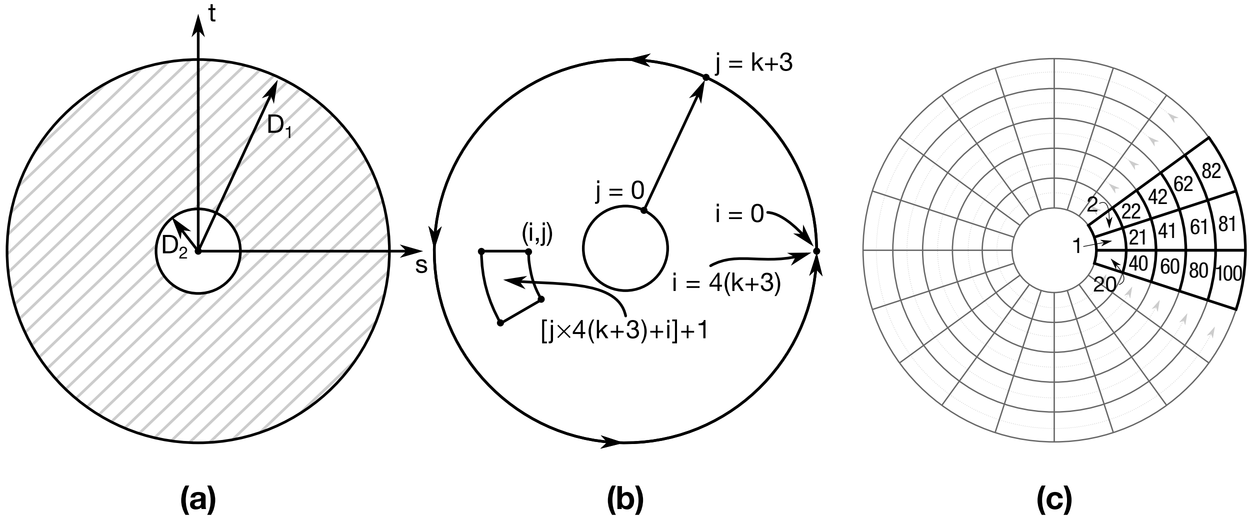


Figure 24-11. Type 9: Tubular. (a) Cross section geometry. (b) Integration point numbering. (c) Example for k = 2.

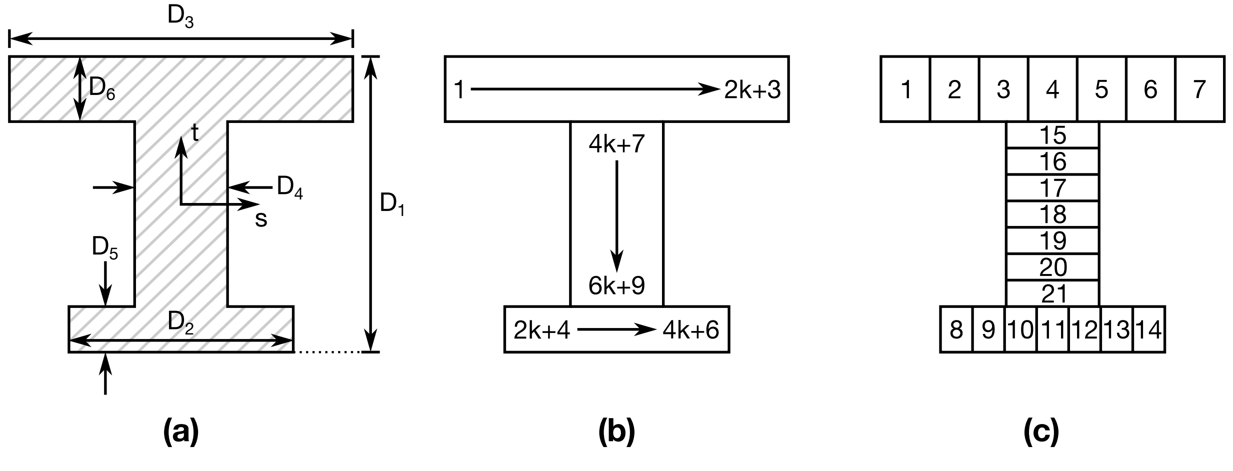


Figure 24-12. Type 10: I-Shape 2. (a) Cross section geometry. (b) Integration point numbering. (c) Example for k = 2.

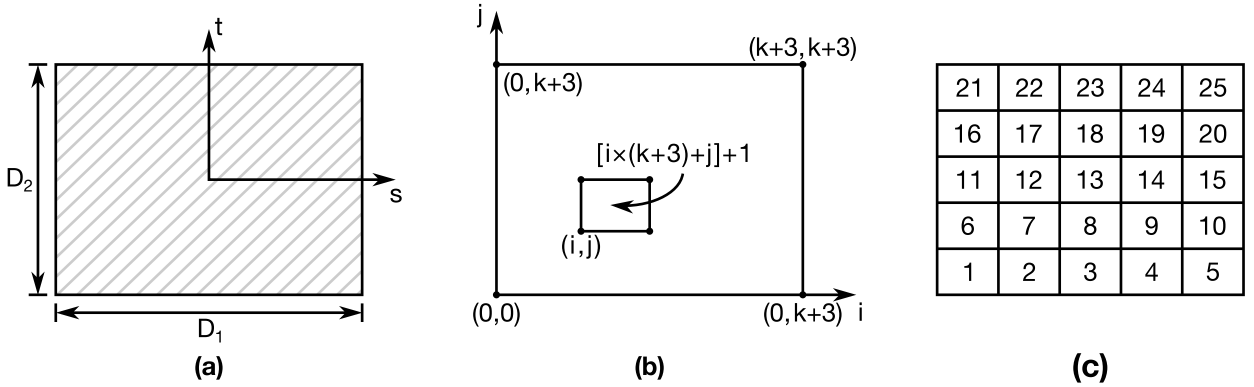


Figure 24-13. Type 11: Solid Box. (a) Cross section geometry. (b) Integration point numbering. (c) Example for k = 2.

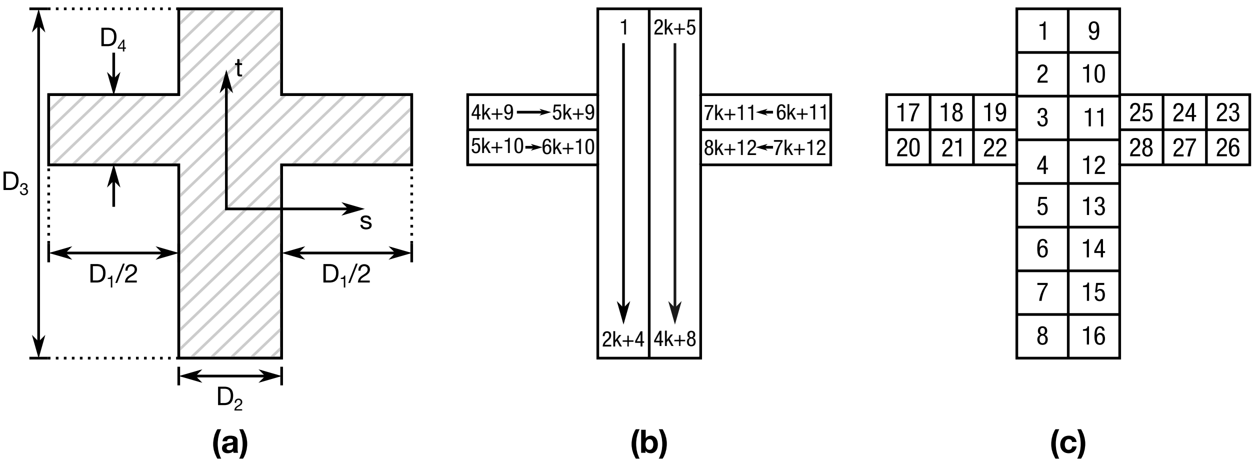


Figure 24-14. Type 12: Cross. (a) Cross section geometry. (b) Integration point numbering. (c) Example for k = 2.

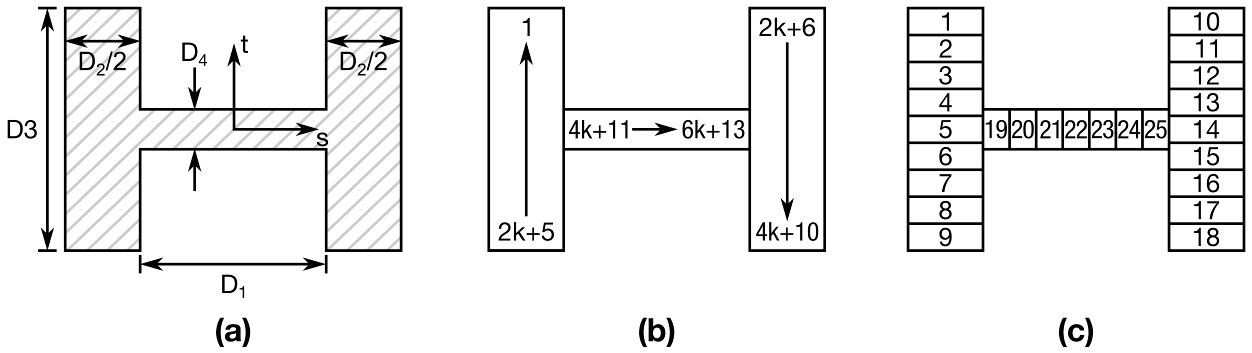


Figure 24-15. Type 13: H-Shape. (a) Cross section geometry. (b) Integration point numbering. (c) Example for k = 2.

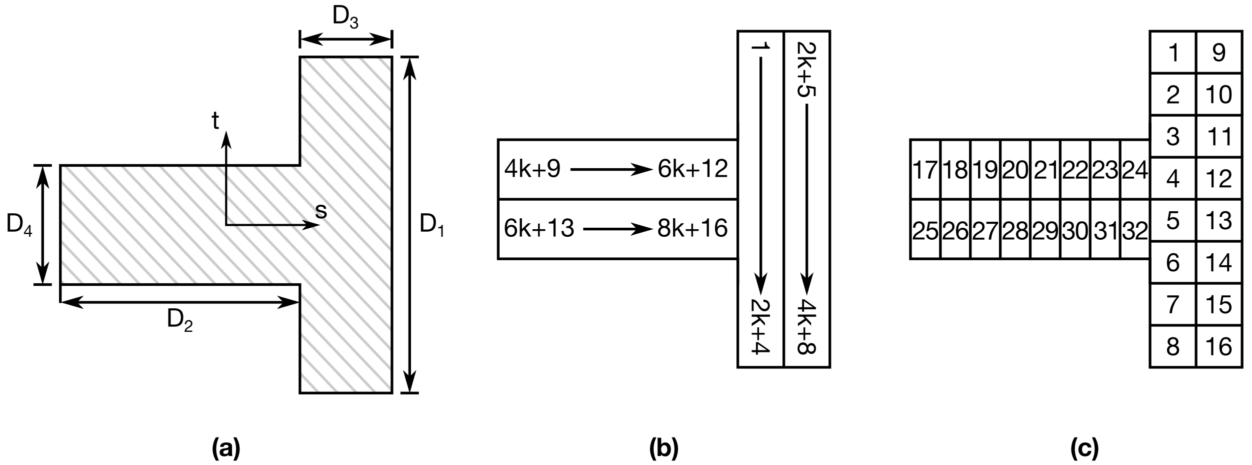


Figure 24-16. Type 14: T-Shape 2. (a) Cross section geometry. (b) Integration point numbering. (c) Example for k = 2.

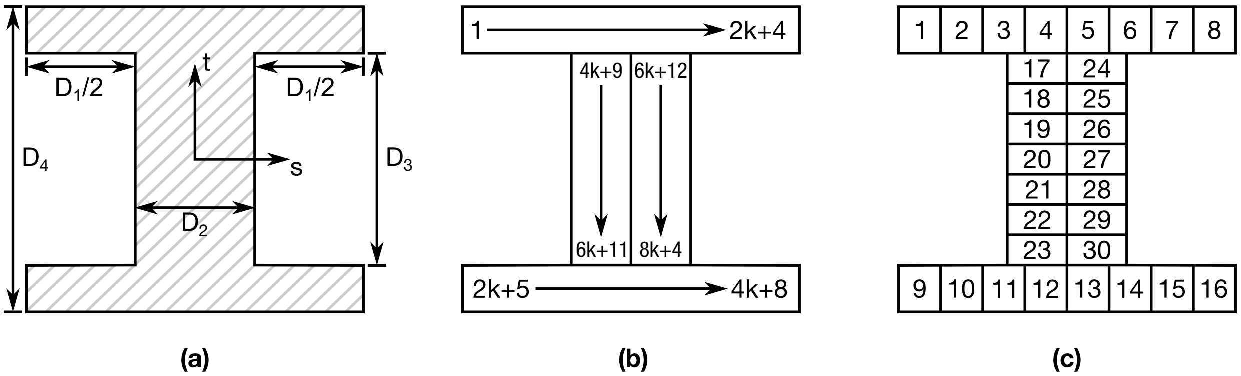


Figure 24-17. Type 15: I-Shape 3. (a) Cross section geometry. (b) Integration point numbering. (c) Example for k = 2.

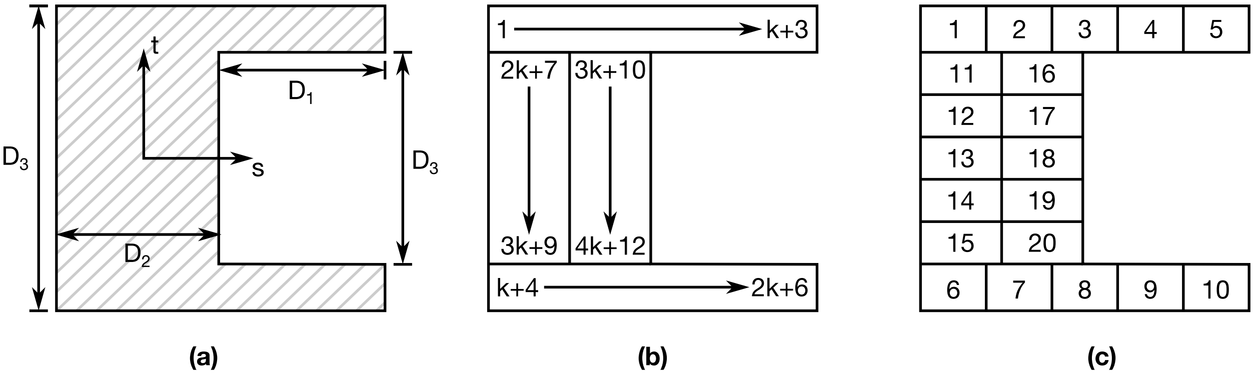


Figure 24-18. Type 16: Channel 2. (a) Cross section geometry. (b) Integration point numbering. (c) Example for k = 2.

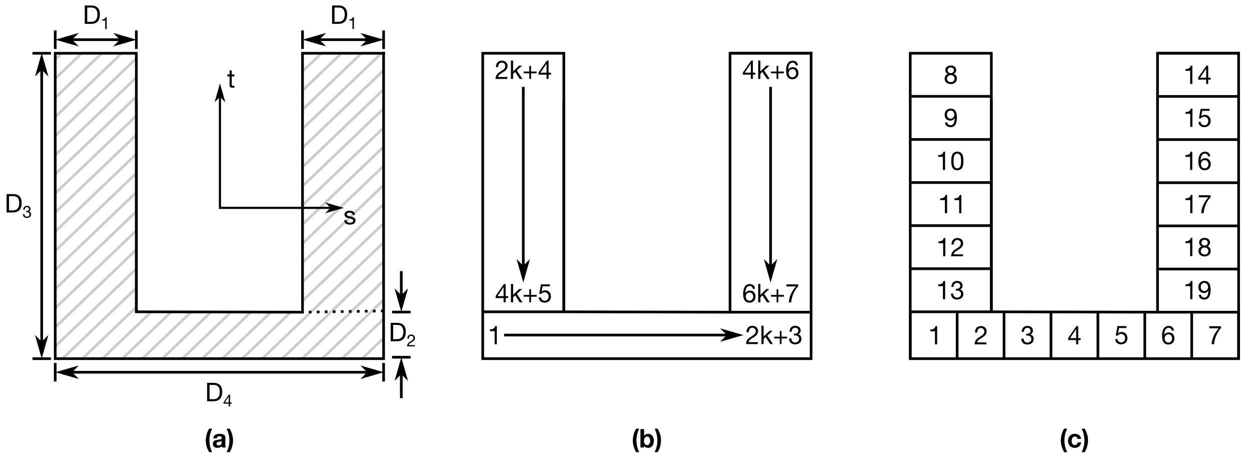


Figure 24-19. Type 17: Channel 3. (a) Cross section geometry. (b) Integration point numbering. (c) Example for k = 2.

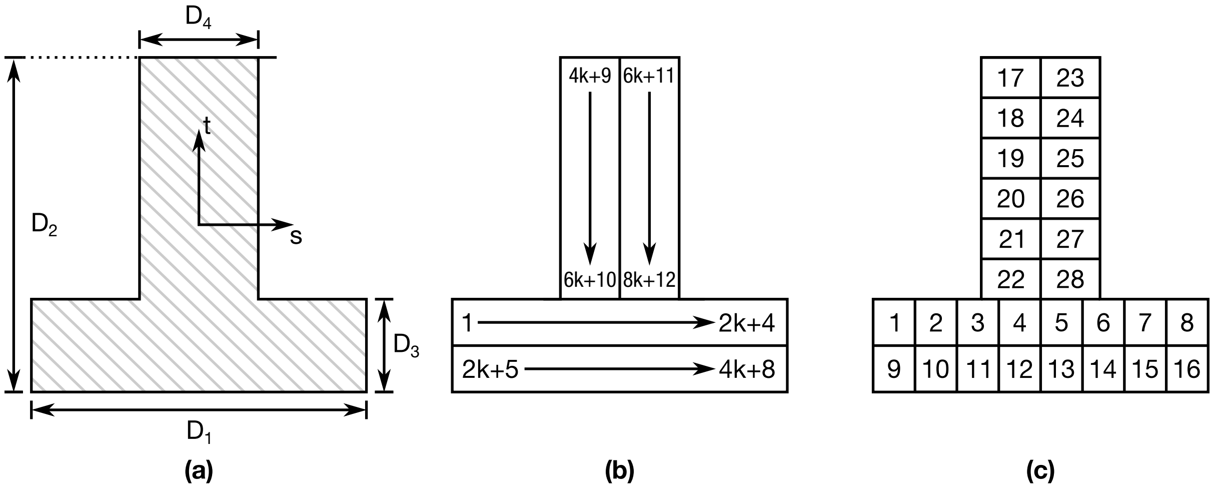


Figure 24-20. Type 18: T-Shape 3. (a) Cross section geometry. (b) Integration point numbering. (c) Example for k = 2.

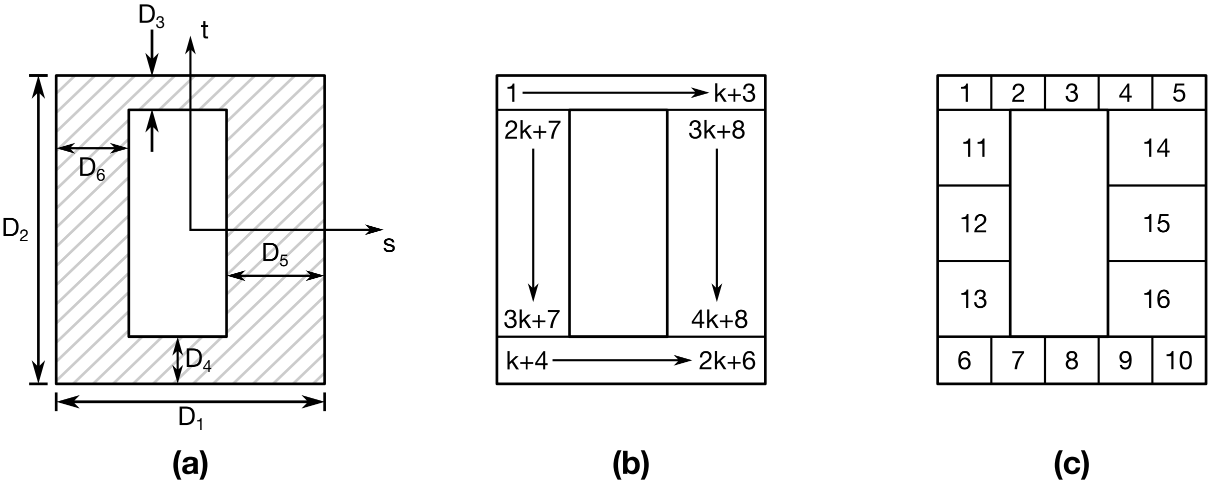


Figure 24-21. Type 19: Box Shape 2. (a) Cross section geometry. (b) Integration point numbering. (c) Example for k = 2.

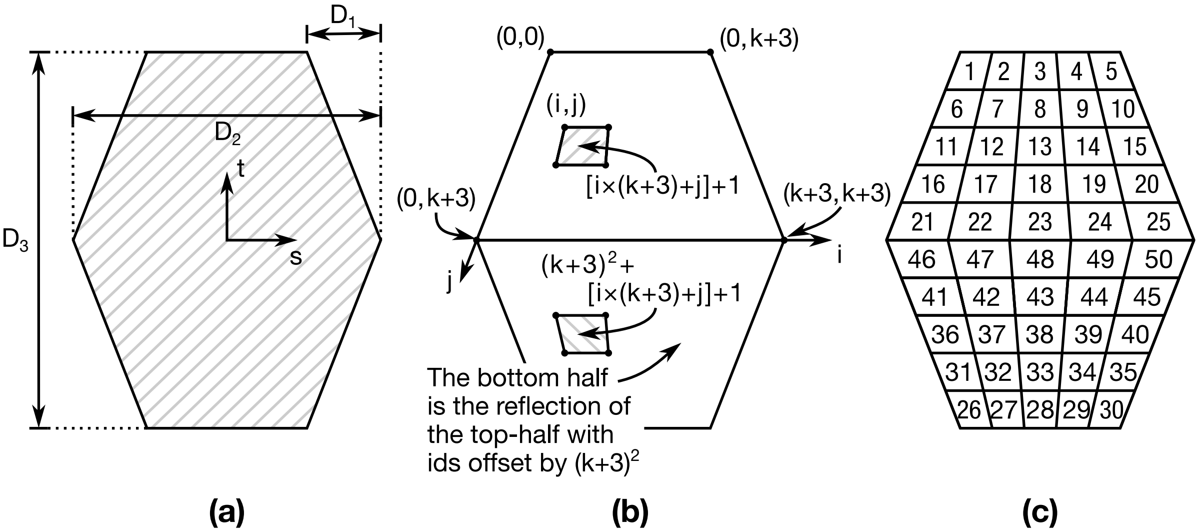


Figure 24-22. Type 20: Hexagon. (a) Cross section geometry. (b) Integration point numbering. (c) Example for k = 2.

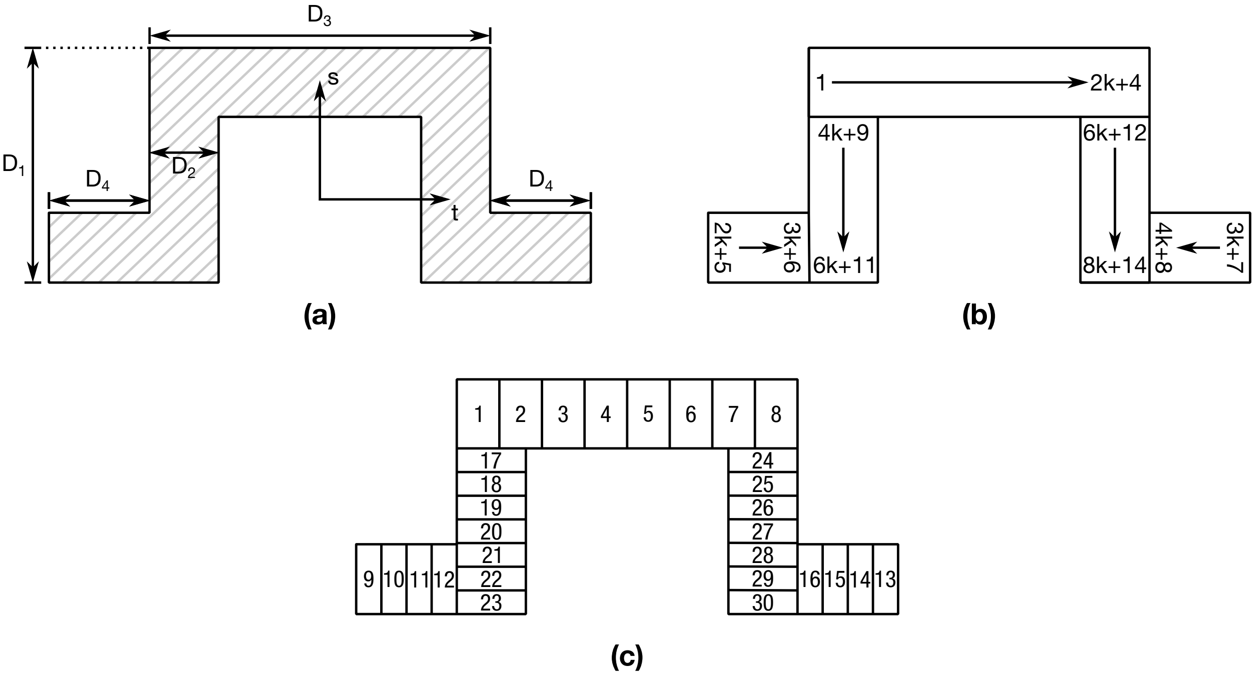


Figure 24-23. Type 21: Hat-Shape. (a) Cross section geometry. (b) Integration point numbering. (c) Example for k = 2.

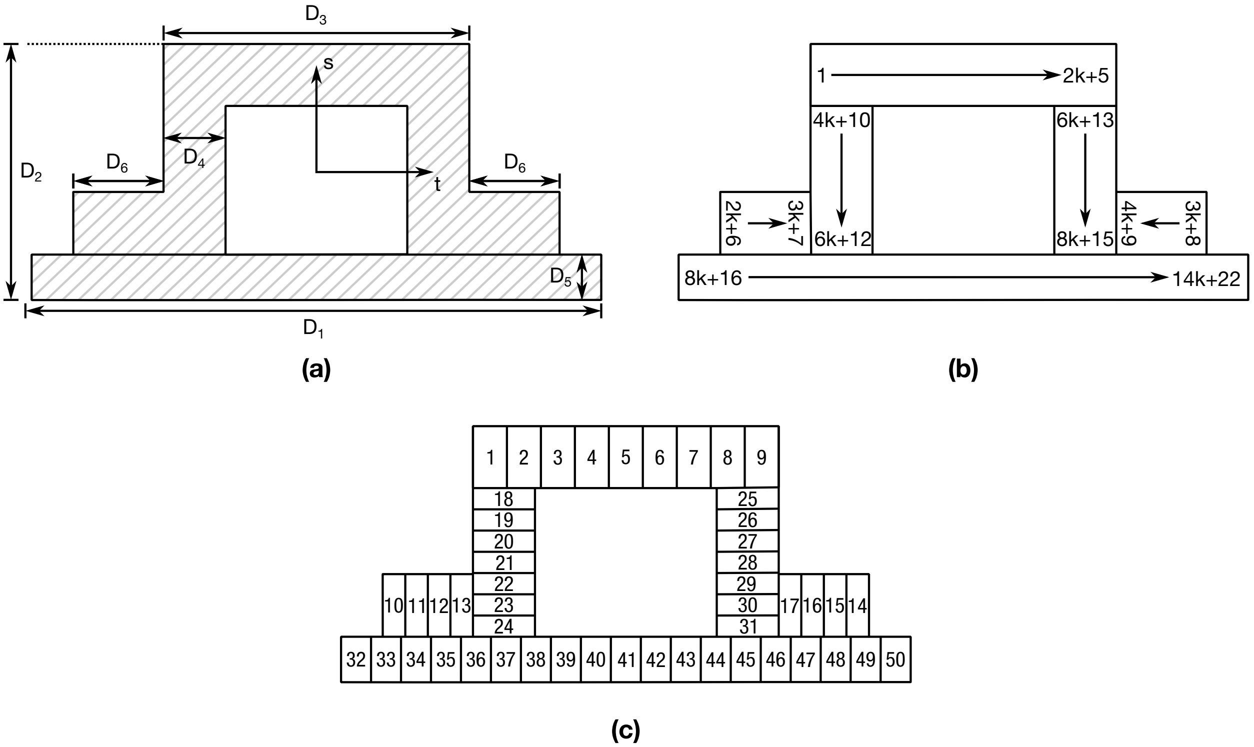


Figure 24-24. Type 22: Hat-Shape 2. (a) Cross section geometry. (b) Integration point numbering. (c) Example for k = 2.

*INTEGRATION_SHELL

Purpose: Define user defined through the thickness integration rules for the shell element. This option applies to three dimensional shell elements with three or four nodes (ELEMENT_SHELL types 1-11 and 16) and to the eight node thick shell (ELEMENT_TSHELL). See *PART_COMPOSITE for a simpler alternative to *PART + *SECTION_SHELL + *INTEGRATION_SHELL.

Card 1	1	2	3	4	5	6	7	8
Variable	IRID	NIP	ESOP	FAILOPT				
Type	I	I	I	I				

Define NIP cards below if ESOP = 0.

Card	1	2	3	4	5	6	7	8
Variable	S	WF	PID					
Type	F	F	I					

VARIABLE	DESCRIPTION
IRID	Integration rule ID (IRID refers to IRID on *SECTION_SHELL card).
NIP	Number of integration points
ESOP	Equal spacing of integration points option: EQ.0: integration points are defined below, EQ.1: integration points are equally spaced through thickness such that the shell is subdivided into NIP layers of equal thickness.

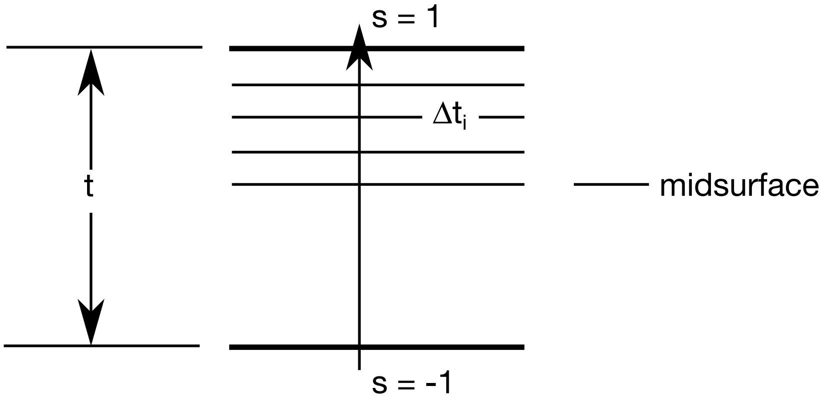


Figure 24-25. In the user defined shell integration rule the ordering of the integration points is arbitrary.

VARIABLE	DESCRIPTION
FAILOPT	Treatment of failure when mixing different constitutive types, which do and do not include failure models, through the shell thickness. For example, consider the case where a linear viscoelastic material model, which does not have a failure option, is mixed with a composite model, which does have a failure option. Note: If the failure option includes failure based on the time step size of the element, element deletion will occur regardless of the value of FAILOPT. EQ.0: Element is deleted when the layers which include failure, fail. EQ.1: Element failure cannot occur since some layers do not have a failure option.
S	Coordinate of integration point in range -1 to 1.
WF	Weighting factor. This is typically the thickness associated with the integration point divided by actual shell thickness, i.e., the weighting factor for the ith integration point $= \frac{\Delta t_i}{t}$ as seen in Figure 24-25.

VARIABLE	DESCRIPTION
PID	Optional part ID if different from the PID specified on the element card. The average mass density for the shell element is based on a weighted average of the density of each layer that is used through the thickness. When modifying the constitutive constants through the thickness, it is often necessary to defined unique part IDs without elements that are referenced only by the user integration rule. These additional part IDs only provide a density and constitutive constants with local material axes (if used) and orientation angles taken from the PID referenced on the element card. In defining a PID for an integration point, it is okay to reference a solid element PID. The material type through the thickness can vary.

*INTERFACE

Interface definitions may be used to define surfaces, nodal lines, and nodal points for which the displacement and velocity time histories are saved at some user specified frequency. This data may then used in subsequent analyses as an interface ID in the *INTERFACE_LINKING_DISCRETE_NODE as master nodes, in *INTERFACE_LINKING_SEGMENT as master segments and in *INTERFACE_LINKING_EDGE as the master edge for a series of nodes. This capability is especially useful for studying the detailed response of a small member in a large structure. For the first analysis, the member of interest need only be discretized sufficiently that the displacements and velocities on its boundaries are reasonably accurate. After the first analysis is completed, the member can be finely discretized in the region bounded by the interfaces. Finally, the second analysis is performed to obtain highly detailed information in the local region of interest. When beginning the first analysis, specify a name for the interface segment file using the Z = parameter on the LS-DYNA execution line. When starting the second analysis, the name of the interface segment file created in the first run should be specified using the L = parameter on the LS-DYNA command line. Following the above procedure, multiple levels of sub-modeling are easily accommodated. The interface file may contain a multitude of interface definitions so that a single run of a full model can provide enough interface data for many component analyses. The interface feature represents a powerful extension of LS-DYNA's analysis capabilities. The keyword cards for this purpose are:

*INTERFACE_COMPENSATION_NEW

INTERFACE_COMPONENT_*OPTION

INTERFACE_LINKING_DISCRETE_NODE_*OPTION

*INTERFACE_LINKING_EDGE

*INTERFACE_LINKING_SEGMENT

INTERFACE_SPRINGBACK_*OPTION1_OPTION2

Interface definitions may also be employed to define soil-structure interfaces in earthquake analysis involving non-linear soil-structure interaction where the structure may be non-linear but the soil outside the soil-structure interface is assumed to be linear. Free-field earthquake ground motions are required only at the soil-structure interface for such analysis. The keyword cards for this purpose are:

*INTERFACE_SSI

*INTERFACE_SSI_AUX

*INTERFACE_SSI_AUX_EMBEDDED

*INTERFACE_SSI_STATIC

INTERFACE_BLANKSIZE_*{OPTION}

Available options include:

DEVELOPMENT

INITIAL_TRIM

INITIAL_ADAPTIVE

SCALE_FACTOR

SYMMETRIC_PLANE

Purpose: This keyword causes LS-DYNA to run a blank-size development calculation instead of a finite element calculation. The input for this feature consists of (1) the result of a completed metal forming simulation, (2) the corresponding initial blank, and (3) the desired result from the simulation in the form of a boundary curve or full mesh. From these inputs the *INTERFACE_BLANKSIZE method adjusts the initial blank so that the resulting formed piece more closely matches the target. The blank's starting geometry may be systematically improved by iterating. A GUI for using this available in *LS-*

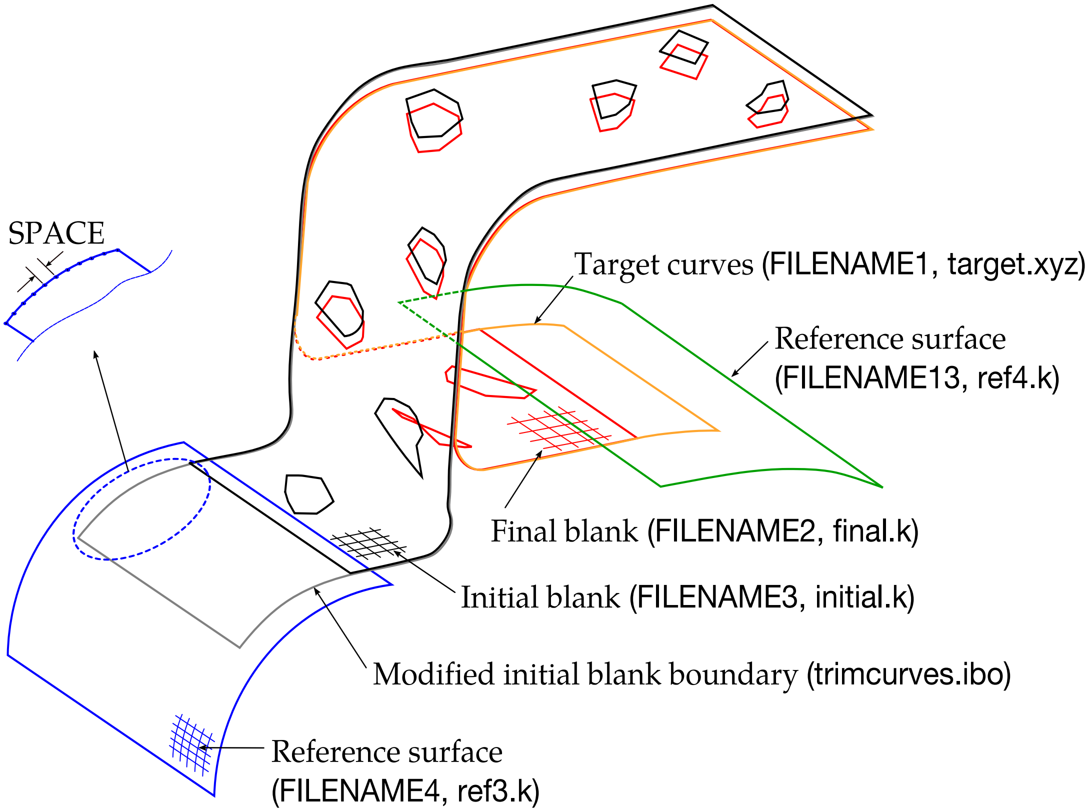

SPACE
Target curves (FILENAME1, target.xyz)
Reference surface
(FILENAME13, ref4.k)
Final blank (FILENAME2, final.k)
Initial blank (FILENAME3, initial.k)
Modified initial blank boundary (trimcurves.ibo)
Reference surface
(FILENAME4, ref3.k)

Figure 25-1. Trim curve development using a reference surface. See Example II.

PrePost as of version 4.2 under APPLICATION → Metal Forming → Blank Size/Trim line.

> **NOTE:** When this card is present LS-DYNA *does not* proceed to the finite element simulation.

This keyword requires one or all three keyword options, each corresponding to a different kind of forming operation:

1. **DEVELOPMENT.** This option takes as its target either a full mesh or a minimum boundary. It adjusts the blank so that he product more closely approximates the target. The computed blank-boundary is written to a file called trimcurves.ibo, which contains a *DEFINE_CURVE_TRIM_3D keyword.

2. **INITIAL_TRIM.** This option adjusts the blank so that trimming and mesh-refinement are mapped back onto the initial blank.

3. **INITIAL_ADAPTIVE.** This option reads in (1) the input mesh for a flanging simulation as well as (2) the adapted mesh calculated during flanging simulation. It maps the refinement back to the initial blank.

Card set for *INITIAL_BLANKSIZE_DEVELOPMENT.

Development Parameter Card.

Card 1	1	2	3	4	5	6	7	8
Variable	IOPTION		IADAPT	MAXSIZE	REFERENC	SPACE	MAXGAP	ORIENT
Type	I		I	F	I	F	F	I
Default	-2		1	30.0	none	2.0	30.0	none

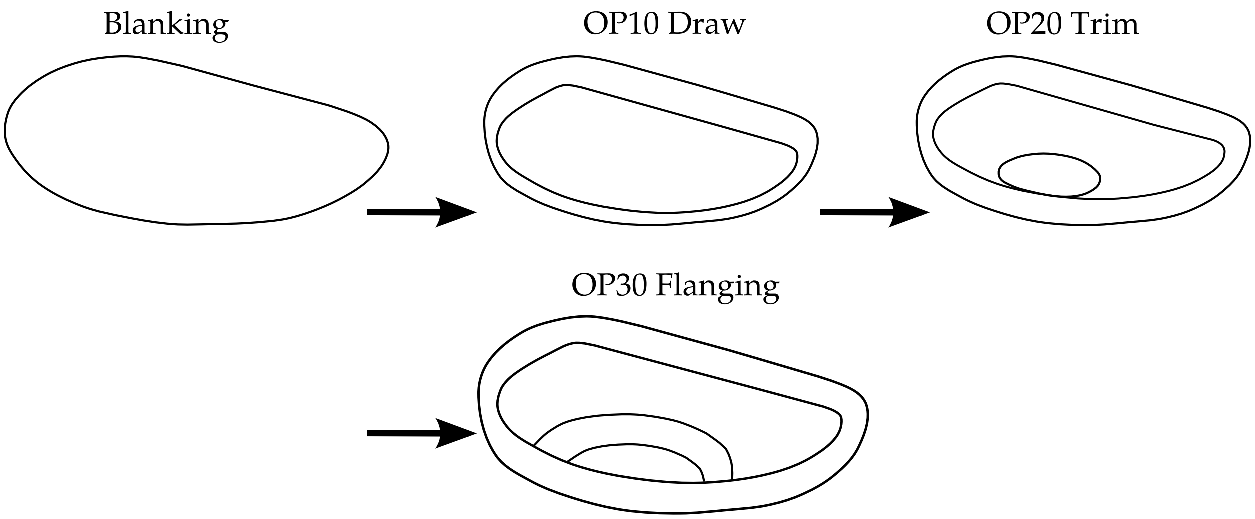


Figure 25-2. A stamping process consists of draw, trim and flanging (*Courtesy of Metal Forming Analysis Corporation*). The labels *OP10*, *OP20*, and *OP30* are used *extensively* in the ensuing discussion.

Target Card. See "*Target curves (target.xyz)*" in Figure 25-1.

Card 2	1	2	3	4	5	6	7	8
Variable	FILENAME1							
Type	A80							
Default	none							

Final Blank Card. See "*Final blank (final.k)*" in Figure 25-1.

Card 3	1	2	3	4	5	6	7	8
Variable	FILENAME2							
Type	A80							
Default	none							

Initial Blank Card. See *"Initial blank (initial.k)"* in Figure 25-1.

Card 4	1	2	3	4	5	6	7	8
Variable	FILENAME3							
Type	A80							
Default	none							

Reference Surface Card. See *"Reference surface (ref3.k)"* in Figure 25-1 and Example II.

Card 4	1	2	3	4	5	6	7	8
Variable	FILENAME4							
Type	A80							
Default	none							

Reference Surface Card. See *"Reference surface (ref4.k)"* in Figure 25-1 and Example II.

Card 4	1	2	3	4	5	6	7	8
Variable	FILENAME13							
Type	A80							
Default	none							

Card set for *INTERFACE_BLANKSIZE_INITIAL_TRIM.

Initial Flat Blank. FILENAME5 should specify the mesh of a blank that has been refined during a subsequent forming operation. FILENAME5 is usually set to the adapt.msh output. See *"OP10 Initial Adapted blank mesh"* in Figure 25-3.

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME5							
Type	A80							
Default	none							

Formed blank. FILENAME6 specifies a dynain file from a forming simulation. See *"OP10 Final Blank"* in Figure 25-3.

Card 2	1	2	3	4	5	6	7	8
Variable	FILENAME6							
Type	A80							
Default	none							

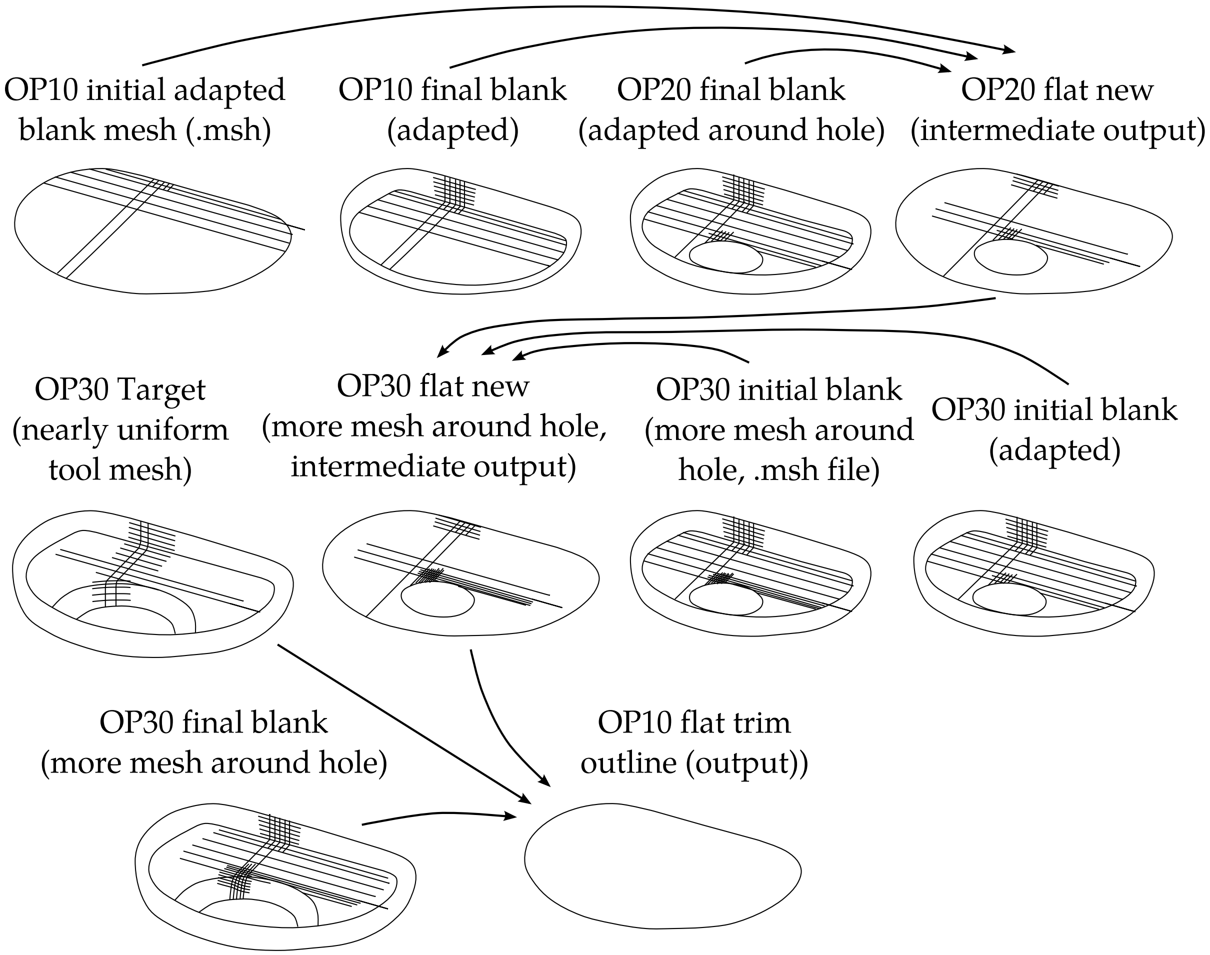

OP10 initial adapted blank mesh (.msh)
OP10 final blank (adapted)
OP20 final blank (adapted around hole)
OP20 flat new (intermediate output)
OP30 Target (nearly uniform tool mesh)
OP30 flat new (more mesh around hole, intermediate output)
OP30 initial blank (more mesh around hole, .msh file)
OP30 initial blank (adapted)
OP30 final blank (more mesh around hole)
OP10 flat trim outline (output))

Figure 25-3. Inputs and outputs (*Courtesy of Metal Forming Analysis Corp.*). For exposition of labels *OP10*, *OP20*, and *OP30* see Figure 25-2.

Trimmed Formed Blank. A dynain file from a trimming simulation that started with the state given in FILENAME6. See "*OP20 Final Blank*" in Figure 25-3.

Card 3	1	2	3	4	5	6	7	8
Variable	FILENAME7							
Type	A80							
Default	none							

Trimmed Flat Blank (output). This field specifies the name for the file to which the trimmed flat blank is written. See "*OP20 Flat New*" in Figure 25-3.

Card 4	1	2	3	4	5	6	7	8
Variable	FILENAME8							
Type	A80							
Default	none							

Card set for *INTERFACE_BLANKSIZE_INITIAL_ADAPTIVE.

Flat Blank. FILENAME9 specifies the mesh of a blank in a flat configuration serving as the basis for a two-step metal forming process. The second-stage simulation produces an adapt.msh file, from which the refinements are to be mapped onto the flat-blank of FILENAME9. See, for example, "*OP20 Flat New*" in Figure 25-3 where FILENAME9 points to result, FILENAME8, of an INITIAL_TRIM calculation.

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME9							
Type	A80							
Default	none							

Initial Blank. FILENAME10 is the result from the first stage of a two-step process. See, for example, "*OP30 Initial Blank*" in Figure 25-3 where FILENAME10 has been formed and trimmed and refined along the way.

Card 2	1	2	3	4	5	6	7	8
Variable	FILENAME10							
Type	A80							
Default	none							

Adapted Initial Blank. FILENAME11 contains a refined version of the mesh in FILENAME10. It is expect to name the adapt.msh file from some operation performed on the mesh of FILENAME10. See, for example, "*OP30 Initial Blank (with more mesh)*" in Figure 25-3, where the adapt.msh file comes from a flanging simulation.

Card 3	1	2	3	4	5	6	7	8
Variable	FILENAME11							
Type	A80							
Default	none							

Refined Flat Blank (output). This field specifies the name of the file to which the refined flat blank is written. The blank's mesh is refined to *exactly* match the forming process. See, for example, "*OP30 Flat New*" in Figure 25-3.

Card 4	1	2	3	4	5	6	7	8
Variable	FILENAME12							
Type	A80							
Default	none							

Card set for *INTERFACE_BLANKSIZE_SCALE_FACTOR.

Scale Factor Card. Define one card for each curve. Include as many cards in the following format as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	IDCRV	SF	OFFX	OFFY	OFFZ			
Type	I	F	F	F	F			
Default	1	0.0	0	0	0			

Card set for *INTERFACE_BLANKSIZE_SYMMETRIC_PLANE.

Symmetric Plane Card.

Card 1	1	2	3	4	5	6	7	8
Variable	X0	Y0	Z0	V1	V2	V3		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	1.0	0.0	0.0		

VARIABLE	DESCRIPTION
IOPTION	Target curve definition input type: EQ.1: (entire) blank mesh in keyword format. EQ.2: consecutive position coordinates of blank boundary loop curve in XYZ format, defined by *DEFINE_TARGET_BOUNDARY. The blank mesh's normal vector and the closed boundary curve are consistently oriented according to the right-hand rule, see Figure 25-4. Note the target boundary curves must have enough points for a successful prediction of the initial blank size. Starting in Revision 100589, curve direction is desensitized, meaning both IOPTION = 2 and IOPTION = -2 will give the same results. EQ.-2: consecutive position coordinates of blank boundary loop in XYZ format, defined by *DEFINE_TARGET_BOUNDARY. The blank mesh's normal vector and the closed boundary curve are consistently oriented according to the left-hand rule, see Figure 25-4. Note the target boundary curves must have enough points for a successful prediction of the initial blank size. Starting in Revision 100589, curve direction is desensitized, meaning both IOPTION = 2 and IOPTION = -2 will give the same results. In *LS-PrePost 4.0*, menu option *GeoTol* → *ID Measure* can be used to show the flow direction of a boundary curve. *Curve* → *Rever(se)* can be used to reverse the curve direction.

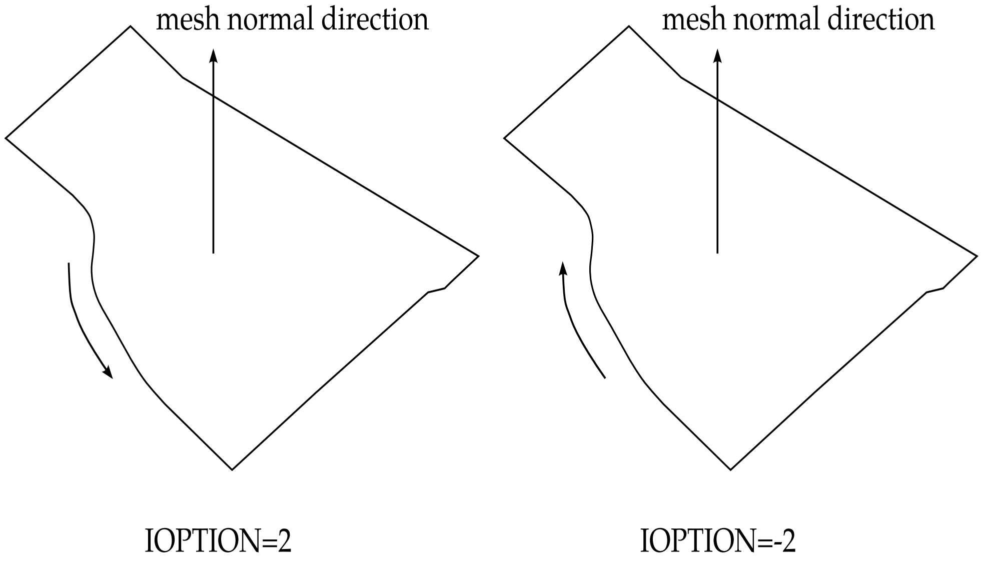


Figure 25-4. Differences between IOPTION 2 and -2. Note Starting in Revision 100589, curve direction is desensitized, meaning both IOPTION = 2 and IOPTION = -2 will give the same results.

VARIABLE **DESCRIPTION**

IADAPT

Adaptive mesh control flag. If IADAPT = 1, number of elements between initial (FILENAME3) and simulated blank (FILENAME2) meshes can be different, so it is not necessary to use the sheet blank from the file "adapt.msh" (created by setting IOFLAG = 1 in *CONTROL_ADAPTIVE) for the initial blank mesh.

MAXSIZE

The maximum change in initial blank size in each iteration. It is used to limit the blank size change in each iteration during mapping to avoid convergence problems when the initial blank is curved.

REFERENC

Flag to indicate trim curve projection to a reference surface (mesh), see Figure 25-1:

EQ.0: no projection.

EQ.1: the trim curves will be projected to the reference surface. In addition, the mesh file for the reference surface is given in FILENAME4.

VARIABLE DESCRIPTION

SPACE

Point spacing distance on the reference surface for the projected curve, see Figure 25-1. If the gap between two neighboring points in the modified trimming curve is larger than this value, extra nodes will be added in the final blank and projected to the reference surface. Smaller value should be used for large reference surface curvature.

MAXGAP

When REFERENC is set to "1", the nodes from the final blank will be projected to the reference surface. However, if the distance between the nodes and the surface is larger than MAXGAP, the nodes will not be projected.

ORIENT

A flag to control iterative optimization efficiency or to include a reference surface file for the final formed blank.

EQ.1: activates a new algorithm to potentially reduce the number of iterations during iterative blank size optimization loop, to be used in conjunction with the option SCALE_FACTOR (0.75~0.9).

EQ.2: to include a reference surface FILENAME13 (in keyword mesh format) for the simulated (formed) blank. This is useful when formed blank is smaller than intended and extended surface are not flat. See Remarks and Figure 25-1.

FILENAME1

Target input file name. When a blank mesh is used (IOPTION = 1), the keyword file must contain *NODE and *ELEMENT_SHELL keywords. When using the blank boundary curve for target (|IOPTION| = 2), the file must consist of *DEFINE_TARGET_BOUNDARY. See the Target Boundaries (and IGES) in the remarks section.

FILENAME2

Simulated (formed or flanged) sheet blank mesh in keyword format. This mesh can be obtained from the final state of any downstream process simulation.

FILENAME3

Initial sheet blank mesh in keyword format. This can be the first state mesh from any process simulation prior to FILENAME2 simulation. Set IADAPT = 1 if adaptive refinement is used in any simulation.

VARIABLE	DESCRIPTION
FILENAME4	Reference surface onto which adjustments to the blank's trim curves in its initial state are projected (*ref3.k* in Figure 25-1). This surface is typically a curved extension of the initial blank and must be defined as mesh in keyword format. This file name must be defined when REFERENC is set to 1. Also see extended reference surface for the final state (formed state) - FILENAME13.
FILENAME5	Initial blank in its flat configuration *with* adapted mesh in keyword format. FILENAME5 usually points to an adapt.msh file. For example see OP10 in Figures 25-2, 25-22 and 25-3, in which FILENAME5 is the adapt.msh from a draw-forming calculation.
FILENAME6	Final formed blank in keyword format. This is usually the dynain file corresponding to the adapt.msh file mentioned above for FILENAME5, see Figures 25-2, 25-22 and 25-3.
FILENAME7	A trimmed blank in keyword format. This file should be derived from FILENAME6 as the dynain file from a trimming simulation. See OP20 in Figures 25-2, 25-22 and 25-3.
FILENAME8	This field specifies the name for the file in which the trimmed flat blank is to be written. See OP20 in Figures 25-2, 25-22 and 25-3.
FILENAME9	FILENAME9 should point to the blank defined by FILENAME8, as in Figures 25-2, 25-22 and 25-3.
FILENAME10	Initial-stage result file name. This may be extracted from d3plot files using *LS-PrePost* or it may be generated by LS-DYNA as a dynain file. See, for example, "*OP30 Initial Blank*" in Figure 25-3 where FILENAME10 has been formed, trimmed and refined along the way. To obtain from d3plot file the necessary mesh in keyword format using *LS-PrePost4.0* select POST → OUTPUT → Dynain ASCII and check the box for "Exclude strain and stress".
FILENAME11	FILENAME11 contains a refined version of the mesh in FILENAME10. FILENAME11 can be obtained from adapt.msh file from the same operation performed on the mesh of FILENAME10. See, for example, "*OP30 Initial Blank (with more mesh)*" in Figure 25-3, where the adapt.msh file comes from a flanging calculation.

VARIABLE	DESCRIPTION
FILENAME12	This field specifies the name of the file to which the refined flat blank is written. The blank's mesh is refined to *exactly* match the forming process. See, for example, "*OP30 Flat New*" in Figure 25-3.
FILENAME13	Reference surface onto which adjustments to the blank's trim curves in its final state are projected (*ref4.k* in Figure 25-1). This surface is typically a curved extension of the formed blank and must be defined as mesh in keyword format. This file name must be defined when ORIENT is set to 2.
IDCRV	Curve ID in the order of appearance as in FILENAME1 in the target card, as defined by *DEFINE_TARGET_BOUNDARY.
SF	Scale factor for the IDCRV defined above. It defines a fraction of the changes required for the predicted initial blank shape. For example, if SF is set to "0.0" the corresponding IDCRV will be excluded from the calculation (although the original initial curve still will be output); on the other hand, if SF is set to "1.0", full change will be applied to obtain the modified initial blank that reflects the forming process. A SF of 0.5 will apply 50% of the changes required to map the initial blank. This feature is especially important for inner holes that are small and hole boundary expansions are large, so the predicted initial hole can avoid "crisscross" situation. An example is provided in Scale Factor and Symmetric Plane.
OFFX, OFFY, OFFZ	Translational move of the target curve. This is useful when multiple target curves (e.g. holes) and formed curves are far away from each other. Input values of OFFX, OFFY and OFFZ helps establish one-to-one correspondence between each target curve and formed curve. OFFX.EQ.-10000.0: offset values are automatically calculated.
X0, Y0, Z0	x, y, z coordinates of a point on the symmetric plane. See example in Scale Factor and Symmetric Plane.
V1, V2, V3	Vector components of the symmetric plane's normal. See example in Scale Factor and Symmetric Plane.

	Compulational Cost	Accuracy	Information Required	Physical Process
Blanksize	Full simulation	Exact	Full Simulation	Any
Unflanging	Fast	Approximate	Process Geometry	Inverse Flanging
Onestep	Fast	Approximate	None (*Path independent*)	Any

Table 25-1. Comparison of inverse methods.

Inverse Methods for Optimizing Blank Size and Trim Lines:

Finding the minimal practicable blank size and developing an optimal set of trim lines is an integral part of the die engineering process. This card, *INTERFACE_BLANKSIZE, is one of several features that have been developed to solve the inverse problem: that is to calculate an initial blank or blank boundary that will yield a desired product based mostly on the target geometry of that final product.

1. **The One-Step Method.** The *CONTROL_FORMING_ONESTEP card is suitable for early blank-size estimates. It invokes the total-strain theory of plasticity thereby bypassing the, as of yet, undetermined specific details of the forming process.

2. **Unflanging.** Once product design and process plan are complete, die development begins with addendum and binder creation, followed by secondary tooling development. In this stage, *CONTROL_FORMING_UNFLANGING can be used to develop trim lines for the secondary tooling; final (or intermediate) desired flange shapes are unfolded onto the addendum or binder to obtain the corresponding flange shapes in its initial shape. It also implements failure criteria to arrive at suggested final flange curves, with strains and thickness output on the unfolded flanges.

3. **Interface Blanksize.** This card, *INTERFACE_BLANKSIZE, can be used to *accurately* determine the optimal initial blank. To do so it requires on initial configuration, the corresponding simulated configuration, and a desired target configuration. This method takes into account the entire metal-forming process.

    a) One application of this keyword is to map trim curves between dies to calculate the trim curves needed for all trim dies. An example of the application can be found in Figures 25-18 through 25-21.

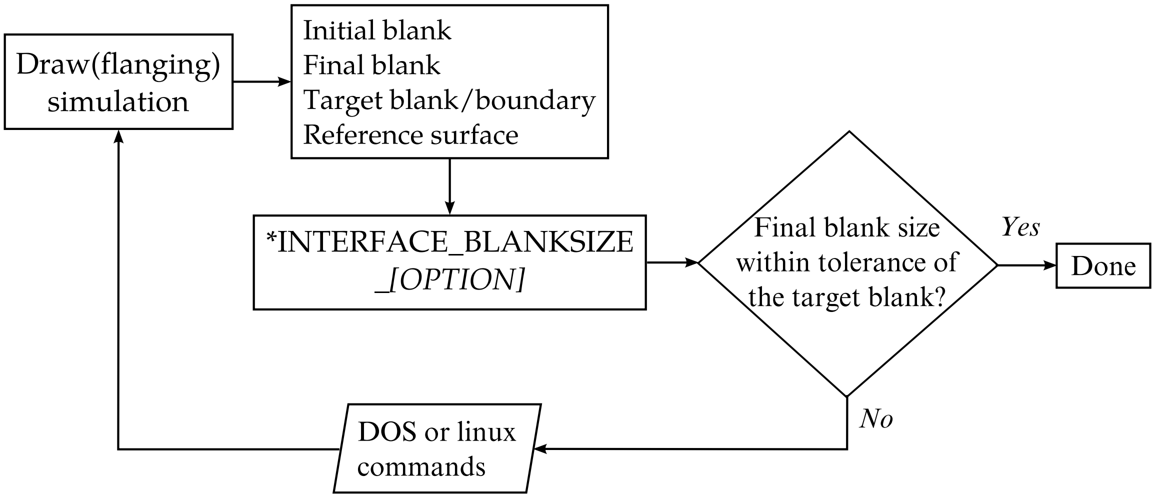

Draw(flanging) simulation
Initial blank
Final blank
Target blank/boundary
Reference surface
*INTERFACE_BLANKSIZE _[OPTION]
Final blank size within tolerance of the target blank?
Yes
Done
No
DOS or linux commands

Figure 25-5. An iterative blank size development process.

b) The keyword can also be used to determine the precise minimal initial blank needed for a draw panel whose blank edge must be at specified distances from the edge of the draw beads.

Iterative Workflow:

The "interface blanksize" command produces an adjusted initial blank. It is not to be expected that the adjustment will be exact. However, this initial blank shape can be run through a second simulation to see if the final shape is close enough to the target blank. If it is not close enough, then the results from the second simulation can be used to repeat the process. Iterations can proceed until the final shape is within the range of the target shape. This iterative blank size development process is schematically presented in Figure 25-5 and exemplified in Figures 25-8 through 25-17.

Target Boundaries (and IGES):

When IOPTION = 2, or -2, a file with the keyword *DEFINE_TARGET_BOUNDARY must be present. This keyword can now be created from an IGES file using *LS-PrePost4.1 (or 4.2)*. To convert from IGES curves to keyword *DEFINE_TARGET_-BOUNDARY in *LS-PrePost4.1*, use menu option *Curve* → *Convert*→ *Method (To Keyword)* → Select *DEFINE_TARGET_BOUNDARY; pick the curves then hit "To Key"; write out the keyword file using File → Save as → Save Keyword As, and select "Output Version" as "V971_R7". In *LS-PrePost 4.2*, a GUI interface was developed located at Application/Metal Forming/Blanksize Trimline so users can import the target curve directly in IGES format, the initial and final blank mesh and then write out a complete LS-DYNA input deck.

In addition, the target curves should be projected onto the final blank mesh if they do not exactly lie on the mesh surface. This can be done with *LS-PrePost4.1* via the menu option *GeoTol* → *Project* → *Project*, select *Closest Projection*, select *Project to Elements*, then define the destination mesh and source curves, and hit *Apply*. 3-D projection in *LS-PrePost4.2* can be critical in obtaining a perfectly smooth predicted initial boundary curve trimcurves.ibo after the LS-DYNA run. Note the projected target curves should be used to import into the GUI.

Computed Initial Blank Boundaries (and IGES):

Computed boundary curves are written with *DEFINE_CURVE_TRIM_3D keyword into a file called trimcurves.ibo. The format of this file follows the keyword's specification. *LS-PrePost4.0* can convert the computed curve to IGES. See Figure 25-10. After hitting *Apply*, the curves will show up in the graphics window, and *File* → *Save as* → *Save Geom as* can be used to write the curves out in IGES format. In the *LS-PrePost4.2* GUI, under *Results*, trimcurves.ibo can be directly imported into the graphics window for viewing and to save in either STEP or IGES format.

To convert IGES to the *DEFINE_CURVE_TRIM_3D keyword format import the IGES file into *LS-PrePost4.0*, and follow the procedures shown in Figure 25-11. After finishing step 2, "*curves have been converted to keyword format*" will be reported in the command prompt. Then use *File* → *Save* → *Save keyword* to write out the keyword file.

Support for Multi-Stage Processes with the Development Option:

Original Implementation.

Prior to Revision 88708 the development option required that the final blank (FILENAME2) differ from the initial blank (FILENAME3) by no more than a deformation and mesh refinement. In practice, this means that the two meshes must come from the same process simulation. For example, in a draw, trim and flanging process, the trimmed panel mesh is used for flanging simulation. Therefore, with the original implementation, LS-DYNA required that the initial blank state (FILENAME3) be trimmed when the final blank state (FILENAME2) is flanged on trimmed panel. Failure to observe this limitation may result in error termination.

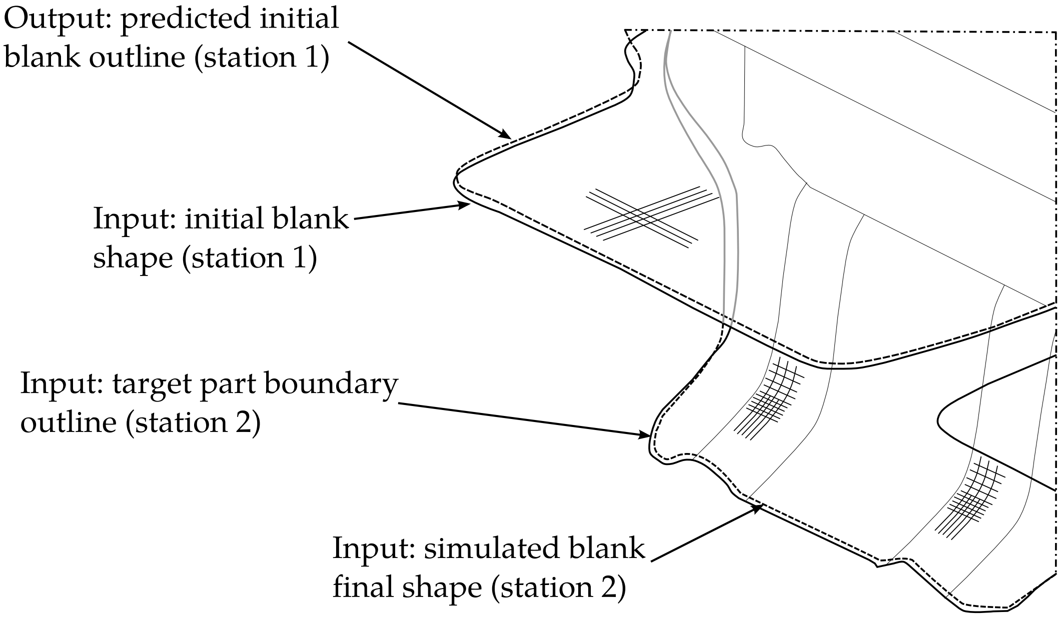

Output: predicted initial blank outline (station 1)
Input: initial blank shape (station 1)
Input: target part boundary outline (station 2)
Input: simulated blank final shape (station 2)

Figure 25-6. Blank size development in a progressive die with IADAPT = 1 in Example I

Enhanced Implementation.

A more recent improvement to the blank size development (Revision 88708) removes the requirement that initial (FILENAME3) and final (FILENAME2) blanks must be from the same process simulations. The initial and final blank states *may* differ by a trimming process. This allows trimming and other process such as flanging to occur between the initial and final blanks, without the need of invoking the INITIAL_TRIM and INITIAL_ADAPTIVE options. For example, the initial blank can be the blank mesh from "Blanking" in Figure 25-2, and the final blank can be the blank mesh from "Flanging," which is also in Figure 25-2.

Scale Factor and Symmetric Plane:

An example of using various scale factors ranging from 0.0 to 1.0 on a model involving an initial hole shape is shown in Figure 25-25. The target curve is given in targetline.k. All nodes along the symmetric plane are constrained by the SYMMETRIC_PLANE option. The symmetric plane is defined going through point coordinates (-76, 2.63844, 0.38) with plane normal vector of (1.0, 0.0, 0.0). A complete input is provided below:

```
*KEYWORD
*INTERFACE_BLANKSIZE_DEVELOPMENT
$  IOPTION              IADAPT
        -2                   1
$ target boundary curves
targetline.k
$ final formed mesh
final.k
$ initial mesh
```

```
initial.k
*INTERFACE_BLANKSIZE_SCALE_FACTOR
$    IDCRV        SF
         1       0.2
*INTERFACE_BLANKSIZE_SYMMETRIC_PLANE
$       X0        Y0        Z0        V1        V2        V3
       -76   2.63844      0.38       1.0       0.0       0.0
*END
```

If taregetline.k consists of multiple curves, the following format can be used:

```
*INTERFACE_BLANKSIZE_SCALE_FACTOR
$    IDCRV        SF
         1       0.2
         2       0.8
         3       1.0
         ⋮         ⋮
```

Example I: Simple Example of the Development Option

Given the initial and final blank configuration and a target, this option calculates a new initial blank outline, corresponding to the target final blank boundary. In this example note that IADAPT = 1, meaning initial and final blank meshes may differ by an adaptively operation. The input and output files are detailed below, and output results are shown in Figure 25-6.

```
*KEYWORD
*INTERFACE_BLANKSIZE_DEVELOPMENT
$  IOPTION              IADAPT
         2                   1
$  input file for target mesh, or target position coordinates
targetpoints.k
$  input file for formed mesh
final.k
$  input file for initial blank mesh
initial.k
*END
```

The file, targetpoints.k, is partially shown below was generated from IGES using *LS-PrePost 4.1*.

```
*KEYWORD
*DEFINE_TARGET_BOUNDARY
    -1.83355e+02    -5.94068e+02    -1.58639e+02
    -1.80736e+02    -5.94071e+02    -1.58196e+02
    -1.78126e+02    -5.94098e+02    -1.57813e+02
    -1.75546e+02    -5.94096e+02    -1.57433e+02
    -1.72888e+02    -5.94117e+02    -1.57026e+02
          ⋮                ⋮               ⋮
    -1.83355e+02    -5.94068e+02    -1.58639e+02
*END
```

The output is the modified initial blank outline in the file trimcurves.ibo.

Example II: The Reference Surface feature for the Development Option

For an initial blank that is not flat, the fields REFERENC and FILENAME4 can be used to define a surface onto which changes in the boundary are needed. This is important when the adjusted boundary is not a simple tangential extension of the initial blank.

In a keyword example below, REFERENC is set to "1" and the reference file for the extended initial shape is given as ref3.k. The maximum change between the initial and final blank size is set to be 20.0 mm per iteration. Point spacing distance (SPACE) of calculated trim curve on the reference surface is set at 2.0 mm. Note that the inner holes and outer boundary curves are defined in "target.xyz". The holes do not necessarily need to exist in the initial or final blank mesh. Also, since ORIENT is set to "2", a reference surface mesh file (ref4.k) is provided for the final (formed) state. The input details and output results are shown in Figure 25-1.

```
*KEYWORD
*INTERFACE_BLANKSIZE_DEVELOPMENT
$        1         2         3         4         5         6         7         8
$  IOPTION              IADAPT   MAXSIZE REFERENCE     SPACE              ORIENT
        -2                   1    20.000         1       2.0                   2
$  input file for target mesh:
target.xyz
$  input file for formed mesh:
final.k
$  input file for initial blank mesh:
initial.k
$  reference file for extended initial shape:
ref3.k
$  reference file for extended final shape:
Ref4.k
*END
```

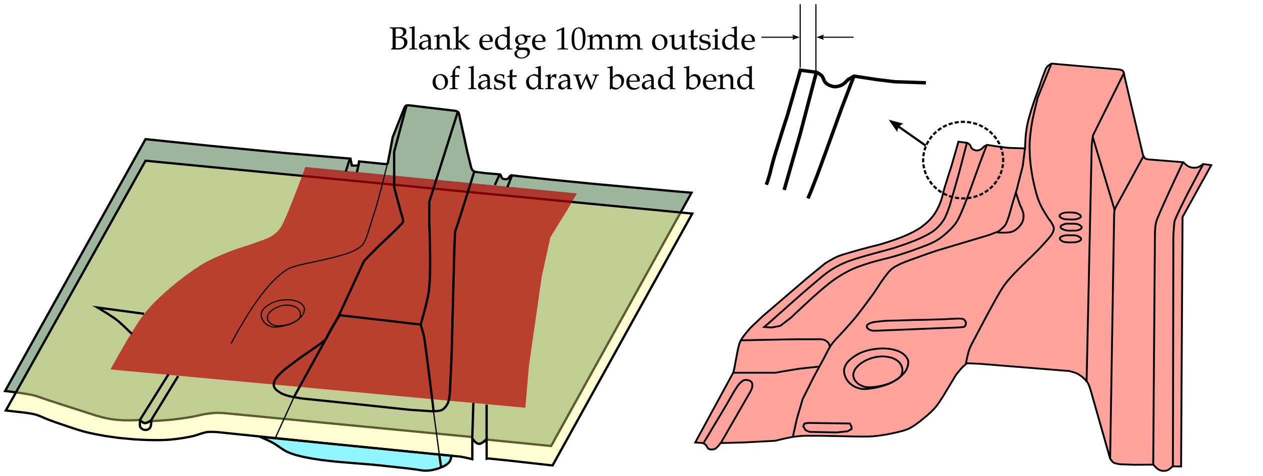


Figure 25-7. NUMISHEET 2005 cross member in Exmaple III.

Example III: Development Feature Applied to a Draw Die with Physical Bead

In this example, which was created from NUMISHEET 2005, the DEVELOPMENT option has been used to design a blank such that, when formed, the edge is a specified distance outside of the last bend of a draw bead. In Figure 25-7, the tooling and blank set up is shown to the left. The right side of the figure shows the target blank, whose left and right edges everywhere are made 10mm outside of the last bending radius of the draw beads. This setup is prototypical of one method to ensure a very stable and high-quality stamping process.

The first step towards setting up this analysis was to use *CONTROL_FORMING_ONESTEP to unfold the target blank and thereby obtain an initial guess of a flat blank, as shown to the left if Figure 25-12. The flat blank is then formed as one would usually do in a regular forming simulation, shown on the right side of the figure. The formed blank (Iteration 0) turns out to be larger than the target.

Next, the DEVELOPMENT is applied to generate a new and better initial blank that will lead to the target blank. The flat blank is used as input for the “initial blank mesh”, the formed blank is used as input for the “simulated mesh”, and the target blank mesh, or boundary points is used to define the target.

In Figure 25-13 (left) the improved initial blank, called *the first compensated blank,* is superimposed onto the original one-step unfolded result. The one-step unfolded result is somewhat larger than the developed blank. When formed the improved blank (Iteration 1) nearly overlaps the target blank, shown in Figure 25-13 (right). If the final formed blank still deviates from the target, another iteration would ensue, until satisfactory results are obtained.

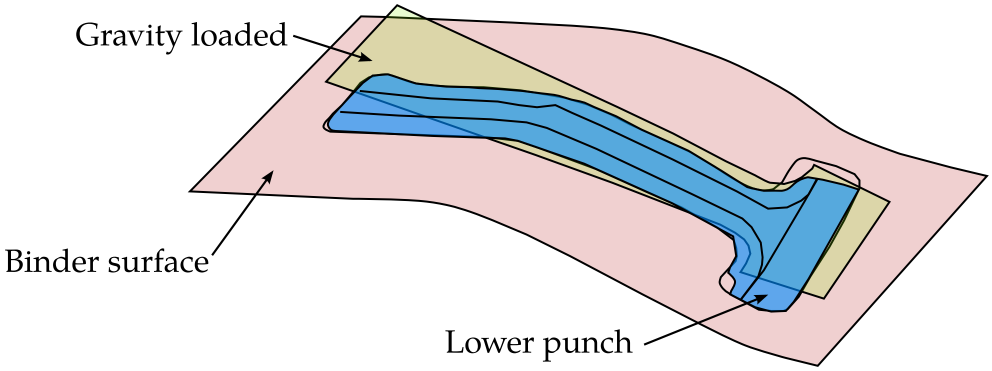


Figure 25-8. NUMISHEET 2008 B-pillar; Gravity loaded blank.

Example IV: Iterating with the Development Option

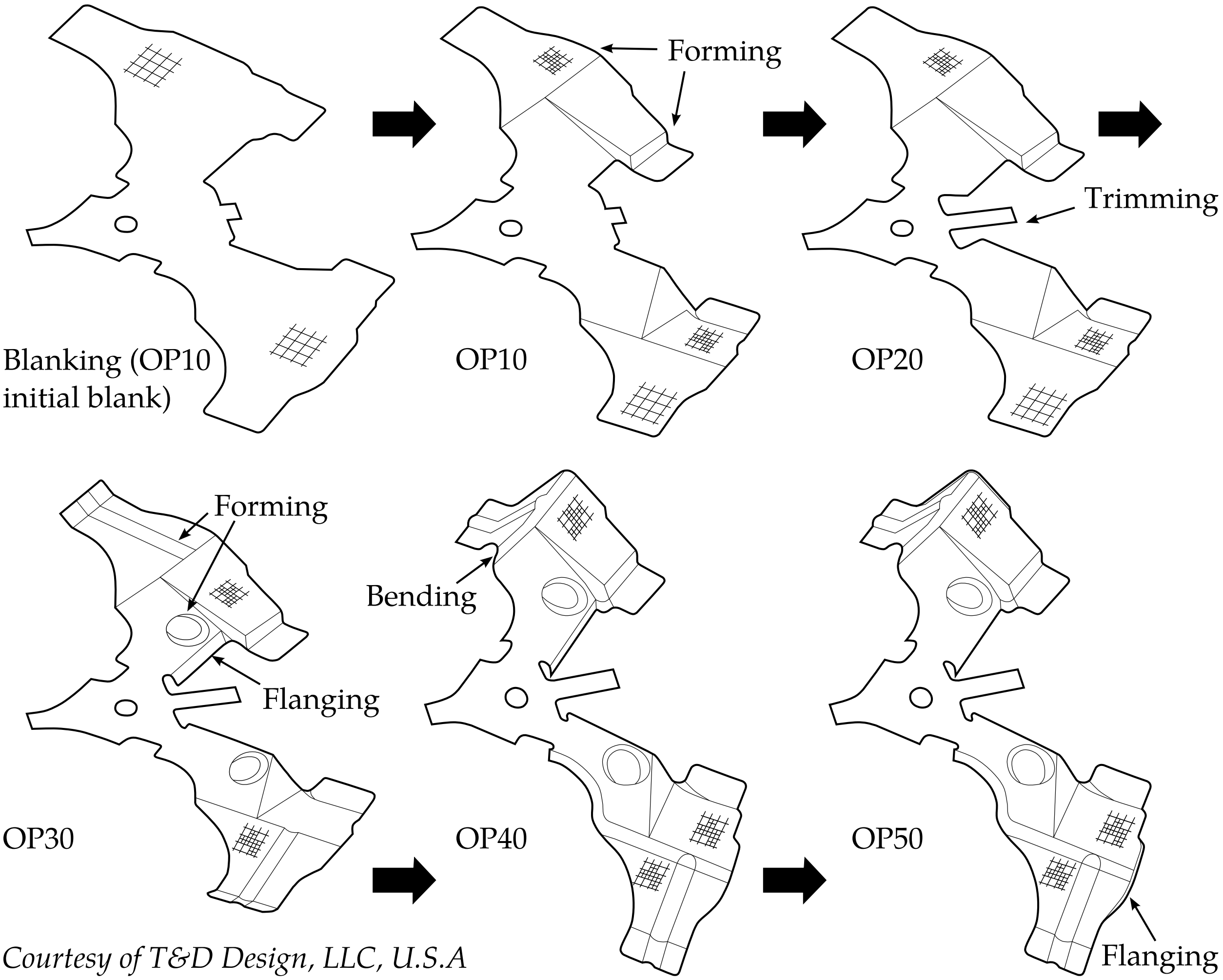
Because the NUMISHEET 2008 B-pillar model involves neither trimming nor flanging, it exmplifies the DEVELOPMENT option in its most direct use case. This model, illustrated in Figure 25-8, simulates a draw-die's action on a gravity-stressed flat blank. The B-pillar undergoes a stamping process including gravity, binder closing, and being drawn. In this example the DEVELOPMENT option is used to calculate the geometry for an initial blank that will exactly satisfy the design specification for a final formed panel. To highlight the efficiency of this feature, we start with an initial blank whose formed product deviates from the specification by a wide margin and then iterate using the DEVELOPMENT feature.

The target and the optimal initial blank are shown in Figure 25-14. The intial guess is intentionally deviated from the optimal initial blank as shown in the left panel of Figure 25-15; while the formed blank is compared with the target in the right panel.

In the first iteration a new initial blank is computed, and illustrated in the left panel of Figure 25-16 bearing the label, *1st compensated blank*. The simulation is repeated using the first compensated blank, and in the right panel the result is compared to the target. The formed blank is narrower in the notched areas as compared with the target.

The second iterate is shown in the left panel of Figure 25-17 bearing the label, *2nd compensated blank*. Again, the simulation is repeated, but this time using the second compenated blank, and the result is compared to the target in the right panel of Figure 25-17. The resulting product is a good match to the target.

Because of the initial blank was intentionally deviated from its ideal shape by a large margin, this example requires two iterations to converge. Generally this processed is bootstrapped with the *CONTROL_FORMING_ONESTEP card, which calculates an initial guess by approximately unfolding the target shape.



Courtesy of T&D Design, LLC, U.S.A

Figure 25-9. Enhanced DEVELOPMENT feature on a progressive die. Even though trimming occurs at OP20 the algorithm requires as input only OP10, OP50, and a target geometry.

Example V: Development Feature Applied to Flanging Process

In this example, which is schematically illustrated in Figure 25-18, the NUMISHEET 2002 fender outer is flanged along the hood line. The development feature adjusts the initial blank's boundary so that the formed piece matches the specified target flanged shape as shown in Figure 25-19. For demonstration purposes, the trimmed blank shape is intentionally deviated from the optimal configuration by a large amount. This error is indicated in Figure 25-20 by the label *initial guess trim curves*. In Figure 25-20 The flanged product is shown to deviate substantial from the flanged target along the boundary. As shown in Figure 25-21 after one iteration the correct initial blank boundary is obtained.

Alternatively, *CONTROL_FORMING_UNFLANGING can be used to unfold the flanged target onto the addendum to obtain the initial blank size, or a starting guess for this process.

Example VI: Enhanced-Development Feature Applied to Progressive Die Process

In the example of Figure 25-9, *courtesy of T&D Design, LLC, U.S.A*, the blank-shape for a five stage progressive die process is calculated. Because this process involves a trimming step, the development capability prior to Revission 88708 does not support this example.

In this example, an initial blank at OP10, undergoes trimming, reforming, bending and flanging to arrive at the blank in OP50. In figure 25-23 the computed product is compared with the specified target. A blank size development calculation produces the modified OP10 initial blank outline. The updated blank is used in a verification simulation. As seen in Figure 25-24 the blank size development feature produces a good result.

Trim lines are not optimized by the development feature, so trimming should only occur along the boundary of the target blank. The modified OP10 blank requires some refitting in the trimmed area.

Revision information:

This feature is available in both SMP and MPP, in double precision only. Note a GUI for using this feature is now available in *LS-PrePost* as of version 4.2. It can be accessed under APPLICATION → Metal Forming → Blank Size/Trim line Dev. Starting revision information for each feature is listed as follows:

1. DEVELOPMENT option: Revision 74605.

2. INITIAL_TRIM and INITIAL_ADAPTIVE: Revision 75023.

3. IADAPT: Revision 75827.

4. Command line option “JOBID=…”: Revision 82861.

5. MAXSIZE: Revision 85633.

6. SPACE: Revision 85755.

7. MAXGAP: Revision 85792.

8. Reference surface (parameter REFERENC): Revision 86086.

9. Removal of the restriction requiring the initial (FILENAME3) and the final (FILENAME2) blanks must be from the same process simulation: Revision 88708.

10. Improve smoothness of the output trim curve trimcurves.ibo in case the given meshes have warpage caused by wrinkles during forming: Revision 96164.

11. Symmetric plane: Revision 97443.

12. Scale factor: Revision 98122.

13. ORIENT = 1: Revision 100453.

14. ORIENT = 2: Revision: 104168.

15. Curve direction is desensitized, meaning both IOPTION = 2 and IOPTION = -2 will give the same results: Revision 100589.

16. OFFX, OFFY, OFFZ: Revision 100660.

17. Automatically calculate offset values (OFFX.EQ.-10000.0): Revision 102609.

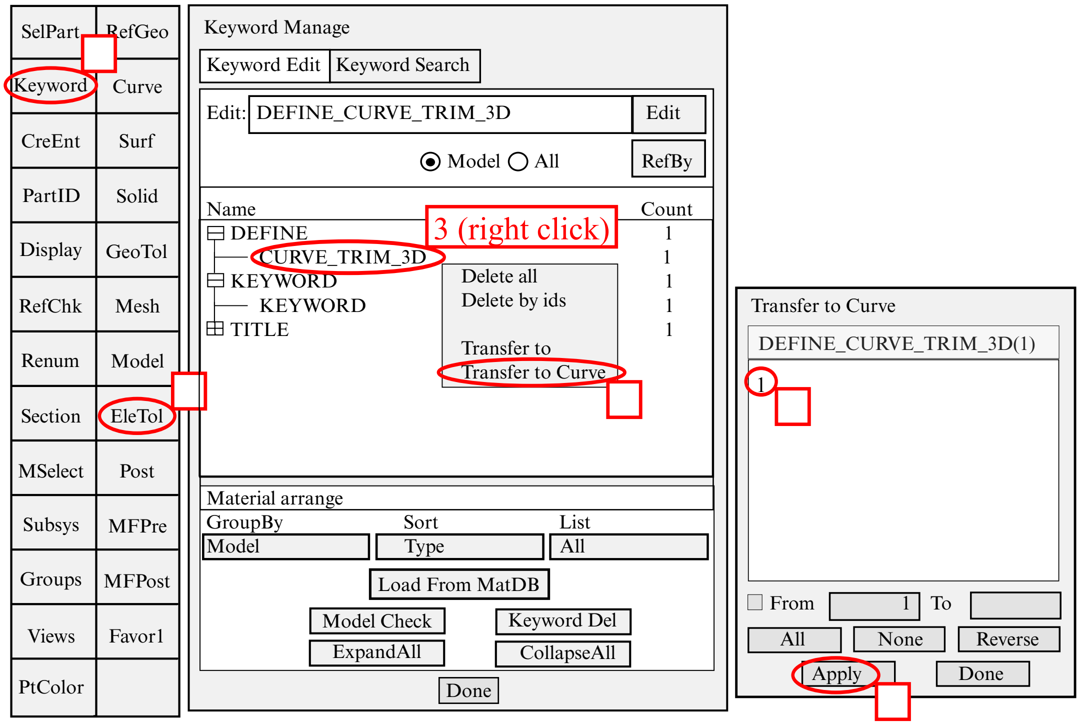


Figure 25-10. Converting trimcurves.ibo to IGES format in LSPP4.0.

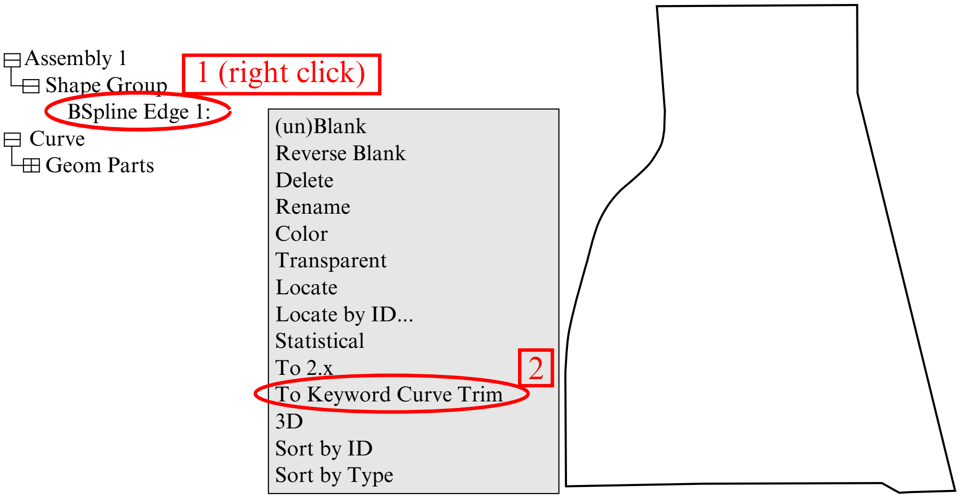


Figure 25-11. Converting IGES file to *DEFINE_CURVE_TRIM_3D.

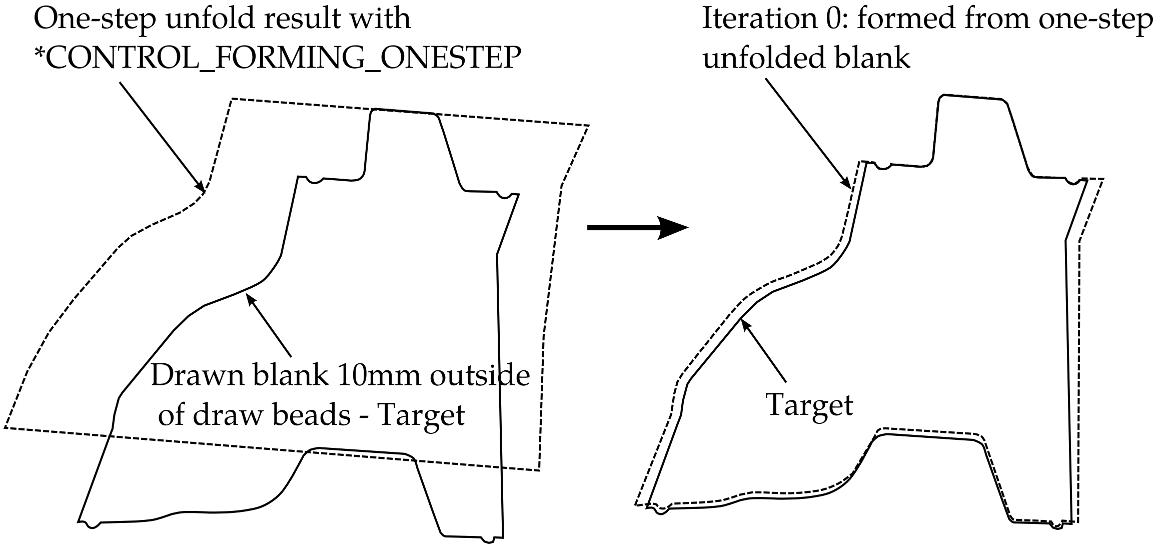


Figure 25-12. Initial blank calculation and baseline formed blank.

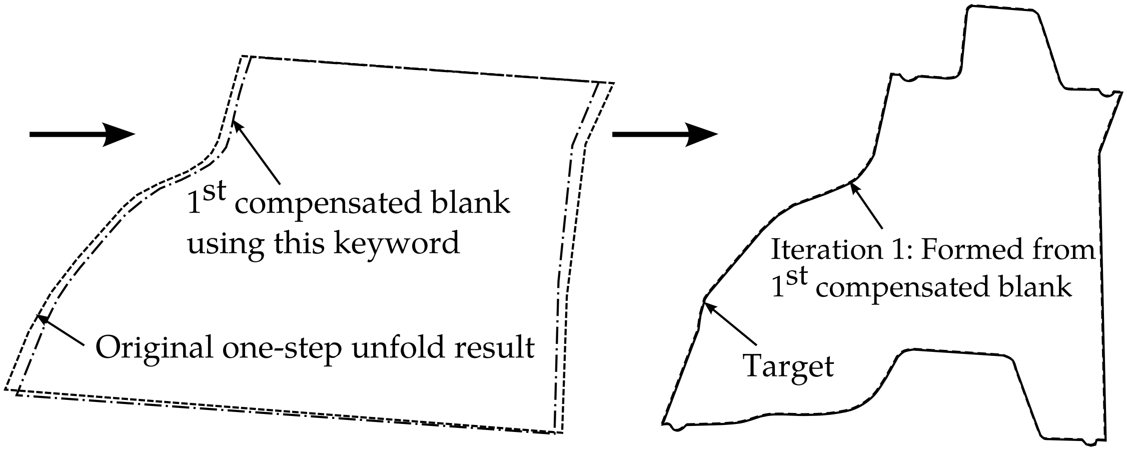


Figure 25-13. The first compensated blank and the final confirmation run.

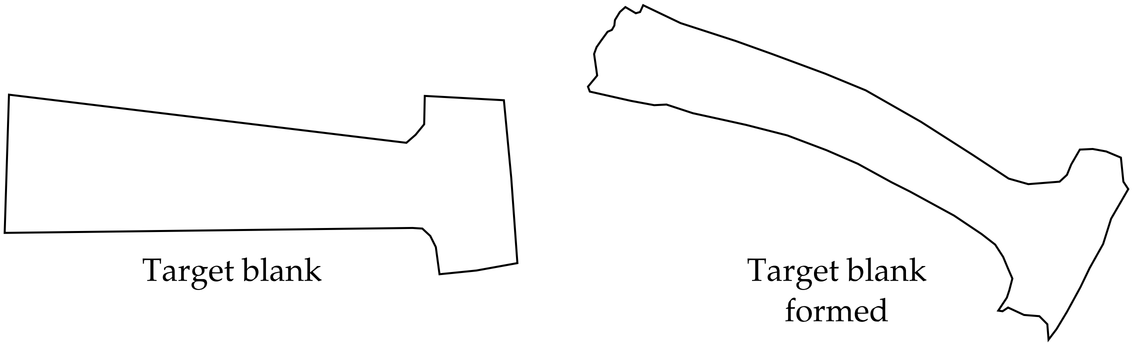


Figure 25-14. Assumed target blanks.

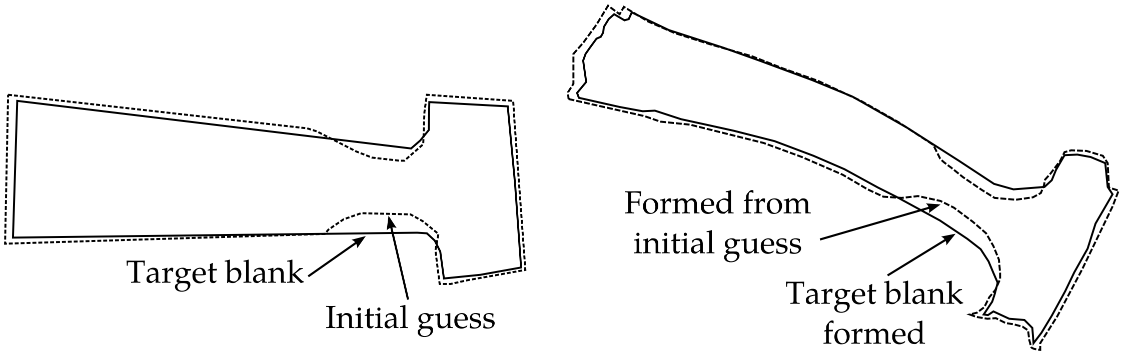


Figure 25-15. Iteration 0 results comparison with target.

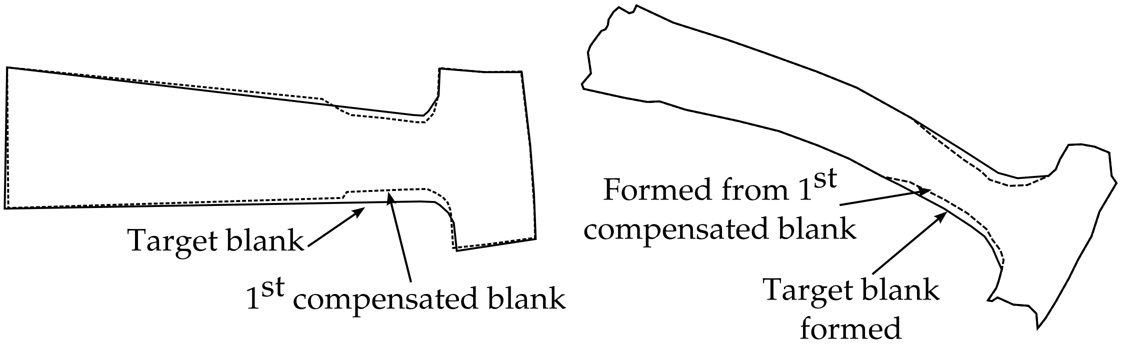


Figure 25-16. Iteration 1 results.

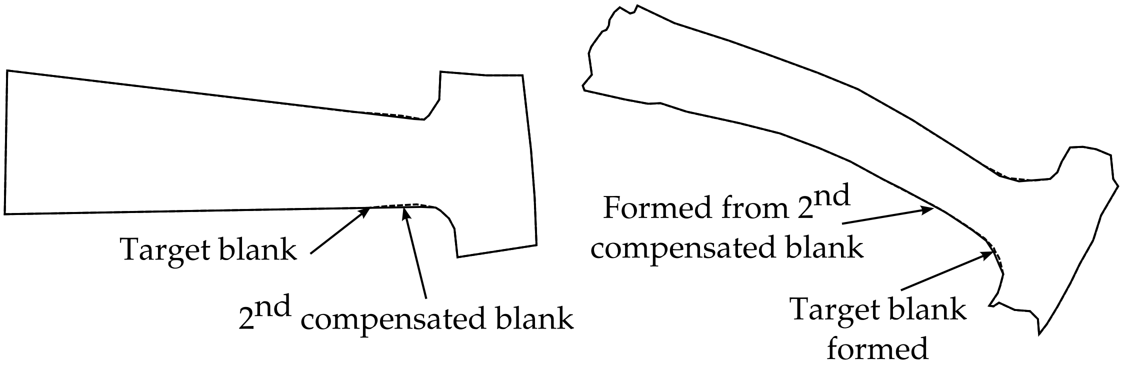


Figure 25-17. Iteration 2 results.

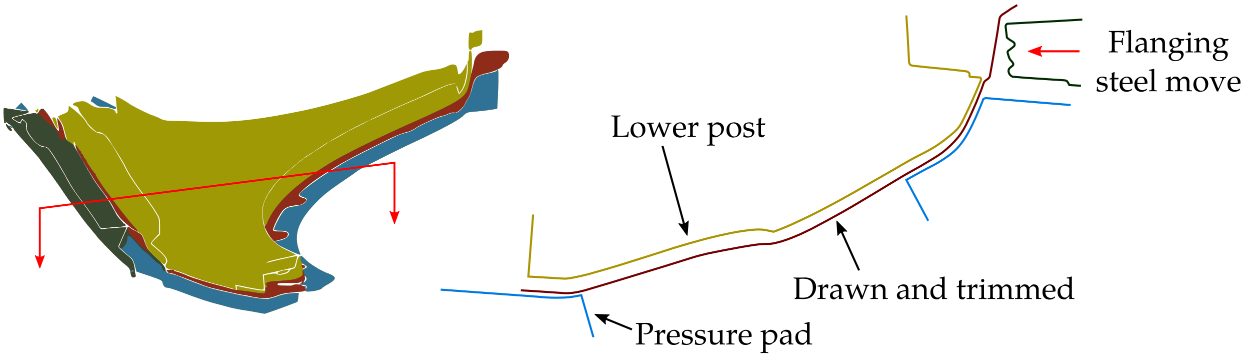


Figure 25-18. The flanging process on NUMISHEET 2002 fender outer.

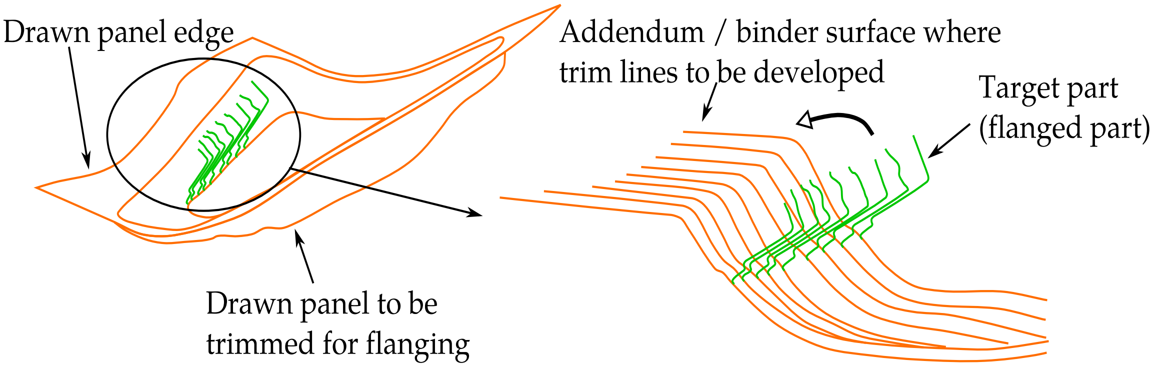


Figure 25-19. Multiple section view showing the target part and addendum surfaces.

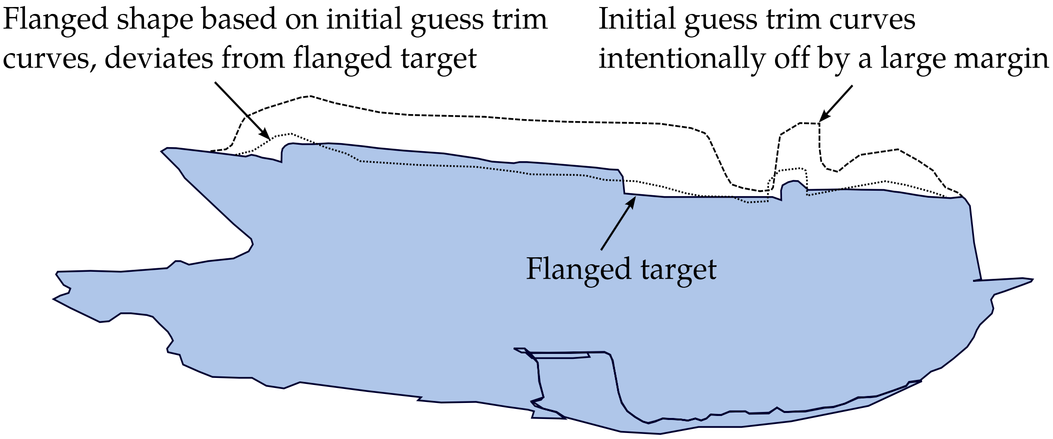

Flanged shape based on initial guess trim
curves, deviates from flanged target
Initial guess trim curves
intentionally off by a large margin
Flanged target

Figure 25-20. Initial trim curves intentionally made to be off by a large margin.

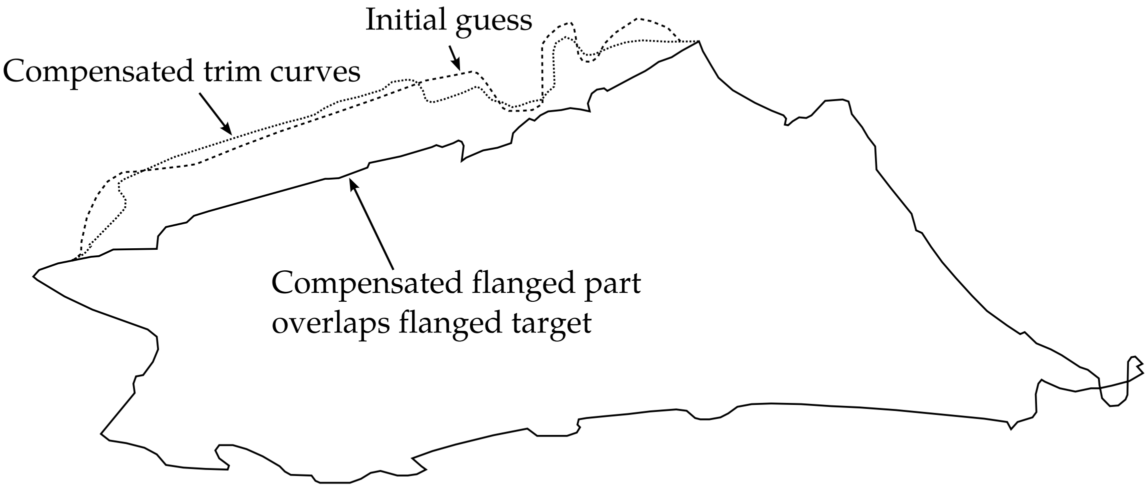
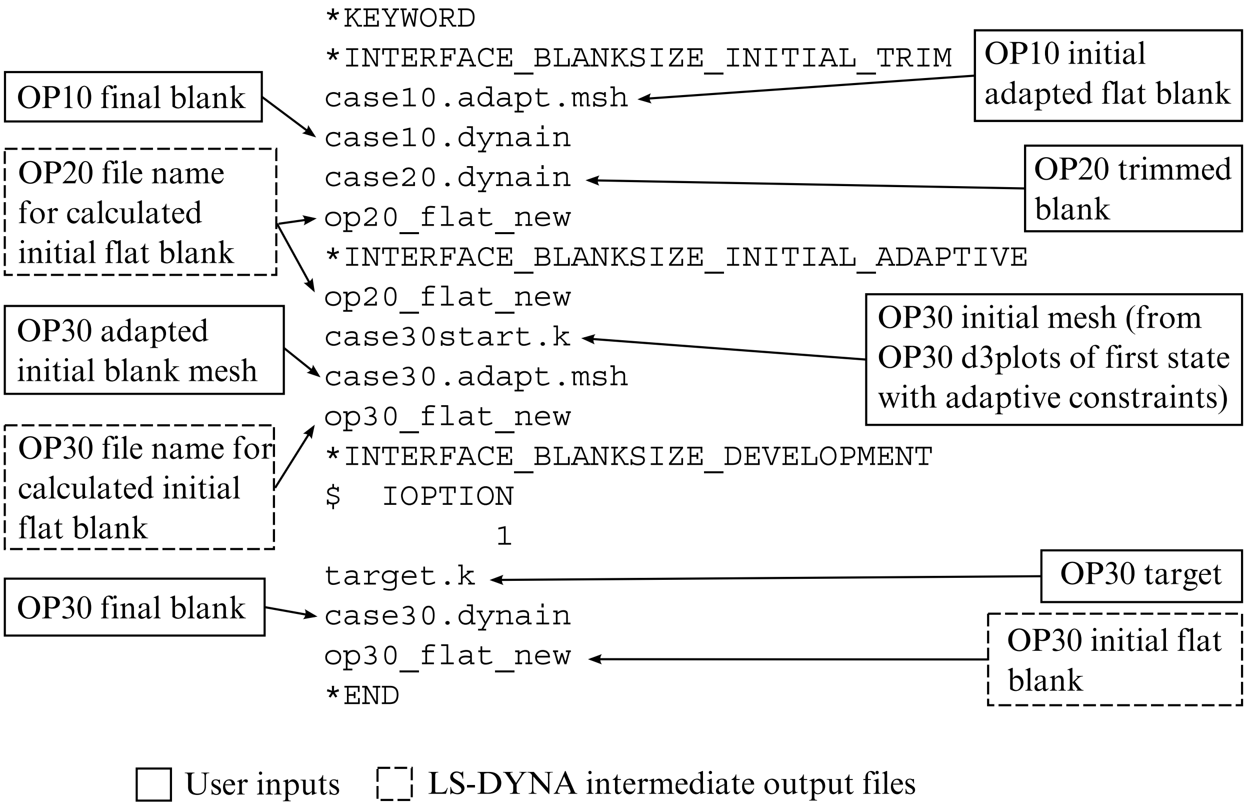


Figure 25-21. Compensated trim curves overlap with the target curves.

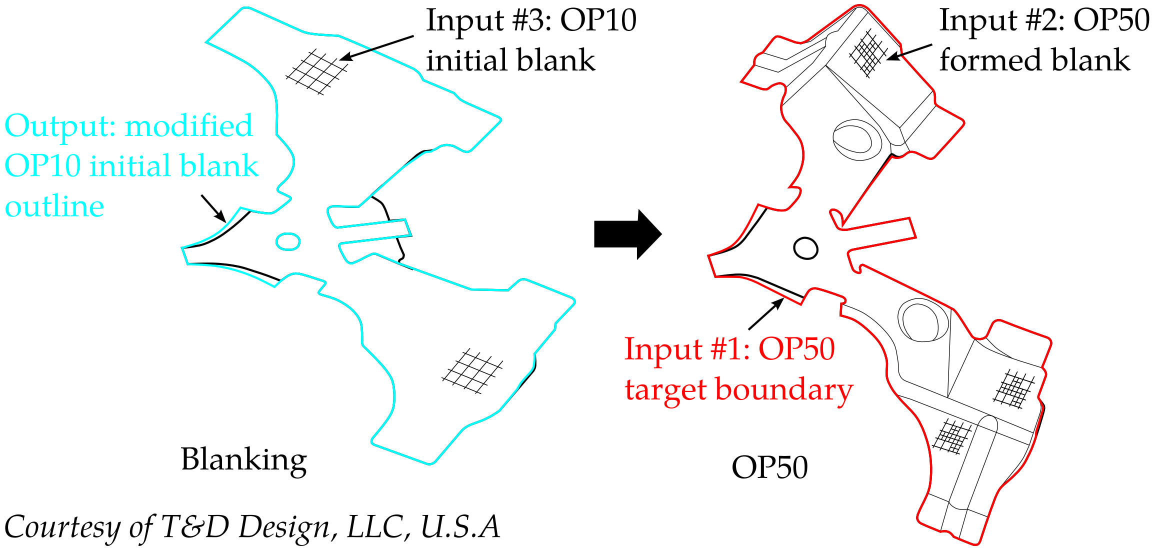


```
*KEYWORD
*INTERFACE_BLANKSIZE_INITIAL_TRIM
case10.adapt.msh
case10.dynain
case20.dynain
op20_flat_new
*INTERFACE_BLANKSIZE_INITIAL_ADAPTIVE
op20_flat_new
case30start.k
case30.adapt.msh
op30_flat_new
*INTERFACE_BLANKSIZE_DEVELOPMENT
$   IOPTION
          1
target.k
case30.dynain
op30_flat_new
*END
```

OP10 final blank
OP20 file name for calculated initial flat blank
OP30 adapted initial blank mesh
OP30 file name for calculated initial flat blank
OP30 final blank
OP10 initial adapted flat blank
OP20 trimmed blank
OP30 initial mesh (from OP30 d3plots of first state with adaptive constraints)
OP30 target
OP30 initial flat blank
User inputs
LS-DYNA intermediate output files

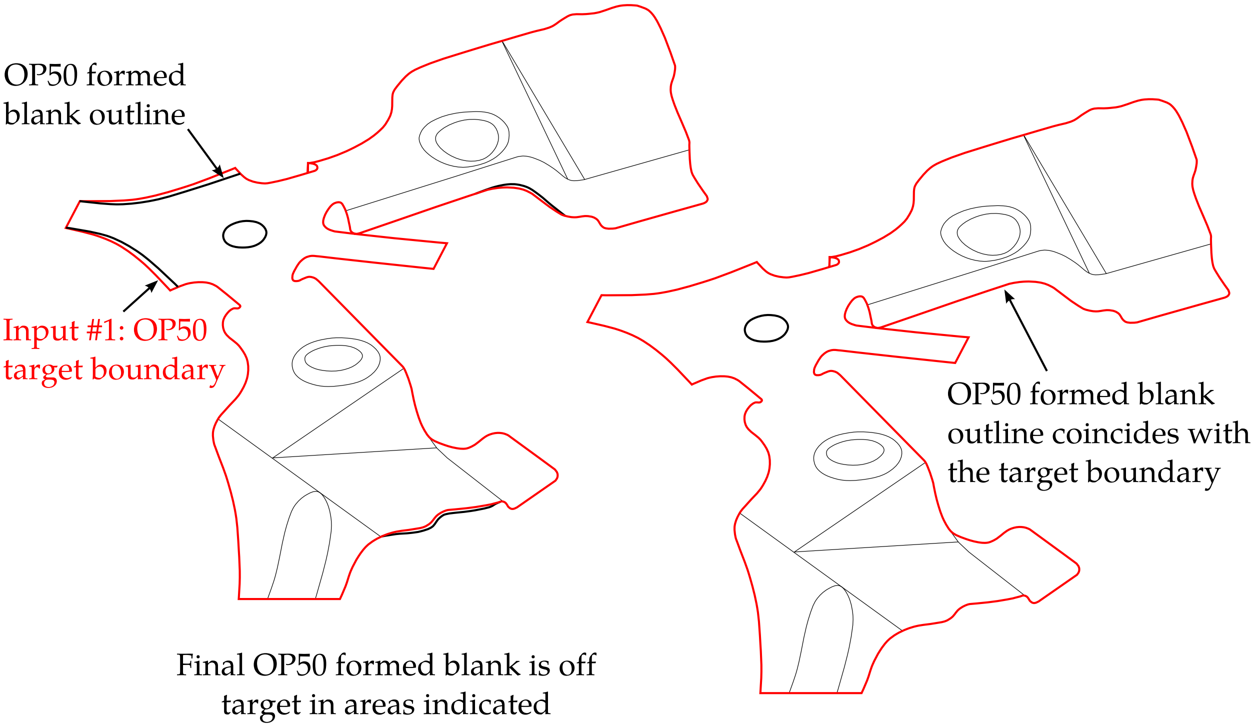
LS-DYNA simulation output: new trim line OP10 (file "trimcurves.ibo")

Figure 25-22. File structures for a multi-process blank development.



Courtesy of T&D Design, LLC, U.S.A

Figure 25-23. Inputs and output for the enhanced DEVELOPMENT feature.



Final OP50 formed blank meets target requirement with OP10 modified blank outline

Courtesy of T&D Design, LLC, U.S.A

Figure 25-24. Verification simulation on a progressive die process.

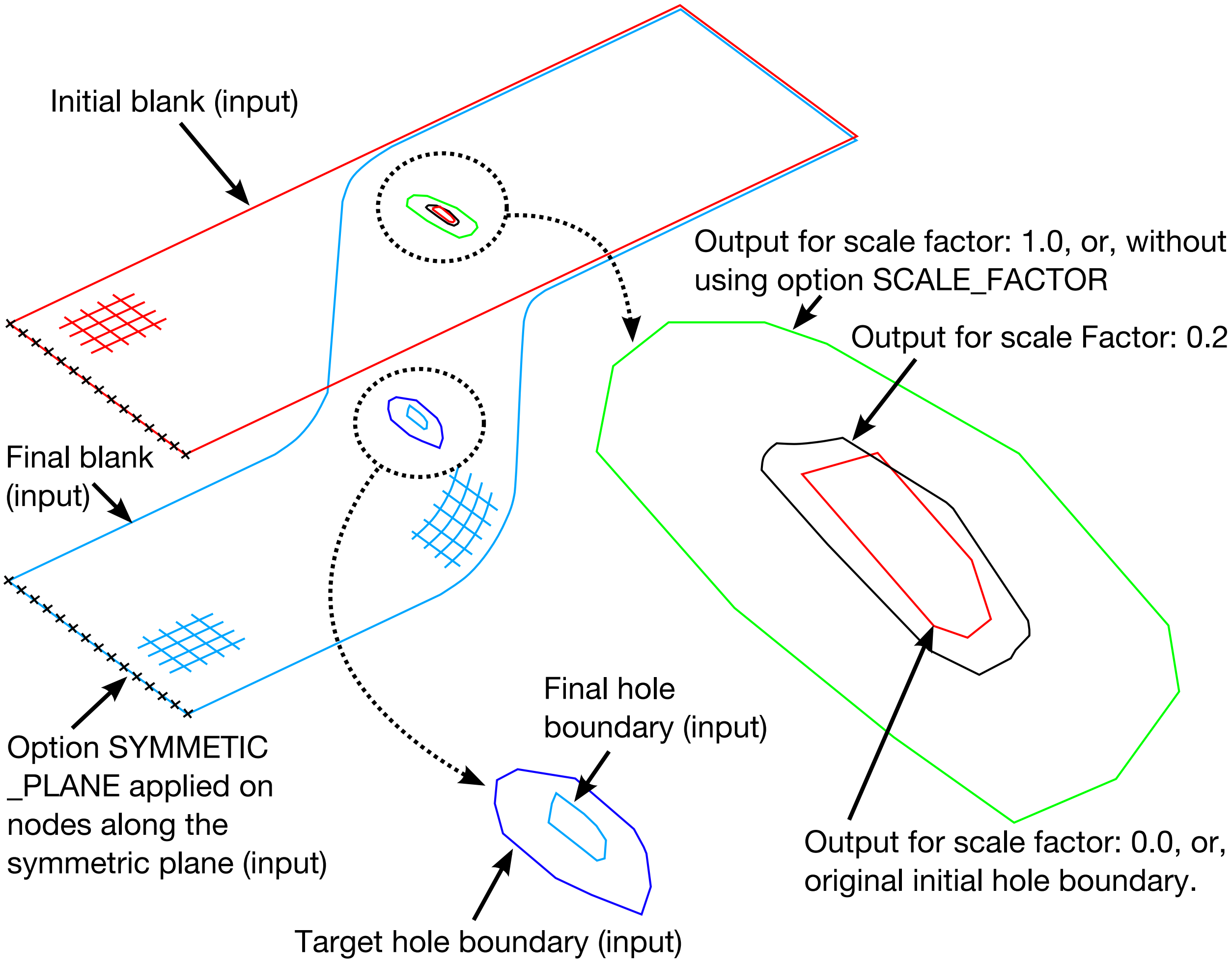

Initial blank (input)
Final blank (input)
Option SYMMETIC _PLANE applied on nodes along the symmetric plane (input)
Output for scale factor: 1.0, or, without using option SCALE_FACTOR
Output for scale Factor: 0.2
Final hole boundary (input)
Output for scale factor: 0.0, or, original initial hole boundary.
Target hole boundary (input)

Figure 25-25. Options SCALE_FACTOR and SYMMETRIC_PLANE.

INTERFACE_COMPENSATION_NEW_*{OPTION}

Available options include:

<BLANK>

ACCELERATOR

MULTI_STEPS

LOCAL_SMOOTH

PART_CHANGE

REFINE_RIGID

Purpose: This card encompasses several methods for spring back compensation in stamping tools. The applications for this card include: (1) calculating the deviation of the stamped part from its intended design, and to automatically compensate the tool to minimize the deviation; (2) to map the existing trim curve to the modified tool; and (3) to automatically detect undercut.

This keyword employs a nonlinear iterative method. Usually, it takes between 2 and 4 iterations to converge within tolerances. Additionally, this method provides a scale factor, which allows the user to decide the ratio of shape deviation the part is compensated.

Options

The **ACCELERATOR** option speeds up the convergence rate in reducing the part deviation to design tolerance thus reducing the number of iterations. This option also allows for a much simpler user interface.

The **MULTI_STEPS** option allows for tooling compensation of the next die process, based on target blank shape, compensated blank shape for the next step, and current tools. This feature is useful in line die process/tooling compensation.

The **LOCAL_SMOOTH** option features smoothing of a tool's local area mesh, which could become distorted because of either bad or coarse mesh of the original tool surface, or in areas where tooling pairs (for example, flanging post and flanging steel) do not maintain a constant gap, or after a few compensation iterations.

The **PART_CHANGE** option allows for updating of the final compensated tool using the changed part or formed blank shape, thus eliminating the need for going through a new compensation iteration loop. This option is used together with *INCLUDE_COMPENSATION_UPDATED_BLANK_SHAPE, and *INCLUDE_COMPENSATION_UPDATED_RIGID_TOOL.

The **REFINE_RIGID** option refines the rigid tool mesh based on user-provided trim curves. It also realigns the mesh so no elements cross the trim curves. This feature only needs to be done once before the iterative springback compensation begins. The modified rigid too mesh will greatly improve the convergence in the iterative process.

Known Limitation

The current methods sometimes fail to eliminate undercut.

All required input files must be included by using various options in the keyword: *INCLUDE_COMPENSATION_{*OPTION*}. The option LOCAL_SMOOTH also needs to use a keyword *SET_NODE_LIST_SMOOTH.

Card 1 for <BLANK>, MULTI-STEPS, and LOCAL_SMOOTH options.

Card 1	1	2	3	4	5	6	7	8
Variable	METHOD	SL	SF	ELREF	PSIDM	UNDCT	ANGLE	NLINEAR
Type	I	F	F	I	F	F	F	I
Default	6	5.0	0.75	1	none	none	0.0	1

Card 1 for keyword ACCELERATOR option.

Card 1	1	2	3	4	5	6	7	8
Variable	ISTEPS	TOLX	TOLY	TOLZ	OPTION			
Type	I	F	F	F	I			
Default	0	0.5	0.5	0.5	1			

Card 1 for keyword option PART_CHANGE:

Card 1	1	2	3	4	5	6	7	8
Variable	MAXGAP							
Type	F							
Default	none							

Card 1 for keyword option REFINE_RIGID:

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME1							
Type	A80							
Default	none							

Card 2 for keyword option REFINE_RIGID:

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME2							
Type	A80							
Default	none							

VARIABLE	DESCRIPTION
METHOD	There are several extrapolation methods for the addendum and binder outside of trim lines, see **Remarks**.
SL	The smooth level parameter controls the smoothness of the modified surfaces. A large value makes the surface smoother. Typically the value ranges from 5 to 10. If spring back is large, the transition region is expected to be large. However, by using a smaller value of SL, the region of transition can be reduced.

VARIABLE **DESCRIPTION**

SF

Shape compensation scale factor. The value scales the spring back amount of the blank and the scaled amount is used to compensate the tooling.

GT.0: compensate in the opposite direction of the spring back;

LT.0: compensate in the punch moving direction (for undercut).

This scale factor scales how much of the shape deviation is compensated. For example, if 10 mm of spring back is predicted, and the scale factor is chosen as 0.75, then the compensation in the opposite direction will only be 7.5 mm.

Experience shows that the best scale factor for reaching a converged solution (within part tolerance) is case dependent. In some cases, a scale factor range of 0.5 to 0.75 is best; while in others, larger values are indicated. Sometimes, the best value can be larger than 1.1. Note that within an automatic compensation loop, this factor does not need to be varied.

Since it is impossible to choose the best value for each application up front 0.75 is recommended for the first attempt. If the spring back cannot be effectively compensated and the calculation diverges, the factor can be moved upward or downward to obtain a converged solution, or more iterations must be used with the initial trial value to compensate the remaining shape deviation.

For channel shaped parts that have a twisting mode of spring back, the scale factor is more important. It was found that a small change of the tool shape might change the twisting mode. If this occurs, using a small value (<0.5) is suggested.

ELREF

Element refinement option:

EQ.1: special element refinement is used with the tool elements (default);

EQ.2: special element refinement is turned off.

VARIABLE DESCRIPTION

PSIDM

Define the part set ID for master parts. It is important to properly choose the parts for the master side. Usually, only one side (master side) of the tool will be chosen as the master side, and the modifications made to the other side (slave side) depends solely on the changes in the master. This allows the two sides to be coupled and a constant (tool) gap between the two sides is maintained. If both sides are chosen to be master, the gap between the two sides might change and become inhomogeneous.

The choice of master side will have an effect on the final result for method 7 when applied to three-piece draw models. At this time, when the punch and binder are chosen as the master side, the binder region will not be changed. Otherwise, when the die is chosen as master side the binder will be changed, since the changes extend to the edges of the master tool.

UNDCT

Tool undercut treatment option:

EQ.0: no check (default);

EQ.1: check and fix undercut.

ANGLE

An angle defining the undercut.

NLINEAR

Activate nonlinear extrapolation.

ISTEPS

Steps in accelerated compensation procedure, see **Remarks**.

TOLX

Part deviation tolerance between current blank and target blank shape in global *x*-direction.

TOLY

Part deviation tolerance between current blank and target blank shape in global *y*-direction.

TOLZ

Part deviation tolerance between current blank and target blank shape in global *z*-direction.

OPTION

Compensation acceleration method. Currently available only for method 1.

MAXGAP

Maximum gap between the original part and changed part.

VARIABLE	DESCRIPTION
FILENAME1	Rigid tool mesh file in keyword format. This should be the tooling mesh used in the forming or flanging simulation, before any compensation is done. The refined rigid tool mesh will be in the file rigid_refined.tmp. See Option REFINE_RIGID: rigid tool mesh refinement for a better convergence.
FILENAME2	Trim curves in keyword format *DEFINE_CURVE_TRIM_3D. The curves will be used to refine and realign the FILENAME1 to improve the convergence in the iterative compensation process. The refined rigid tool mesh will be in the file rigid_refined.tmp. See Option REFINE_RIGID: rigid tool mesh refinement for a better convergence.

Compensation Methods Overview:

After trimming, only a limited part of the tool has direct relationship with the spring back of the blank part. Modifications of the rigid tool outside the trimmed region involves extrapolation. Unfortunately, extrapolating is unstable and tends to generate non-smooth surfaces. To resolve this problem, seven smoothing algorithms are implemented. The most frequently used methods are methods 7, 8 and -8, while the others are used only occasionally.

Method 7

If the punch is chosen as the master side, the binder will not be changed. Aside from the region inside the punch opening the rest of the model is untouched. Smoothing has little effect on method 7. The smoothness of the modified tool depends on the magnitude of the spring back and the size of the addendum region. This method is nonlinearly and, therefore, necessitates an iterative solve.

Advantages: The binder will not be changed.

Disadvantages: The change will be limited inside the addendum region, and the modified surface may not be smooth if the spring back magnitude is large and the transition is small.

Method 6

The smoothness and the transition region of the modified surface will depend on the spring back magnitude and the smoothing factor. If the spring back magnitude is large, the transition region will be increased automatically. On the other hand, the transition region will be smaller if the spring back magnitude is small. At the same time, a larger smoothing factor will result in a smaller transition region. Like method 7, this too is nonlinear.

Advantages: The smoothness of the modified surfaces can be controlled.

Disadvantages: It is impossible to limit the transition region, and the binder surface (and therefore, draw beads) could change if the spring back is large.

Method 3

Similar to Method 6, however, it is a linear method and no iteration is necessary.

Method 8

This is an enhanced version of Method 6, and can account for addendum and binder changes. Usually the upper tooling including addendum and binder (in an air draw) are included in the PSIDM definition.

Method -8

This method is a modification of Method 8, and is used for trim die nesting (from the drawn panel shape).

Methods 1, 2, 4, and 5

These methods are deprecated and may be removed in the future. They are included only for maintaining backwards compatibility.

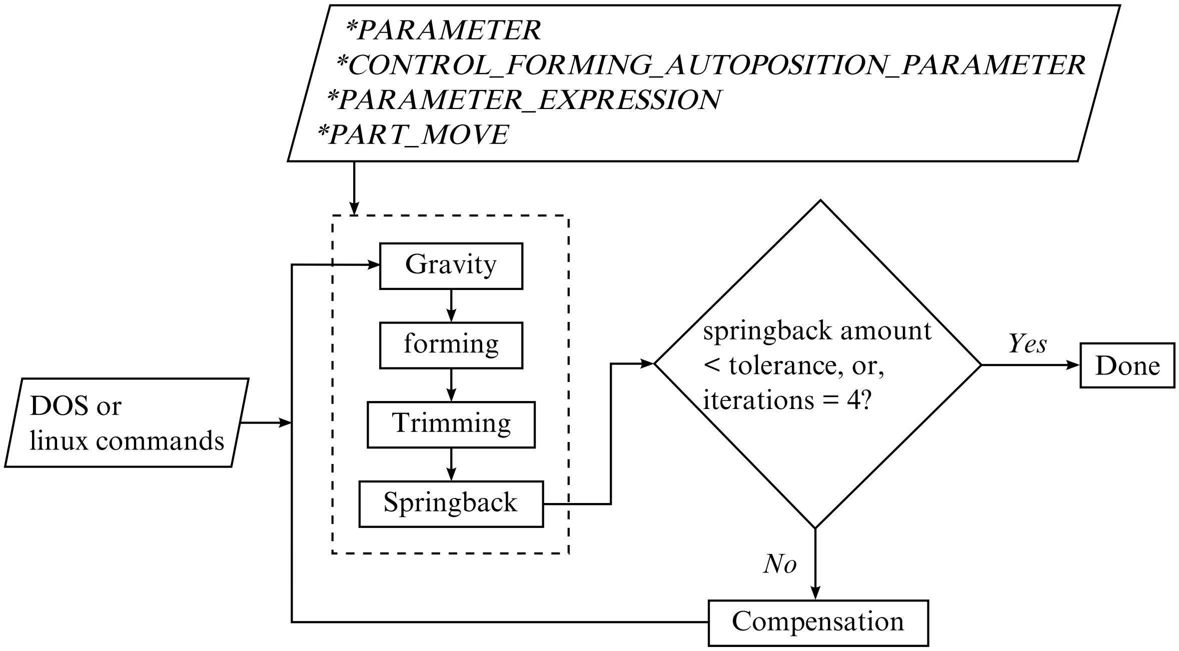


Figure 25-26. Iterative compensation flow chart

Preventing Undercut

When the draw wall is steep, it is likely that undercut will occur. Since undercut is not acceptable in real world die manufacturing, it must be prevented.

The compensation code can automatically detect undercut and issue a warning message. Additionally, LS-DYNA will write a list of undercut elements to a file called blankundercut.tmp so that the user can easily identify which elements may be problematic.

If the undercut is limited to only a few elements, it is possible to fix the problem manually.

Undercut can be reduced by compensating the spring back only in the punch moving direction (by using a negative scale factor). This method is not 100% reliable and more robust solutions are being studied.

Iterative spring back compensation

Figure 25-26 is a flow chart showing the iterative spring back compensation algorithm as applied to a typical stamping process. The first stamping process simulation is done following gravity→forming→trimming→spring back (ITERATION 0). The stamping process simulation is set up using eZ-Setup (http://ftp.lstc.com/anonymous/outgoing/lsprepost/4.0/metalforming/). With the use of the parameterized automatic tool/blank positioning feature, the process simulation is fully automated (no user intervention required). Based on the calculated spring back amount, tooling geometry is compensated through a compensation run. The stamping process simulation is conducted again, automatically, based on the new compensated tooling, followed by a second tooling compensation (ITERATION 1). Iterations 2, 3, and 4 follow the same pattern. The iteration process is repeated until blank spring back shape conforms to tooling designed intent (target), or until it reaches 4 iterations (typically required to achieve part tolerance). With some shell scripting this iterative loop can be completed automatically. These tools allow the user to toggle between single and double precision version of LS-DYNA.

The task of tracking the files involved in the iterative process can be daunting, especially in the advanced stage of the iterations. Figure 25-27 indicate what is written to storage during the process.

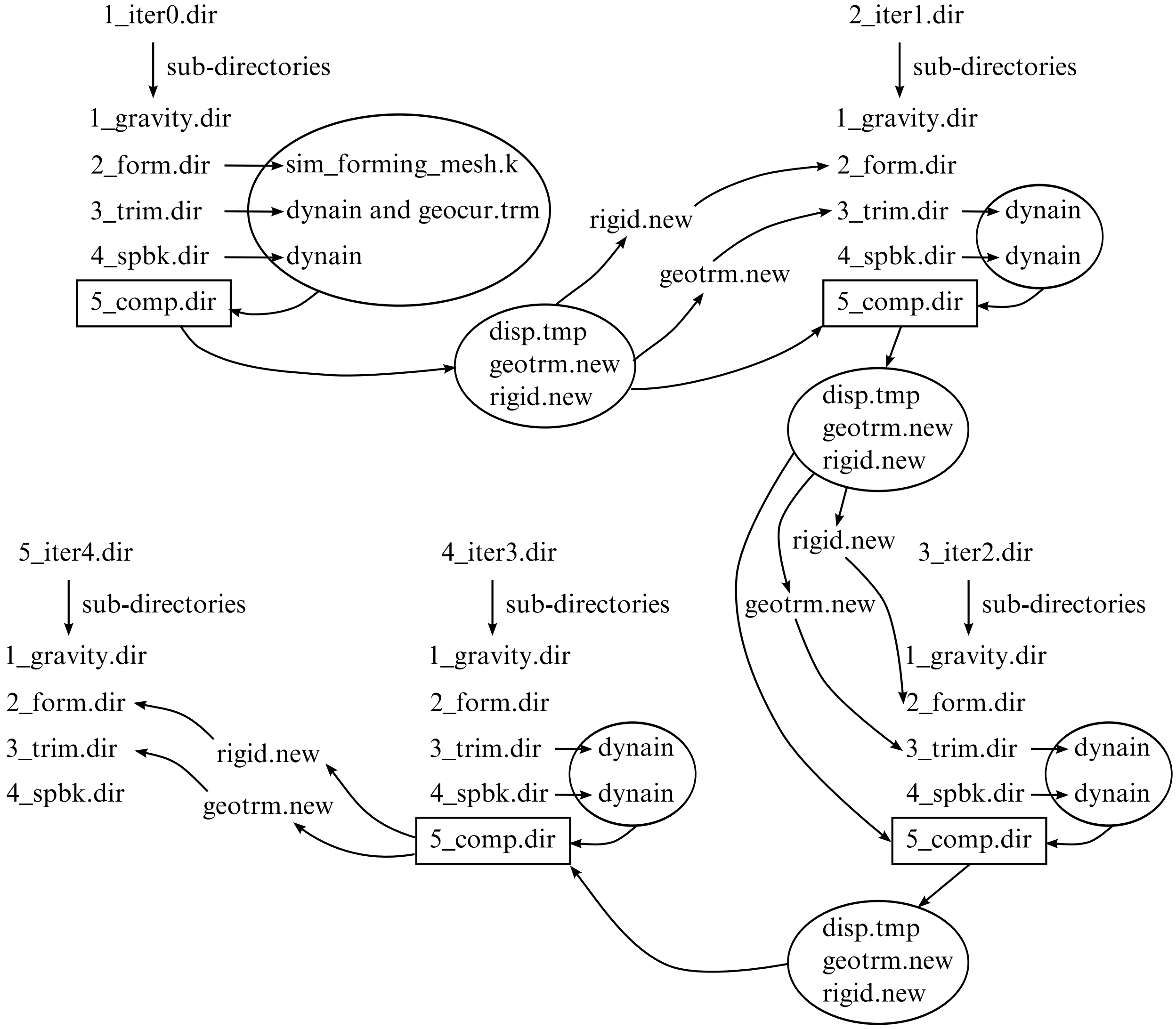


Figure 25-27. File structure for compensation

An input deck defining a spring back compensation model is given below. The keyword file blank0.k includes node and element information of the blank shape before spring back (after forming and trimming) with adaptive constraints (if exist). The keyword file spbk.k includes node and element information of the blank after spring back, with adaptive constraints (if they exist). The blank shapes before and after springback (blank0.k and spbk.k) may be based on either the original die design (ITER0), or based on an intermediate compensated die design (say the nth iteration).

The keyword file reference0.k is the blank shape before spring back for iteration 0 (ITER0). This file is blank0.k and should *not* change from iteration to iteration. For iteration 0 the file reference1.k is also the same as blank0.k, but for iteration 1 refereince1.k should be the disp.tmp generated from the compensation calculation during iteration 0 and so on and so forth for the subsequent iterations.

The keyword input tools.k must contain the mesh information for all of the stamping tools in their home positions. Compensated tools will be written to rigid.new with the original constant gap being maintained among the tools. During the baseline calculation, iteration 0, a keyword file called geocur.trm, generated during a LS-DYNA trimming simulation based on trimming curve input (usually in IGES format), is used for keyword *INCLUDE_COMPENSATION_TRIM_CURVE. In the compensation run of the ITER1, geocur.trm is used to generate new trim curves called geotrm.new, which conforms to the current compensated tools; and this new mapped trim curves are used for the ensuing ITER2, so on and so forth. The new trim file, geotrm.new is also is also in keyword format and contains a *DEFINE_CURVE_TRIM_3D card.

In the example below models a three-piece air draw process. The upper die cavity (including binder) has a part ID 2, which is included in the part set ID 1 and is used for variable PSIDM. Method 8 will compensate all the tools included in file tools.k based on compensated shape for the upper cavity.

```
*KEYWORD
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+--
*INTERFACE_COMPENSATION_NEW
$   METHOD        SL        SF     ELREF      PSIDm    UNDRCT     ANGLE NLINEAR
         8    10.000     1.000         0          1         0       0.0       1
*INCLUDE_COMPENSATION_BLANK_BEFORE_SPRING BACK
blank0.k
*INCLUDE_COMPENSATION_BLANK_AFTER_SPRINBACK
spbk.k
*INCLUDE_COMPENSATION_DESIRED_BLANK_SHAPE
reference0.k
*INCLUDE_COMPENSATION_COMPENSATED_SHAPE
reference1.k
*INCLUDE_COMPENSATION_CURRENT_TOOLS
tools.k
*INCLUDE_COMPENSATION_TRIM_CURVE
geocur.trm
*SET_PART_LIST
$      PSID
          1
$      PID
          2
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+--
*END
```

NUMISHEET 2005:

In Figure 25-28, the NUMISHEET 2005 cross member is compensated following the flow chart. In two iterations the spring back is reduced from 13mm to 1.7mm. Further iterations will reduce the part deviation down to a specific design target. Typically, four iterations are needed.

Iterative compensation applied during die construction

The blank shape after spring back can be obtained from the actual experimental shape of the spring back panel, if available. For example, in hard tool construction, the trimmed panel can be scanned using white light technology and the panel shape can be written to an STL file. The STL format can be easily converted to LS-DYNA keyword format and the trimmed panel can be used as a rigid tool onto which the baseline (ITER0) trimmed panel (deformable) can be "pushed" using element normal pressure, and using *CONTROL_IMPLICIT_FORMING type 1. In this scenario, the adaptive refinement is turned off to maintain the one-to-one correspondence of the elements and nodes information. An advantage of this method is that the spring back shape used for compensation will be exactly the same as the actual panel spring back, therefore the best tooling compensation result is expected. An example of such is shown in Figures 25-29 and 25-30.

Compensation of localized regions

Compensation of a localized tooling region is possible, with the keyword *INCLUDE_-COMPENSATION_CURVE, by incorporating the following lines into the above example inputs:

```
*INCLUDE_COMPENSATION_CURVE
curves.k
```

The file curves.k defines the two enclosed "begin" and "end" curves using *DEFINE_-CURVE_COMPENSATION_CONSTRAINT_BEGIN/END. More explanations can be found in the corresponding keyword manual entries. In Figure 25-31, the NUMISH-EET'05 decklid inner is compensated locally in the horizontal area above the backlite. Tangency of the compensated tool is maintained at the "End Curve" as shown in the section A-A. Also shown in Figure 25-32 includes color contours of part-separation distance throughout the iterations between compensated panel and the target design intent. Part tolerance is achieved in two iterations.

Accelerated spring back compensation (ASC)

The option ACCELERATOR can be used in conjunction with *INCLUDE_COMPENSA-TION, with options ORIGINAL_DYNAIN and SPRING BACK_INPUT to compensate spring back with a faster convergence rate and a simplified user interface. A complete example is provided below. The example uses a spring back input file spbk.dyn, and a trimmed panel, with file name case20trimmed.dynain (including all stress and strain tensors and adaptive constraints). The variable ISTEPS is increased from 0 to 3, representing 3 compensation iterations. ISTEPS = 0 represents the baseline spring back simulation (ITER0); while ISTEPS = 1, 2, 3 represent the compensation iterations. This feature requires the user to change only one variable (ISTEPS), and then submit the same input file to continue the next iteration.

Many scratch files, including a file named acceltmp.tmp, will be generated. *Do not delete them*. They are used to pass data between steps. A file, compensation.info, is generated and updated after each ISTEPS calculation. It contains iteration information, including maximum deviations in the *x*, *y*, and *z* directions. When the maximum deviation is within the tolerances specified in the TOLX, TOLY, and TOLZ fields, a message appears in the file indicating the compensation iterations have converged, along with a message bootstrapping the next step. A file spbk.new is generated in the same directory and is used by the *INCLUDE_COMPENSATION_BLANK_AFTER_-SPRING BACK keyword with the scale factor for the tool compensation set to one. After the compensation, a verification calculation may be needed.

```
*KEYWORD
*INTERFACE_COMPENSATION_NEW_ACCELERATOR
$   ISTEPS      TOLX      TOLY      TOLZ    OPTION
         3      0.20      0.20       0.2         1
*INCLUDE_COMPENSATION_ORIGINAL_DYNAIN
./case20trimmed.dynain
*INCLUDE_COMPENSATION_SPRING BACK_INPUT
./spbk.dyn
*END
```

Currently, mesh coarsening and checking are not supported in the accelerated mode. Also, the dynain file from the previous die process is not necessary.

An example of this feature is shown for a simple channel type of draw (one-half model) in Figure 25-33, which converged in three iterations; while four iterations were needed for the non-accelerated compensation.

Line die compensation

The option MULTI_STEPS can be used together with *INCLUDE_COMPENSATION_-COMPENSATED_SHAPE_NEXT_STEP to enable compensation of tools for the next die process. An example is given below. In this example the target blank, named reference0.tmp, and the current tool named rigid.tmp come from the first die process. The disp.tmp file comes from the compensation in the second die process step. For example, a flanging die compensation can be a second die process step, preceded by a redraw die process as the first die process step.

```
*KEYWORD
*INTERFACE_COMPENSATION_NEW_MULTI_STEPS
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$   METHOD        SL        SF     ELREF      PSID    UNDRCT     ANGLE   NLINEAR
         8     6.000      1.00         1         1         0         0         1
*INCLUDE_COMPENSATION_DESIRED_BLANK_SHAPE
reference0.tmp
*INCLUDE_COMPENSATION_COMPENSATED_SHAPE_NEXT_STEP
disp.tmp
*INCLUDE_COMPENSATION_CURRENT_TOOLS
rigid.tmp
*SET_PART_LIST
```

```
$      PSID
          1
$       PID
          2
*END
```

Compensation of trim dies (trim die nesting)

The trim die can be compensated using the drawn panel's springnack shape when METHOD is set to -8. In the example below, which is also shown in Figure 25-34, the draw panel, state1.k, is taken as the blank before spring back, and, draw panel spring back shape, state2.k, is taken as the blank after spring back. The tool shape for the draw process, drawtool.k, is used as the current tool. After the simulation, LS-DYNA will create a compensated tool named rigid.new, which can be used for the trim die shape.

```
*KEYWORD
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*INTERFACE_COMPENSATION_NEW
$   METHOD        SL        SF     ELREF      PSID    UNDRCT     ANGLE   NLINEAR
        -8    10.000     1.000         2         1         0       0.0         1
*INCLUDE_COMPENSATION_BLANK_BEFORE_SPRINGBACK
state1.k
*INCLUDE_COMPENSATION_BLANK_AFTER_SPRINGBACK
state2.k
*INCLUDE_COMPENSATION_DESIRED_BLANK_SHAPE
ref0.tmp
*INCLUDE_COMPENSATION_COMPENSATED_SHAPE
ref1.tmp
*INCLUDE_COMPENSATION_CURRENT_TOOLS
drawtool.k
*INCLUDE_COMPENSATION_TRIM_CURVE
orginaltrim.k
*SET_PART_LIST
$      PSID
         1
$      PID
         3
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*END
```

Local smoothing of tooling mesh:

The option LOCAL_SMOOTH can be used, along with a few more keywords, to smooth and restore the distorted tooling mesh after iterative compensation. In the example below, the keyword *INCLUDE_COMPENSATION_ORIGINAL_RIGID_-TOOL includes an original tool, rigid.tmp, which has a good and smooth mesh. The keyword *INCLUDE_COMPENSATION_NEW_RIGID_TOOL includes a compensated tool, rigidnew.bad, which could have a distorted mesh arising from the reasons listed in the "Purpose" section of this keyword.

The last keyword *SET_NODE_LIST_SMOOTH defines a node set in and surrounding a distorted local area in the distorted mesh. Each node set defines a region needing smoothing. The node set should not include any boundary nodes of the tooling parts, otherwise position of the tooling may be altered undesirably. Smoothed tooling is stored in a file called rigid.new. In this example method 7 is active, the variable ELREF is set to 2, and PSID left as undefined. Note also the *INCLUDE keyword is not supported here. For example, *SET_NODE_LIST_SMOOTH must not be in a separate file and included in the main input file.

```
*KEYWORD
*INTERFACE_COMPENSATION_NEW_LOCAL_SMOOTH
$   METHOD        SL        SF     ELREF      PSID    UNDRCT     ANGLE   NLINEAR
         7    10.000     1.000         2                   0       0.0         1
*INCLUDE_COMPENSATION_ORIGINAL_RIGID_TOOL
rigid.tmp
*INCLUDE_COMPENSATION_NEW_RIGID_TOOL
rigidnew.bad
*SET_NODE_LIST_SMOOTH
         1
     61057     61058     61059     61060     61061     61062     61063     61064
...
*SET_NODE_LIST_SMOOTH
         2
     56141     56142     56143     56144     56145     56146     56147     56148
...
*END
```

In an example shown in Figures 25-35 and 25-36, smoothing of the local mesh is performed in the draw bead area of the NUMISHEET 2005 cross member. In this case the die gap is not maintained throughout the tooling surface. Typically this happens in the draw bead regions when male beads have lower bending radii (missing upper radii) and female beads have only upper bending radii (missing lower radii). Two node sets are defined for local areas of left and right female draw beads (Figure 25-37), which needed smoothing. It is important to include in the node sets some of the nodes on the relatively flat portion of the binder immediately off the bend radii. Smoothed results show original distorted meshes on the lower beads corner areas are corrected and is satisfactory, Figure 25-38.

In another example, a corner of a flanging die on a fender outer is being smoothed. The mesh becomes distorted after a few compensation iterations, as shown in Figure 25-39. In Figure 25-40, the result of local smoothing is shown, and the improvement is remarkable.

Compensation with symmetric boundary condition

A keyword example is provided in the manual pages related to *INCLUDE_COMPENSATION_{OPTION}.

Global compensation using the original tool mesh

For some tooling meshes, the compensated die surfaces will be distorted. The keyword option *INCLUDE_COMPENSATION_ORIGINAL_TOOL causes the compensation code to use the original tooling mesh (starting in the second compensation) to extrapolate the addendum and binder in the compensated tooling surfaces. This minimizes the accumulative error, compared with using the last compensated tooling mesh, and therefore is a preferred method.

A complete keyword example is included below. In it, part ID 3 (included in part set ID 1) is being compensated after ITERATION #3 (ITER3), using method #8, with a scale factor of 0.5. The dynain files from the trimmed ITER3 is taken as the "BEFORE" state (see the _BEFORE_SPRINGBACK option) and the dynain file from the springback calculation is taken as the "AFTER" state. The "DESIRED" blank shape is given by the dynain is from the trimmed ITER0 output. The "COMPENSATED_SHAPE" is taken from the disp.tmp file of the last compensation run. "CURRENT_TOOL" is also from last compensation iteration. The "ORIGINAL_TOOL", is taken from the tool mesh in ITER0. Updated trim curves geotrm.new are taken from the mapped trim lines of last compensation.

It should be noted that, in an automatic compensation-loop calculation, as shown in the path of the input files, input files disp.tmp, rigid.new, and geotrm.new, taken from the default file names of the previous compensation, should not be in the same directory as the current compensation run, as these files *will* be overwritten.

```
*KEYWORD
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*INTERFACE_COMPENSATION_NEW
$   METHOD        SL        SF     ELREF      PSID    UNDRCT     ANGLE   NLINEAR
         8    10.000     0.500         2         1         1       0.0         1
*INCLUDE_COMPENSATION_BLANK_BEFORE_SPRINGBACK
../7_iter3.dir/2_trim.dir/dynain
*INCLUDE_COMPENSATION_BLANK_AFTER_SPRINGBACK
../7_iter3.dir/3_spbk.dir/dynain
*INCLUDE_COMPENSATION_DESIRED_BLANK_SHAPE
../1_iter0.dir/2_trim.dir/dynain
*INCLUDE_COMPENSATION_COMPENSATED_SHAPE
../6_compensation.dir/disp.tmp
*INCLUDE_COMPENSATION_CURRENT_TOOLS
../6_compensation.dir/rigid.new
*INCLUDE_COMPENSATION_ORIGINAL_TOOLS
../1_iter0.dir/sim_forming_mesh.k
*INCLUDE_COMPENSATION_TRIM_CURVE
../6_compensation.dir/geotrm.new
*SET_PART_LIST
$      PSID
         1
$      PID
         3
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*END
```

Updating compensated tool with small amount of part shape change

Often times a part will have some small amount of shape change as a result of a product change. If the amount of shape change does not significantly alter the spring back results, the compensated tools can be updated with the part mesh (inside the trim lines) or formed blank shape without going through another iterative compensation loop. This is accomplished using the PART_CHANGE option. Within the specified MAXGAP, compensated tool shape can be updated. Changes to geometry involving sharp corners and transition with no fillet are not permissible. A complete keyword example is provided below, where a maximum gap of 5mm is specified between the original shape and modified product shape. The updated part file name is updatepart.tmp and output file for the new rigid tool is newrigid.k.

```
*KEYWORD
*INTERFACE_COMPENSATION_NEW_PART_CHANGE
$   MAXGAP
       5.0
*INCLUDE_COMPENSATION_DESIRED_BLANK_SHAPE
../1_iter0.dir/2_trim.dir/dynain
*INCLUDE_COMPENSATION_COMPENSATED_SHAPE
../6_compensation.dir/disp.tmp
*INCLUDE_COMPENSATION_CURRENT_TOOLS
../6_compensation.dir/rigid.new
*INCLUDE_COMPENSATION_UPDATED_BLANK_SHAPE
./updatedpart.tmp
*INCLUDE_COMPENSATION_UPDATED_RIGID_TOOL
$ file name to output the new rigid tools
./newrigid.k
*END
```

Option REFINE_RIGID: rigid tool mesh refinement for a better convergence

The following keyword example refines and breaks the elements in the original tool mesh file rigid.new along provided trim curves trimcurve106.k, defined using the keyword *DEFINE_CURVE_TRIM_3D. The refined rigid tool mesh will be output as rigid_refined.tmp, which can be used to start the iterative springback compensation process.

```
*KEYWORD
*INTERFACE_COMPENSATION_NEW_REFINE_RIGID
rigid.new
trimcurve106.k
*END
```

Reference:

The manual pages related to *INCLUDE_COMPENSATION_{*OPTION*} can be further referenced for details.

Revision Information:

This keyword requires double precision executable. The option of ACCELERATOR is available starting in Revision 61264. The option of MULTI_STEPS is available starting in Revision 61406. The option of LOCAL_SMOOTH is available starting in Revision 73850. The keyword option *INCLUDE_COMPENSATION_ORIGINAL_TOOL is available starting in Revision 82701. The keyword option PART_CHANGE is available starting in Revision 82698. The REFINE_RIGID option is available starting in Revision 113089.

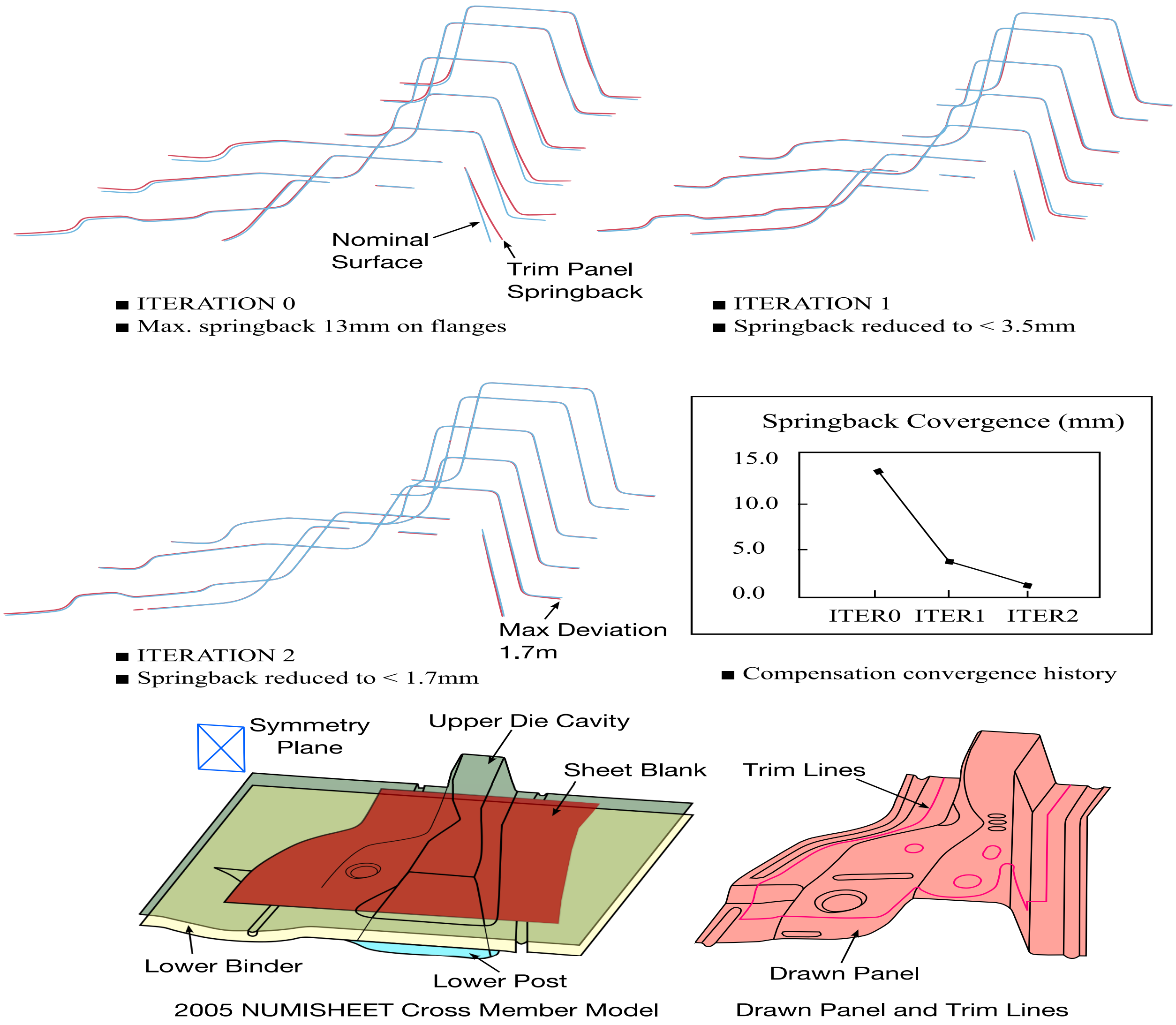

Nominal Surface
Trim Panel Springback
■ ITERATION 0
■ Max. springback 13mm on flanges
■ ITERATION 1
■ Springback reduced to < 3.5mm
■ ITERATION 2
■ Springback reduced to < 1.7mm
Max Deviation 1.7m
Springback Covergence (mm)
15.0
10.0
5.0
0.0
ITER0
ITER1
ITER2
■ Compensation convergence history
Symmetry Plane
Upper Die Cavity
Sheet Blank
Lower Binder
Lower Post
2005 NUMISHEET Cross Member Model
Trim Lines
Drawn Panel
Drawn Panel and Trim Lines

Figure 25-28. Iterative springback compensation on NUMISHEET'05 Xmbr(red – springback, blue – design intent)

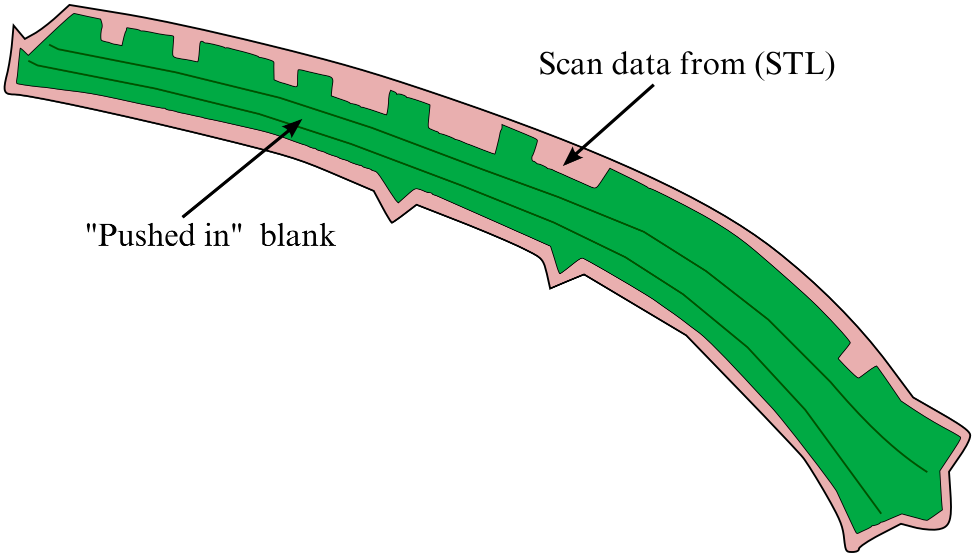


Figure 25-29. A trimmed panel “pushed” onto the scan data (rigid body).

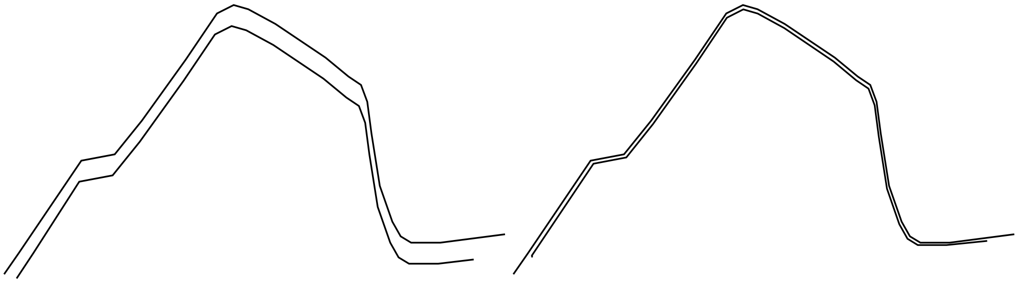


Figure 25-30. Section showing the “push” results – before and after.

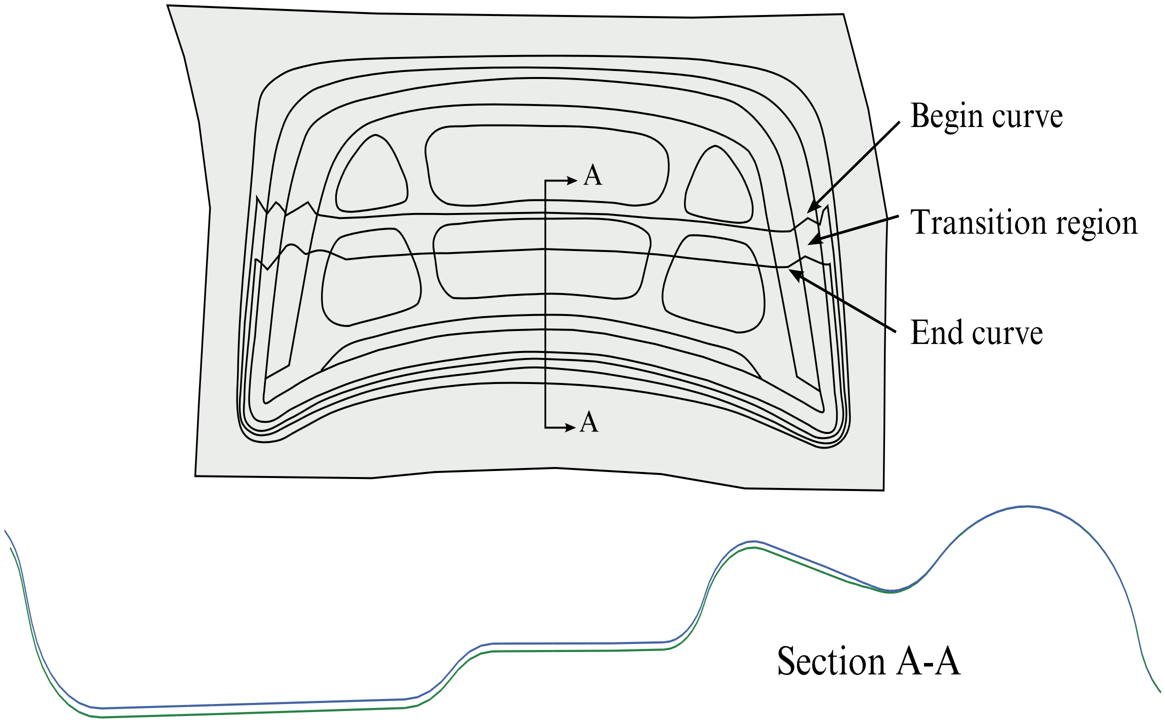


Figure 25-31. Two curves defining a localized area of a decklid inner.

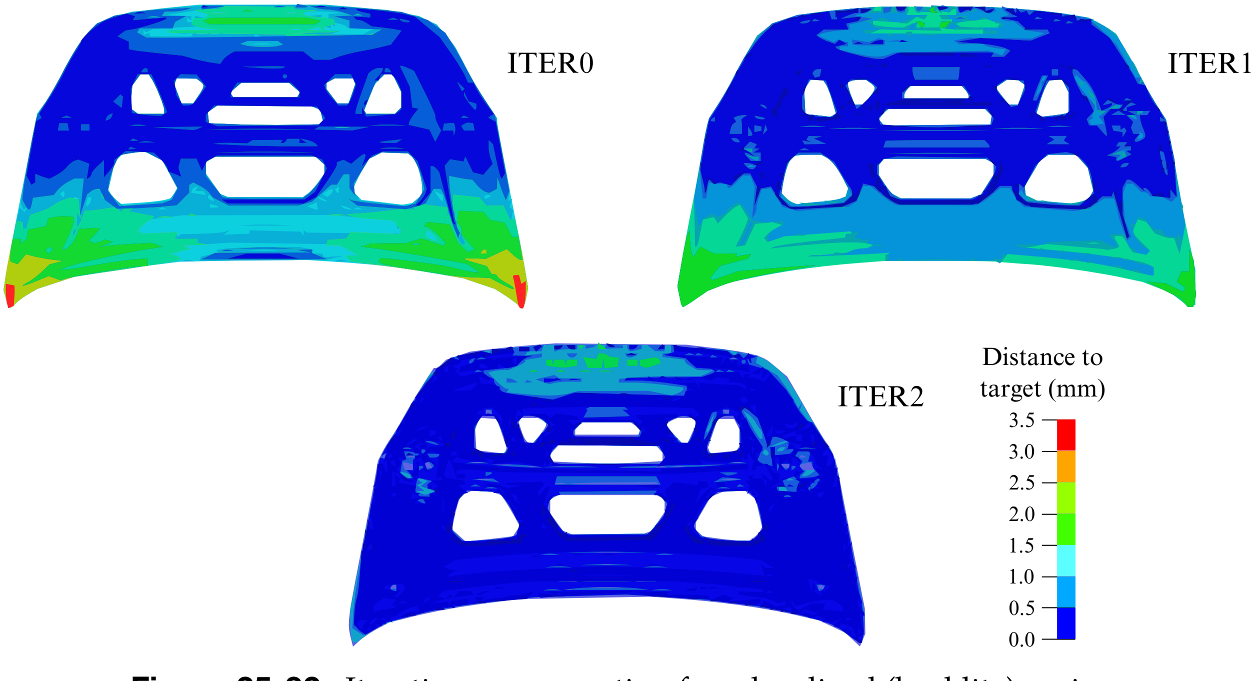


Figure 25-32. Iterative compensation for a localized (backlite) region.

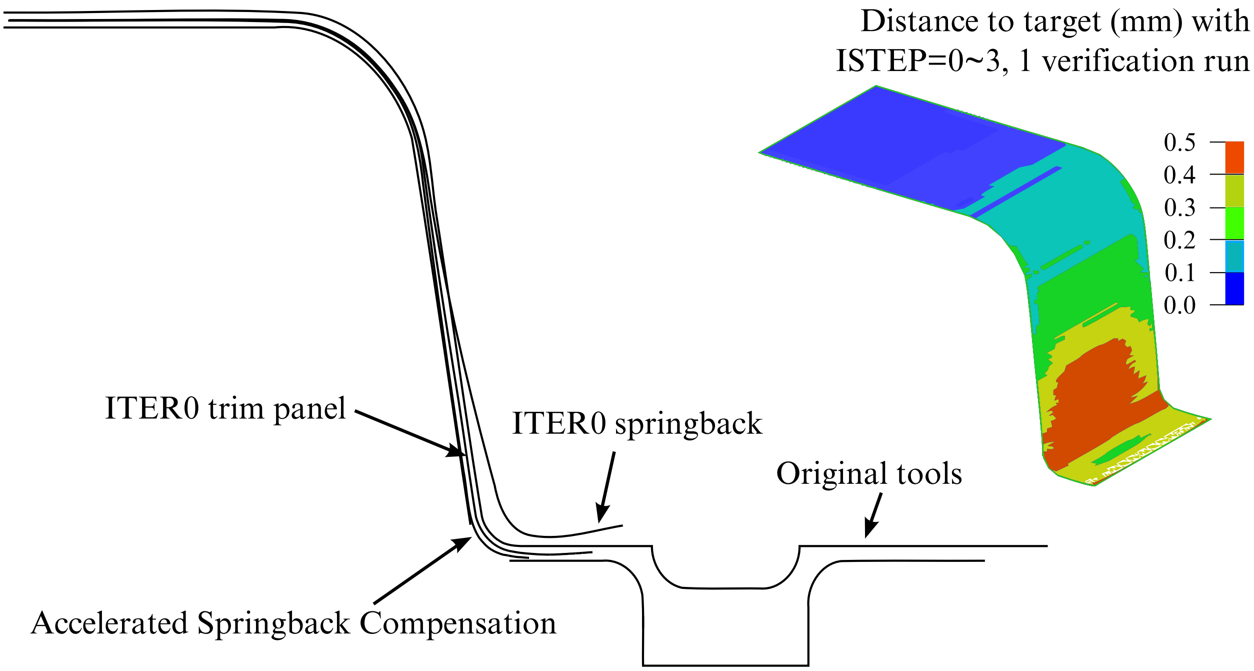

Distance to target (mm) with
ISTEP=0~3, 1 verification run
0.5
0.4
0.3
0.2
0.1
0.0
ITER0 trim panel
ITER0 springback
Original tools
Accelerated Springback Compensation

Figure 25-33. Accelerated Springback Compensation.

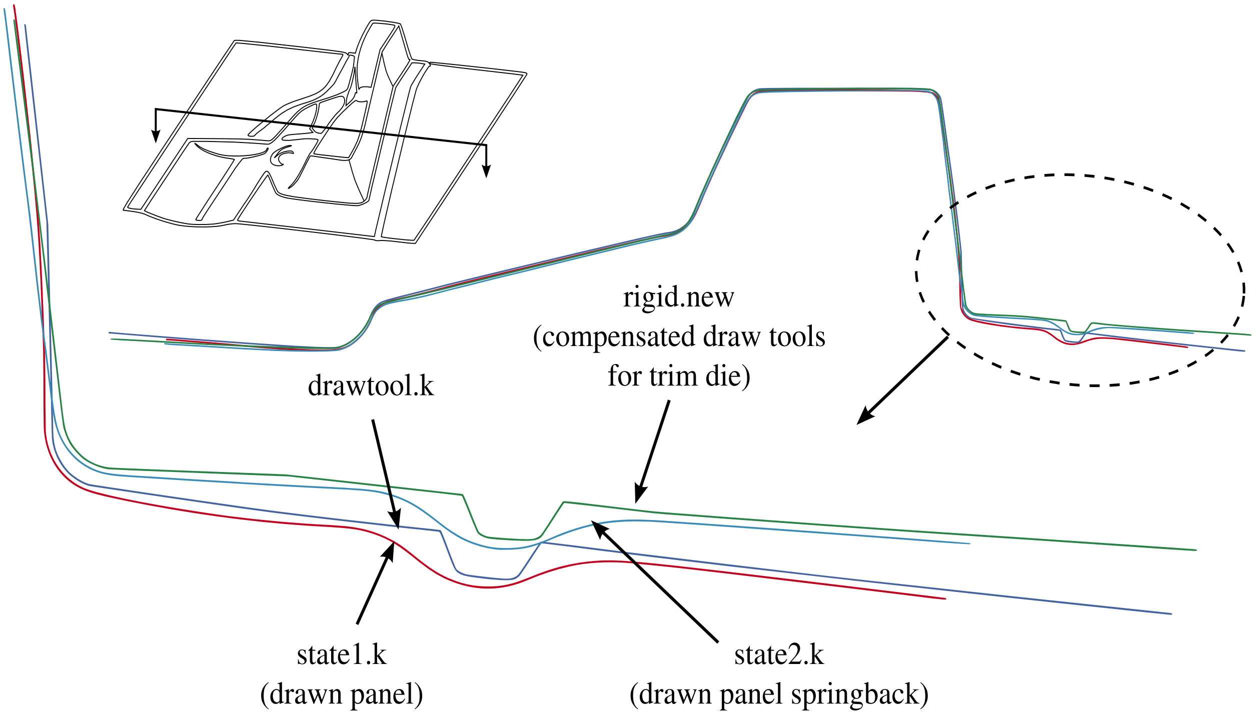


Figure 25-34. Trim die compensation with drawn panel springback shape.

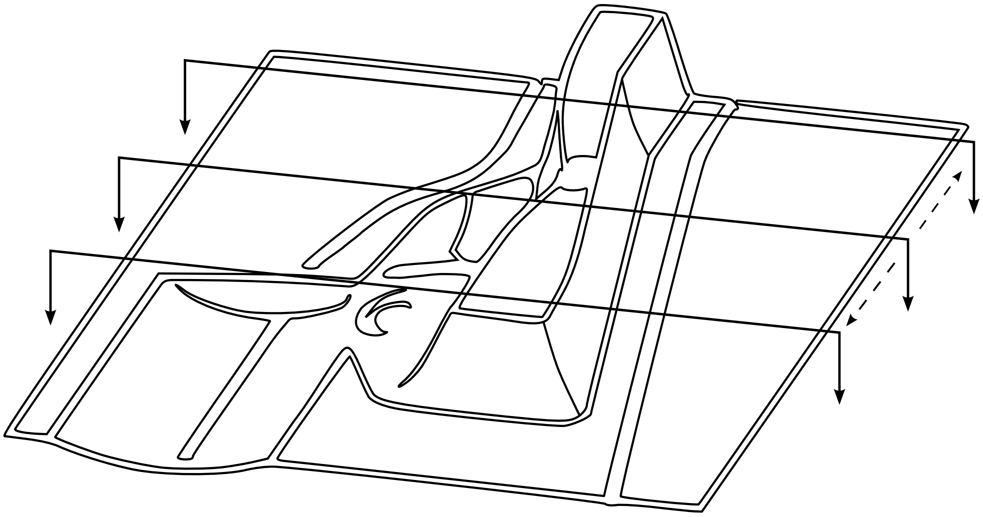


Figure 25-35. The NUMISHEET 2005 cross member.

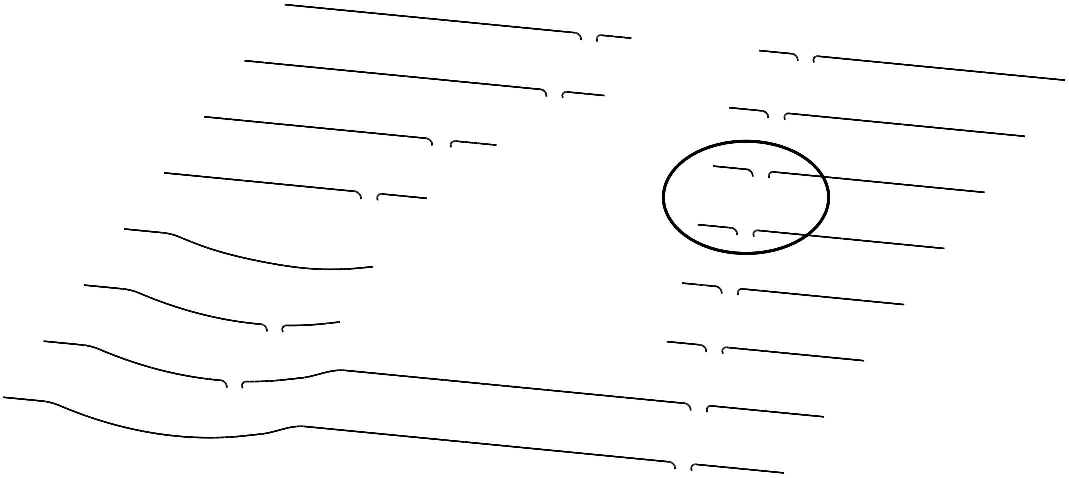


Figure 25-36. Multiple sections cut on the lower binder.

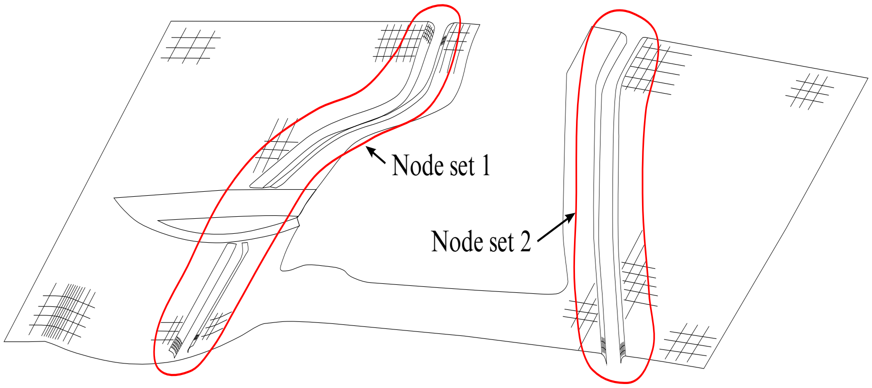


Figure 25-37. Local smoothing - two node sets defined including some nodes on the relatively flat binder area for both left and right draw beads.

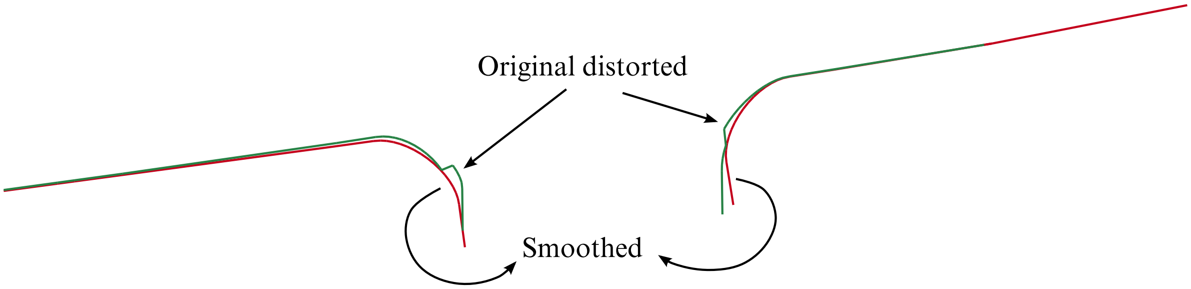


Figure 25-38. Comparison between original and smoothed tooling mesh.

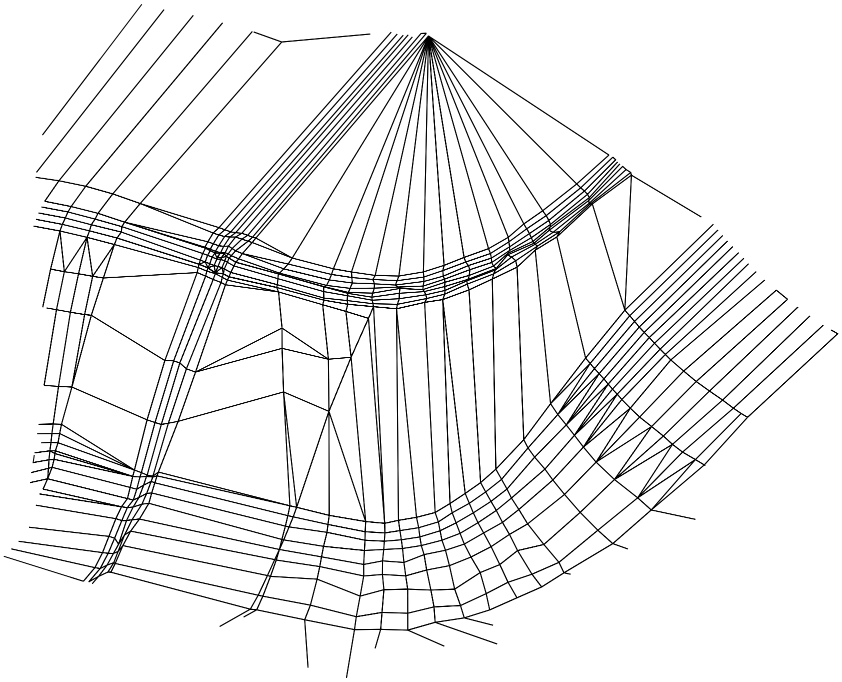


Figure 25-39. Original distorted tooling mesh.

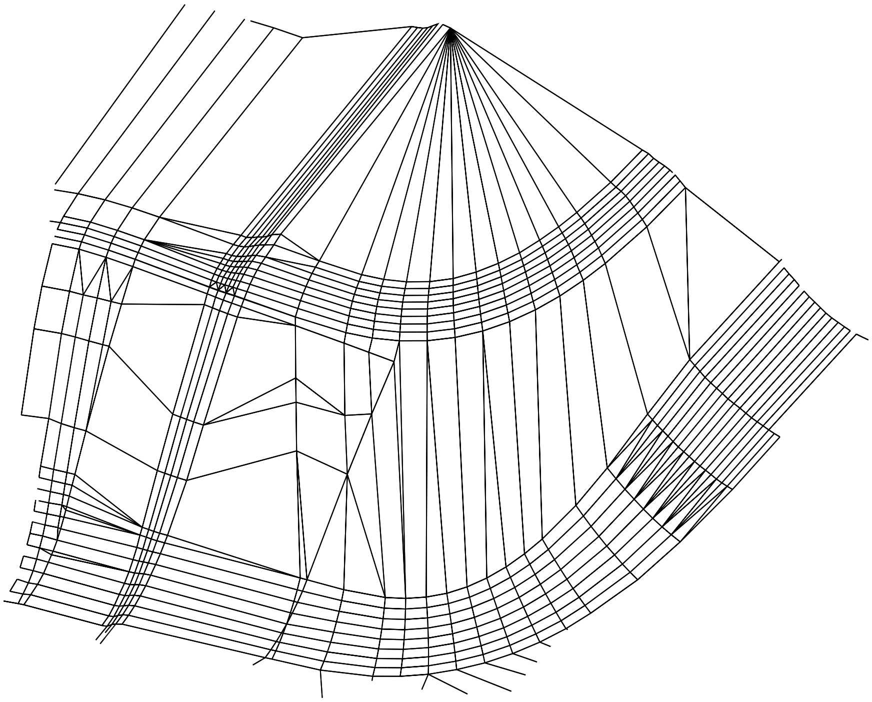


Figure 25-40. Smoothed tooling mesh.

*INTERFACE_COMPONENT_FILE

Purpose: Allow for the specification of the file where the component interface data should be written, and the optional use of a new binary format for that data.

Card 1	1	2	3	4	5	6	7	8
Variable	Filename							
Type	A80							
Default	none							

Optional Card.

Card 2	1	2	3	4	5	6	7	8
Variable	Format							
Type	I							
Default	2							

VARIABLE	DESCRIPTION
FNAME	Name of the file where the component data will be written
FORMAT	File format to use: EQ.1: Use old binary file format EQ.2: Use new LSDA file format

Remarks:

If Z = is used on the command line, this card will be ignored. If this card is in effect, the new LSDA file format is the default format to be used. The new format has certain advantages, and one possible drawback:

1. It allows for the use of the_TITLE modifier on all *INTERFACE_COMPONENT inputs, so that subsequent *INTERFACE_LINKING cards can refer to components by a user specified ID.

2. It is fully portable between machines with different precision and byte order.

3. It maintains the full precision of the coordinate vector. The internal coordinate vector has been in double precision for quite some time, even for single precision executables. The old binary format writes 32 bit data for single precision executables, losing some precision in the process.

4. Because of the maintained precision, the new format files will be significantly larger when running in single precision.

Of course, the new file format cannot be used for subsequent analysis with older versions of LS-DYNA, particularly those with a Product ID less than 50845. Executables which can read the new format for *INTERFACE_LINKING analysis will automatically detect whether the new or old format is in use.

*INTERFACE_COMPONENT_*OPTION1*_*{OPTION2}*

Available values for OPTION1 include:

NODE

SEGMENT

OPTION2 only allows the value:

TITLE

Purpose: Create an interface for use in subsequent linking calculations. This command applies to the first analysis for storing interfaces in the interface file specified either by "Z=**isf1**" on the execution line or by the *INTERFACE_COMPONENT_FILE command. The output interval used to write data to the interface file is controlled by OPIFS on *CONTROL_OUTPUT. If OPIFS is not specified, the interval defaults to 1/10th the value of DT specified in *DATABASE_BINARY_D3PLOT.

This capability allows the definition of interfaces that isolate critical components. A database is created that records the motion of the interfaces. In later calculations the isolated components can be reanalyzed with arbitrarily refined meshes with the motion of their boundaries specified by the database created by this input. The interfaces defined here become the masters in the tied interface options.

Each definition consists of a set of cards that define the interface. Interfaces may consists of a set of segments for later use with *INTERFACE_LINKING_SEGMENT, an ordered line of nodes for use with *INTERFACE_LINKING_EDGE, or an unordered set of nodes for use with *INTERFACE_LINKING_NODE.

Title Card. Additional card for TITLE keyword option.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	Title						
Type	I	A70						
Default	none	None						

VARIABLE	DESCRIPTION
ID	ID for this interface in the linking file
Title	Title for this interface

Card 2	1	2	3	4	5	6	7	8
Variable	SID	CID	NID					
Type	I	I	I					

VARIABLE	DESCRIPTION
SID	Set ID, see *SET_NODE or *SET_SEGMENT.
CID	Coordinate system ID
NID	Node ID

Remarks:

CID and NID are optional. If CID appears, the transformation matrix for this coordinate system is written to the linking file at each output state. If NID appears, the displacement of this node is also written to the file. This information is then available to be used by the *INTEFACE_LINKING_NODE_LOCAL. If either of these is non-zero, then the linking file will be written in the LSDA format, as the old format cannot support this optional output.

If the old style binary format is used for the linking file (see *INTERFACE_COMPONENT_FILE) then the ID values are ignored and all components are numbered according to their input order, starting from 1.

*INTERFACE_DE_HBOND

Purpose: Define the failure models for bonds linking various discrete element (DE) parts within one heterogeneous bond (*DEFINE_DE_HBOND).

Card 1	1	2	3	4	5	6	7	8
Variable	IID							
Type	I							
Default	none							

Bond Definition Cards. For each bond definition, include an additional card. This input ends at the next keyword ("*") card.

Optional	1	2	3	4	5	6	7	8
Variable	PID1	PID2	PTYPE1	PTYPE2	FRMDL	FRGK	FRGS	DMG
Type	I	I	I	I	I	F	F	F
Default	none	none	none	none	1	none	none	1.0

VARIABLE	DESCRIPTION
IID	Interface ID. All interfaces should have a unique ID
PID1	First part ID.

VARIABLE	DESCRIPTION
PID2	Second part ID. PID1 and PID2 define the bonds that this fracture model is applied to. There are three combinations as Case a: PID1.EQ.0 This is the default model for all bonds, overriding the default model defined in Card 2 of *DEFINE_DE_HBOND. Case b: PID1.GT.0 and PID2.EQ.0 This model is applied to the bonds within part PID1, instead of the default model. Case c: PID1.GT.0 and PID2.GT.0 This model is applied to the bonds between parts PID1 and PID2 only, but not to those within part PID1 or part PID2 (as in case b). Notes: 1. The default fracture model is applied to all parts that are not specified in case b. 2. The fracture model of the part with a smaller part id is applied to the bonds between two different parts if not specified in case c.
PTYPE1	First part type: EQ.0: DES part set EQ.1: DES part
PTYPE2	Second part type: EQ.0: DES part set EQ.1: DES part
FRMDL	Fracture model. (same as FRMDL in Card2 of keyword *DEFINE_DE_HBOND.)
FRGK	Fracture energy release rate for volumetric deformation. (same as FRGK in Card2 of keyword *DEFINE_DE_HBOND.)
FRGS	Fracture energy release rate for shear deformation. (same as FRGS in Card 2 of keyword *DEFINE_DE_HBOND.)
DMG	Continuous damage development. (same as DMG in Card 2 of keyword *DEFINE_DE_HBOND.)

INTERFACE_DE_HBOND EXAMPLE:

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  *INTERFACE_DE_HBOND
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$  Using DE_HBOND to bond 4 parts with different failure models
$
*SET_PART_LIST_TITLE
DES BOND PARTS
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      sid       da1       da2       da3       da4
         1
$      id1       id2       id3       id4
       101       102       103       104
*SET_PART_LIST_TITLE
DES HBOND SUB PARTS
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      sid       da1       da2       da3       da4
         2
$      id1       id2
       103       104
*INTERFACE_DE_HBOND
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      iid
         1
$     pid1      pid2    ptype1    ptype2     frmdl      frgk      frgs       dmg
       101         0         1         0         1     1.0E1     1.0E1       1.0
         2         0         0         0         1     1.0E3     1.0E3       1.0
       102       103         1         1         2     1.0E2     1.0E2       1.0
$
*DEFINE_DE_HBOND
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      sid     stype    bdform      idim
         1         2         2         3
$   pbk_sf    pbs_sf      frgk      frgs    bondr     alpha       dmg     frmdl
         1         1     1.0E0     1.0E0       2.5      0.01       1.0         1
$   precrk    cktype               itfid
         0         0                   1
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$ the fracture models for the bonds within each part are determined as
$
$   pid1    pid2    frgk    comments
$    101     101   1.0E1    Line 1 in the interface card
$    102     102   1.0E0    default (defined in hbond card)
$    103     103   1.0E3    Line 2 in the interface card
$    104     104   1.0E3    Line 2 in the interface card
$
$ for the bonds between two parts:
$    101     102   1.0E1    taken from Part 101 (smaller part id)
$    101     103   1.0E1    taken from Part 101 (smaller part id)
$    101     104   1.0E1    taken from Part 101 (smaller part id)
$    102     103   1.0E2    Line 3 in the interface card
$    102     104   1.0E0    taken from Part 102 (smaller part id)
$    103     104   1.0E3    taken from Part 103 (smaller part id)
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

INTERFACE_LINKING_DISCRETE_NODE_*OPTION

Available options include:

NODE

SET

Purpose: Link node(s) to an interface in an existing interface file. This link applies to all element types. The interface file is specified using *INTERFACE_LINKING_FILE or by including "L=**filename**" on the execution line.

With this command, nodes in a node set must be given in the same order as they appear in the interface file. This restriction does not apply to the more recent keyword *INTERFACE_LINKING_NODE_….

Card 1	1	2	3	4	5	6	7	8
Variable	NID/NSID	IFID						
Type	I	I						

VARIABLE	DESCRIPTION
NID	Node ID or Node set ID to be moved by interface file, see *NODE or *SET_NODE.
IFID	Interface ID in interface file.

*INTERFACE_LINKING_EDGE

Purpose: Link a series of nodes to an interface in an existing interface force file. The interface file is specified using *INTERFACE_LINKING_FILE or by including "L=**filename**" on the execution line.

Card	1	2	3	4	5	6	7	8
Variable	NSID	IFID						
Type	I	I						

VARIABLE	DESCRIPTION
NSID	Node set ID to be moved by interface file.
IFID	Interface ID in interface file.

Remarks:

The set of nodes defined will be constrained to follow the movement of the interface IFID, which should correspond to a curve output on a previous analysis via *INTERFACE_COMPONENT_NODE. The order of the nodes in the first analysis is important. The nodes, in the order specified in the first analysis, represent a curve. Each node in set NSID will be tied to the point on the curve nearest to its initial position, and then will be constrained to follow that point on the curve for the duration of the analysis. This option is intended to be used with beam or shell elements, as both translational and rotational degrees of freedom are constrained.

*INTERFACE_LINKING_FILE

Purpose: Allow for the specification of the file from which the component interface data should be read.

Card 1	1	2	3	4	5	6	7	8
Variable	Filename							
Type	A80							
Default	none							

VARIABLE	DESCRIPTION
FNAME	Name of the file from which the component data will be read

Remarks:

If L= is used on the command line, this card will be ignored. There is no option to specify the file format, as the file format is automatically detected.

INTERFACE_LINKING_NODE_*OPTION

Available options include:

SET

LOCAL

SET_LOCAL

Purpose: Link nodes(s) to an interface in an existing interface file. This link applies to all element types. The interface file is specified using *INTERFACE_LINKING_FILE or by including "L=**filename**" on the execution line.

Node/Set ID Card. Include as many cards as desired. Input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	NID/NSID	IFID	FX	FY	FZ			
Type	I	I	I	I	I			

VARIABLE	DESCRIPTION
NID	Node ID or Node set ID to be moved by interface file, see *NODE or *SET_NODE.
IFID	Interface ID in interface file.
FX	The ID of a *DEFINE_FUNCTION which determines the x direction displacement scale factor
FY	The ID of a *DEFINE_FUNCTION which determines the y direction displacement scale factor
FZ	The ID of a *DEFINE_FUNCTION which determines the z direction displacement scale factor

Card 2. This card appears after Card 1 when the_LOCAL option is used

Card 2	1	2	3	4	5	6	7	8
Variable	LCID	LNID	USEC	USEN				
Type	I	I	I	I				

VARIABLE **DESCRIPTION**

LCID Local coordinate system ID for transforming displacements.

LNID Local node ID for transforming displacements

USEC 0/1 flag to indicate the use of the coordinate system in the linking file during displacement transformation

USEN 0/1 flag to indicate the use of the node displacement in the linking file during displacement transformation.

Remarks:

The set of nodes is constrained to follow the displacement of the interface having ID IFID in the linking file. Note that the linking file is usually generated by the *INTERFACE_COMPONENT_NODE kekyword in a previous analysis.

The order of the nodes is not important. Each node in set NID will be tied to the nearest node in IFID using a bucket sort during the initialization phase. Nodes not found are reported and subsequently not constrained. Translational degrees of freedom are constrained. If the constrained analysis has rotational degrees of freedom, then the rotation degrees of freedom will be likewise constrained and the linking file *must* include rotational degrees of freedom.

The displacements in the linking file can be scaled upon input so that

$$\mathbf{u}_{\text{constrained}} = \begin{bmatrix} f_{\text{FX}}(\dots) & 0 & 0 \\ 0 & f_{\text{FY}}(\dots) & 0 \\ 0 & 0 & f_{\text{FZ}}(\dots) \end{bmatrix} \mathbf{u}_{\text{linking file}}$$

where $f_{\text{FX}}(\dots)$ is the *DEFINE_FUNCTION function having ID FX and so on. When a scaling function is not specified the corresponding component is imported unscaled as if the scaling function had a constant value of unity. These functions may take either 0, 1, 3, or 4 input arguments. The functions FX, FY, and FZ may be different and they may take different numbers of arguments. The data passed into the scaling function depends on the number of arguments that the function takes and the possibilities can be broken down into four cases:

1. *0 variables*. A function taking no inputs is evaluate constant over space and time. LS-DYNA evaluates such a function at the start of the calculation and uses that value for the duration of the run.
2. *1 varriables.* LS-DYNA passes in the simulation time at each step and the resulting value is applied to all nodes in the set.
3. *3 varriables.* LS-DYNA passes in the *initial position* of each constrained node as an (x, y, z) triple at the start of the calculation and then uses the result for the duration of the run.
4. *4 varriables*. LS-DYNA passes in the *current* simulation time and the *initial position* of each of the constrained nodes as an (x, y, z, t) tuple. This function is updated at each time step. Using scaling functions of 4 variables may result in a performance penalty as each function must be evaluated for every slave node every cycle.

If time dependent scaling functions are used, then the constrained nodes must start with coordinates identical to the constraining nodes in the linking file.

The LOCAL option and the values of the LCID, LNID, USEC, USEN flags, which was designed in conjunction with Honda R&D Co., Ltd., allow for the interface displacements to be transformed in various ways. By default, the scale factors FX, FY, FZ act on the nodal displacements in the global coordinate system of the constrained calculation. This may be undesirable depending on how the global coordinate system of the linked calculation is defined. The most general transformation rule is:

$$\mathbf{u}_{\text{constrained}} = \mathbf{Q}_2 \begin{bmatrix} f_{\text{FX}}(\dots) & 0 & 0 \\ 0 & f_{\text{FY}}(\dots) & 0 \\ 0 & 0 & f_{\text{FZ}}(\dots) \end{bmatrix} \mathbf{Q_1}(\mathbf{u}_{\text{linked}} - \mathbf{c_1}) - \mathbf{c_2}$$

where,

$\mathbf{c}_1$ = The displacement of the NID node in the linking file

$\mathbf{c_2}$ = The displacement of node LNID

$\mathbf{Q}_1$ = Rotation into the local coordinates of the linking file

$\mathbf{Q}_2$ = Rotation into the local coordinate system, if unset the inverse of $\mathbf{Q}_1$

If USEC = 0, then $\mathbf{Q}_1$ is the identity rotation and any coordinate system in the linking file is ignored. If USEN = 0, then $\mathbf{c}_1$ is set to 0 and NID in the linking file is ignored.

*INTERFACE_LINKING_SEGMENT

Purpose: Link segments to an interface in an existing interface file. The interface file is specified using *INTERFACE_LINKING_FILE or by including "L=**filename**" on the execution line.

Segment Set ID Card. Include as many cards as desired. Input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	SSID	IFID						
Type	I	I						

VARIABLE	DESCRIPTION
SSID	Segment set to be moved by interface file.
IFID	Interface ID in interface file.

Remarks:

The set of segments defined will be constrained to follow the movement of the interface IFID, which should correspond to an interface output on a previous analysis using the *INTERFACE_COMPONENT_SEGMENT keyword. The behavior will be the same as if set SSID is the slave side of a *CONTACT_TIED_SURFACE_TO_SURFACE with IFID as the master. Translational movement will be constrained, but not rotations.

*INTERFACE_SPRINGBACK_*OPTION1*_*OPTION2*

Available options included for *OPTION1* are:

LSDYNA

NASTRAN

SEAMLESS

EXCLUDE (see Remark 7)

and for *OPTION2:*

THICKNESS

NOTHICKNESS

See Remark 1.

Purpose: Define a material subset for an implicit springback calculation in LS-DYNA and any nodal constraints to eliminate rigid body degrees-of-freedom.

Card 1	1	2	3	4	5	6	7	8
Variable	PSID	NSHV	FTYPE		FTENSR	NTHHSV		INTSTRN
Type	I	I	I		I	I		I

Irregular Optional Card. The keyword reader will interpret the card following Card 1 as new optional Card 2 if the first column of the card is occupied by the string "OPTCARD". Otherwise, it is interpreted as first Node Card, see below.

Card 2	1	2	3	4	5	6	7	8
Variable	OPTC	SLDO	NCYC	FSPLIT	NDFLAG			
Type	A	I	I	I	I			

Node Cards. Define a list of nodal points that are constrained for the springback. This section is terminated by an "*" indicating the next input section.

Card	1	2	3	4	5	6	7	8
Variable	NID	TC	RC					
Type	I	F	F					

VARIABLE	DESCRIPTION
PSID	Part set ID for springback, see *SET_PART.
NSHV	Number of shell or solid history variables (beyond the six stresses and effective plastic strain) to be initialized in the interface file. For solids, one additional state variable (initial volume) is also written. If NSHV is nonzero, the element formulations, unit system, and constitutive models should not change between runs. If NHSV exceeds the number of integration point history variables required by the constitutive model, only the number required is written; therefore, if in doubt, set NHSV to a large number.
FTYPE	File type: (See Remark 5) EQ.0: ASCII, EQ.1: binary EQ.2: both ASCII and binary. EQ.3: LSDA format (for LSDYNA only) EQ.10: ASCII large format (see *INITIAL_STRESS_SHELL) EQ.11: binary large format EQ.12: both ASCII and binary large format
FTENSR	Flag for dumping tensor data from the element history variables into the dynain file. EQ.0: Don't dump tensor data from element history variables EQ.1: Dump any tensor data from element history variables into the dynain file in GLOBAL coordinate system. Currently, only Material 190 supports this option.

VARIABLE	DESCRIPTION
NTHHSV	Number of thermal history variables.
INTSTRN	Output of strains at all integration points of shell element is requested, see also *INITIAL_STRAIN_SHELL.
SLDO	Output of solid element data as EQ.0: *ELEMENT_SOLID, or EQ.1: *ELEMENT_SOLID_ORTHO.
NCYC	Number of process cycles this simulation corresponds to in the simulation of wear processes, see Remark 6.
FSPLIT	Flag for splitting of the dynain file (only for ASCII format). EQ.0: dynain file written in one piece. EQ.1: Output is divided into two files, dynain_geo including the geometry data and dynain_ini including initial stresses and strains.
NDFLAG	Flag to dump nodes into dynain file. EQ.0: default, dump only sph and element nodes EQ.1: dump all nodes
NID	Node ID, see *NODE.
TC	Translational Constraint: EQ.0: no constraints, EQ.1: constrained x displacement, EQ.2: constrained y displacement, EQ.3: constrained z displacement, EQ.4: constrained x and y displacements, EQ.5: constrained y and z displacements, EQ.6: constrained z and x displacements. EQ.7: constrained x, y, and z displacements.

VARIABLE **DESCRIPTION**

RC Rotational constraint:

EQ.0: no constraints,

EQ.1: constrained x rotation,

EQ.2: constrained y rotation,

EQ.3: constrained z rotation,

EQ.4: constrained x and y rotations,

EQ.5: constrained y and z rotations,

EQ.6: constrained z and x rotations,

EQ.7: constrained x, y, and z rotations.

Remarks:

1. **NOTHICKNESS Option.** The NOTHICKNESS option is available when the keyword's first option is either LS-DYNA or NASTRAN. With the NOTHICKNESS option the shell element thickness is not output.

2. **Filenames.** The file name for the LS-DYNA option is dynain and for NASTRAN is nastin.

3. **Trimming.** Trimming is available for the adaptive mesh, but it requires manual intervention. To trim an adaptive mesh use the following procedure:

   a) Generate the file, dynain, using the keyword *INTERFACE_SPRINGBACK_LSDYNA.

   b) Prepare a new input deck including the dynein file.

   c) Add the keyword *ELEMENT_TRIM to this new deck.

   d) Add the keyword *DEFINE_CURVE_TRIM to this new deck.

   e) Run this new input deck with i=**input_file_name**. The adaptive constraints are eliminated by remeshing and the trimming is performed.

   f) In case this new trimmed mesh is needed, run a zero termination time job and output the file generated via the keyword, *INTERFACE_SPRINGBACK_LSDYNA.

4. **Temperature.** The file new_temp_ic.inc will be created for a thermal solution and a coupled thermal-mechanical solution. The file new_temp_ic.inc is a

KEYWORD include file containing *new temperature initial conditions* for the nodes belonging to the PSID.

a) For thermal user materials it is possible to dump thermal history variables. See the NTHHSV field.

5. **FTYPE.** The choice of format size in option FTYPE is only available for shell stresses and shell history data, see parameter LARGE on *INITIAL_STRESS_-SHELL. For solid and beam elements, always the large format is written to dynain, i.e. LARGE is automatically set to 1 on *INITIAL_STRESS_SOLID and *INITIAL_STRESS_BEAM respectively.

6. **NCYC.** When simulating wear processes, this represents the number of process cycles this particular simulation corresponds to and *INITIAL_CONTACT_-WEAR cards are generated accordingly in the dynain file (only ascii format supported). Cards will only be generated for nodes in contact interfaces associated with a *CONTACT_ADD_WEAR, and having SPR or MPR set to 2 on the first card on *CONTACT. This wear data is in a subsequent simulation accounted for NCYC times when modifying the worn geometry, or alternatively processed in LS-PrePost

7. **EXCLUDE.** This option is used to limit what data will be output to the LSDYNA dynain file. The input format is completely different, and consists of any number of keyword cards WITHOUT the leading *. These cards and their associated data will not be output. For example:

```
*INTERFACE_SPRINGBACK_EXCLUDE
BOUNDARY_SPC_NODE
CONSTRAINED_ADAPTIVITY
```

would output all the normal dynain data except for the SPC and adaptive constraints. The currently recognized keywords that can be excluded are:

BOUNDARY_SLIDING_PLANE
BOUNDARY_SPC_NODE
CONSTRAINED_ADAPTIVITY
DEFINE_COORDINATE_NODES
DEFINE_COORDINATE_VECTOR
ELEMENT_BEAM
ELEMENT_SHELL
ELEMENT_SOLID
INITIAL_STRAIN_SHELL
INITIAL_STRAIN_SOLID
INITIAL_STRESS_BEAM
INITIAL_STRESS_SHELL

INITIAL_STRESS_SOLID
INITIAL_TEMPERATURE_NODE
INITIAL_VELOCITY_NODE
NODE
REFERENCE_GEOMETRY

Remarks for Seamless Springback:

When seamless springback is invoked, the solution automatically and seamlessly switches from explicit or implicit dynamic to implicit static mode at the termination time, and continues to run the static springback analysis. Seamless springback can be activated in the original LS-DYNA input deck, or later using a small restart input deck. In this way, the user can decide to continue a previous analysis by restarting to add the implicit springback phase. (Another alternative approach to springback simulation is to use the keyword *INTERFACE_SPRINGBACK_LSDYNA to generate a dynain file after forming, and then perform a second simulation running LS-DYNA in fully implicit mode for springback. See Appendix P for a description of how to run an implicit analysis using LS-DYNA.

The implicit springback phase begins when the forming simulation termination time ENDTIM is reached, as specified with the keyword *CONTROL_TERMINATION. Since the springback phase is static, its termination time can be chosen arbitrarily (unless material rate effects are included). The default choice is 2.0 × ENDTIM, and can be changed using the *CONTROL_IMPLICIT_GENERAL keyword; see variables DT0 and NSBS..

Since the springback analysis is a static simulation, a minimum number of essential boundary conditions or Single Point Constraints (SPC's) can be input to prohibit rigid body motion of the part. These boundary conditions can be added for the springback phase using the input option on the *INTERFACE_SPRINGBACK_SEAMLESS keyword above.

If no boundary conditions are added with the SEAMLESS option an eigenvalue computation is automatically performed using the Inertia Relief Option to find any rigid body modes and then automatically constrain them out of the springback simulation (see *CONTROL_IMPLICIT_INERTIA_RELIEF). This approach introduces no artificial deformation and is recommended for many simulations.

An “SPR” option is available for several *CONTROL_IMPLICIT keywords to further control the implicit springback phase. Generally, default settings can be used, in which case the SPR option for *CONTROL_IMPLICIT keywords is not necessary.

To obtain accurate springback solutions, a nonlinear springback analysis must be performed. In many simulations, this iterative equilibrium search will converge without difficulty. If the springback simulation is particularly difficult, either due to nonlinear deformation, nonlinear material response, or numerical precision errors, a multi-step springback simulation will be automatically invoked. In this approach, the springback deformation is divided into several smaller, more manageable steps.

Two specialized features in LS-DYNA are used to perform multi-step springback analyses. The addition and gradual removal of artificial springs is performed by the artificial stabilization feature. Simultaneously, the automatic time step control is used to guide the solution to the termination time as quickly as possible, and to persistently retry steps where the equilibrium search has failed. By default, both of these features are active during a seamless springback simulation. However, the default method attempts to solve the springback problem in a single step. If this is successful, the solution will terminate normally. If the single step springback analysis fails to converge, the step size will be reduced, and artificial stabilization will become active. Defaults for these features can be changed using the following keywords:

- *CONTROL_IMPLICIT_GENERAL,
- *CONTROL_IMPLICIT_AUTO, and
- *CONTROL_IMPLICIT_STABILIZATION.

*INTERFACE_SSI_*{OPTION}*_ID

Purpose: This card creates a tied-contact soil-structure interface for use in a transient analysis of a soil-structure system subjected to earthquake excitation. This card allows the analysis to start from a static state of the structure, as well as to read in ground motions recorded on the interface in an earlier analysis.

Available options are:

<BLANK>

OFFSET

CONSTRAINED_OFFSET

LS-DYNA implements the effective seismic input method [Bielak and Christiano (1984)] for modeling the interaction of a non-linear structure with a linear soil foundation subjected to earthquake excitation. Note that any non-linear portion of the soil near the structure may be incorporated with the structure into a larger generalized structure, but the soil is assumed to behave linearly beyond a certain distance from the structure.

The effective seismic input method couples the dynamic scattered motion in the soil, which is the difference between the motion in the presence of the structure and the free-field motion in its absence, with the total motion of the structure. This replaces the distant earthquake source with equivalent effective forces adjacent to the soil-structure interface and allows truncation of the large soil domain using a non-reflecting boundary (e.g. *MAT_PML_ELASTIC) to avoid unnecessary computation. These effective forces can be computed using the free-field ground motion at the soil-structure interface, thus avoiding deconvolution of the free-field motion down to depth.

Non-linear behavior of the structure may be modeled by first carrying out a static analysis of the soil-structure system, and then carrying out the transient analysis with only the structure initialized to its static state. Because the transient analysis employs the dynamic scattered motion in the soil, the soil cannot have any static loads only it only the structure is subjected to static forces. Consequently, the structure must be supported by the static reactions at the soil-structure interface. Additionally, the soil nodes at the interface must be initialized to be compatible with the initial static displacement of the structure. LS-DYNA will do these automatically if the soil-structure interface is identified appropriately in the static analysis and reproduced in the transient analysis.

Thus, soil-structure interaction analysis under earthquake excitation may be carried out in LS-DYNA as follows:

1. Carry out a static analysis of the soil-structure system (e.g. using dynamic relaxation; see *CONTROL_DYNAMIC_RELAXATION), with the soil-structure interface identified using *INTERFACE_SSI_STATIC_ID

Optionally, carry out a free-field analysis to record free-field motions on the future soil-structure interface, using either *INTERFACE_SSI_AUX or *INTERFACE_SSI_AUX_EMBEDDED, for surface-supported or embedded structures respectively.

2. Carry out the transient analysis as a full-deck restart job (see *RESTART), with only the structure initialized to its static stress state (see *STRESS_INITIALIZATION), and the same soil-structure interface identified using *INTERFACE_SSI_ID with the same ID as in static analysis:

   a) The structure mesh must be identical to the one used for static analysis.

   b) The soil mesh is expected to be different from the one used for static analysis, especially because non-reflecting boundary models may be used for transient analysis.

   c) The meshes for the structure and the soil need not match at the interface.

   d) Only the structure must be subjected to static loads, via *LOAD_BODY_PARTS

   e) The earthquake ground motion is specified using *LOAD_SEISMIC_SSI, and/or read from motions recorded from a previous analysis using *INTERFACE_SSI_AUX or *INTERFACE_SSI_AUX_EMBEDDED.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	HEADING						
Type	I	A70						

Card 2	1	2	3	4	5	6	7	8
Variable	STRID	SOILID	SPR	MPR				
Type	I	I	I	I				
Default	none	none	0	0				

Optional card

Card 3	1	2	3	4	5	6	7	8
Variable	GMSET	SF	BIRTH	DEATH	MEMGM			
Type	I	F	F	F	I			
Default	none	1.	0.	10^{28}	2500000			

VARIABLE	DESCRIPTION
ID	Soil-structure interface ID. This is required and must be unique amongst all the contact interface IDs in the model.
HEADING	A descriptor for the given ID.
STRID	Segment set ID of base of structure at soil-structure interface.
SOILID	Segment set ID of soil at soil-structure interface.
SPR	Include the slave side in the *DATABASE_NCFORC and the *DATABASE_BINARY_INTFOR interface force files: EQ.1: slave side forces included.
MPR	Include the master side in the *DATABASE_NCFORC and the *DATABASE_BINARY_INTFOR interface force files: EQ.1: master side forces included.
GMSET	Identifier for set of recorded motions from *INTERFACE_SSI_AUX or *INTERFACE_SSI_AUX_EMBEDDED
SF	Recorded motion scale factor. (default = 1.0)
BIRTH	Time at which specified recorded motion is activated.
DEATH	Time at which specified recorded motion is removed: EQ.0.0: default set to 10^{28}
MEMGM	Size in words of buffer allocated to read in recorded motions

Remarks:

1. A tied contact interface (*CONTACT_TIED_SURFACE_TO_SURFACE) is created between the structure and the soil using the specified segment sets, with the soil segment set as the master segment set and the structure segment set as the slave. Naturally, the two segment sets should not have merged nodes and can be non-matching in general. However, the area covered by the two surfaces should match.

2. The options OFFSET and CONSTRAINED_OFFSET create the corresponding tied surface-to-surface contact interface.

3. The soil-structure interface ID is assigned as the ID of the generated contact interface.

4. It is assumed that the soil segment set is oriented toward the structure.

5. Multiple soil-structure interfaces are allowed, e.g. for bridge analysis.

6. The recorded motions are read in from a binary file named gmbin by default, but a different filename may be chosen using the option GMINP on the command line (see INTRODUCTION, Execution Syntax).

7. If the motions from *INTERFACE_SSI_AUX or *INTERFACE_SSI_AUX_EMBEDDED were recorded on a segment set, then the free-field motions on each node in the master segment set of the soil-structure interface are calculated from the nearest segment of the segment set used to record the motions.

   If, however, the motions were recorded on a node set, then the motions on the master segment set nodes is found by interpolation as is done for *LOAD_SEISMIC_SSI.

INTERFACE_SSI_AUX_*{OPTION}

Available options are:

<BLANK>

NODE

Purpose: This card records the motion at a free surface, or on a set of nodes on a free surface, for the purpose of using the recorded motion as a free-field motion in a subsequent interaction analysis using *INTERFACE_SSI. By default, this card records motions on a segment set defining a surface, but can record motions on a node set using the option NODE. Only one of *INTERFACE_SSI_AUX and *INTERFACE_SSI_AUX_-EMBEDDED is to be used for a particular soil-structure interface.

Card 1	1	2	3	4	5	6	7	8
Variable	GMSET	SETID						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
GMSET	Identifier for this set of recorded motions to be referred to in *INTERFACE_SSI. Must be unique.
SETID	Segment set or node set ID where motions are to be recorded.

Remarks:

1. The motions on the specified segment set or node set is recorded in a binary file named **gmbin** by default, but a different filename may be chosen using option GMOUT on the command line (see INTRODUCTION, Execution Syntax).

2. The output interval for the motions may be specified using the parameter GMDT on the *CONTROL_OUTPUT card, with the default value being 1/10-th of the output interval for D3PLOT states.

*INTERFACE_SSI_AUX_EMBEDDED_*{OPTION1}*_*{OPTION2}*

Purpose: This card creates a tied-contact interface and records the motions and contact forces in order to use them as free-field motion and reactions in a subsequent soil-structure interaction analysis using *INTERFACE_SSI, where the structure is embedded in the soil after part of the soil has been excavated. Only one of *INTERFACE_SSI_AUX and *INTERFACE_SSI_AUX_EMBEDDED is to be used for a particular soil-structure interface.

Available options for *OPTION1* are:

<BLANK>

OFFSET

CONSTRAINED_OFFSET

OPTION2 allows an optional ID to be given:

ID

ID Card. Additional card for ID keyword option.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	HEADING						
Type	I	A70						

Card 2	1	2	3	4	5	6	7	8
Variable	GMSET	STRID	SOILID	SPR	MPR			
Type	I	I	I	I	I			
Default	none	none	none	0	0			

VARIABLE	DESCRIPTION
ID	Soil-structure interface ID. This is required and must be unique amongst all the contact interface IDs in the model.
HEADING	A descriptor for the given ID.

VARIABLE **DESCRIPTION**

GMSET Identifier for this set of recorded motions to be referred to in *INTERFACE_SSI. Must be unique.

STRID Segment set ID at base of soil to be excavated.

SOILID Segment set ID at face of rest of the soil domain.

SPR Include the slave side in the *DATABASE_NCFORC and the *DATABASE_BINARY_INTFOR interface force files:

EQ.1: slave side forces included.

MPR Include the master side in the *DATABASE_NCFORC and the *DATABASE_BINARY_INTFOR interface force files:

EQ.1: master side forces included.

Remarks:

1.The motions on the specified segment set or node set is recorded in a binary file named **gmbin** by default, but a different filename may be chosen using option GMOUT on the command line (see INTRODUCTION, Execution Syntax).

2.The output interval for the motions may be specified using the parameter GMDT on the *CONTROL_OUTPUT card, with the default value being 1/10-th of the output interval for D3PLOT states.

*INTERFACE_SSI_STATIC_*{OPTION}*_ID

Purpose: This card creates a tied-contact soil-structure interface in order to record the static reactions at the base of the structure, which are to be used in a subsequent dynamic analysis of the soil-structure system subjected to earthquake excitation. This card is intended to be used with the initial static analysis of the structure subjected to gravity loads.

Available options are:

<BLANK>

OFFSET

CONSTRAINED_OFFSET

Card 1	1	2	3	4	5	6	7	8
Variable	ID	HEADING						
Type	I	A70						

Card 2	1	2	3	4	5	6	7	8
Variable	STRID	SOILID	SPR	MPR				
Type	I	I	I	I				
Default	none	none	0	0				

VARIABLE	DESCRIPTION
ID	Soil-structure interface ID. This is required and must be unique amongst all the contact interface IDs in the model.
HEADING	A descriptor for the given ID.
STRID	Segment set ID of base of structure at soil-structure interface.
SOILID	Segment set ID of soil at soil-structure interface.

VARIABLE	DESCRIPTION
SPR	Include the slave side in the *DATABASE_NCFORC and the *DATABASE_BINARY_INTFOR interface force files: EQ.1: slave side forces included.
MPR	Include the master side in the *DATABASE_NCFORC and the *DATABASE_BINARY_INTFOR interface force files: EQ.1: master side forces included.

Remarks:

See *INTERFACE_SSI_ID. The ID used for a particular interface in the static analysis must also be used for the same interface identified using *INTERFACE_SSI_ID during dynamic analysis.

*INTERFACE_WELDLINE_DEVELOPMENT

Purpose: This keyword causes LS-DYNA to run a weld line development calculation instead of a finite element calculation. The input for this feature consists of (1) the formed blank from a completed metal forming simulation, (2) the corresponding initial blank, and (3) if the desired weld curve on the formed blank is provided, the *INTERFACE_WELDLINE_DEVELOPMENT method creates a weld curve on the initial blank; if the initial weld curve on the initial blank is provided, this method creates a weld curve on the final blank. Outputs also include nodes of any element edges that intersect the weld curve on the initial (affectednd_i.ibo) and final blanks (affectednd_f.ibo).

Three additional keywords must be used together (and exclusively) with this keyword. They are: *INITIAL_BLANK, *FINAL_PART, and *WELDING_CURVE.

> **NOTE:** When this card is present LS-DYNA *does not* proceed to the finite element simulation.

Card set for *INTERFACE_WELDLINE_DEVELOPMENT.

Development Parameter Card.

Card 1	1	2	3	4	5	6	7	8
Variable	IOPTION							
Type	I							
Default	1							

Initial Blank Card. Following keyword *INITIAL_BLANK:

Card 2	1	2	3	4	5	6	7	8
Variable	FILENAME1							
Type	A80							
Default	none							

Final Part Card. Following keyword *FINAL_PART:

Card 3	1	2	3	4	5	6	7	8
Variable	FILENAME2							
Type	A80							
Default	none							

Welding Curve Card. Following keyword *WELDING_CURVE:

Card 4	1	2	3	4	5	6	7	8
Variable	FILENAME3							
Type	A80							
Default	none							

VARIABLE	DESCRIPTION
IOPTION	Welding curve development options: EQ.1: Calculate initial weld curve from final (given) weld curve, with output file name weldline.ibo, which will be on the initial blank mesh. EQ.-1:Calculate final weld curve from initial weld curve, with output file name weldline_f.ibo, which will be on the formed blank mesh.
FILENAME1	Initial blank file name in keyword format.
FILENAME2	Final formed blank file name in keyword format.
FILENAME3	File name of the weld curve, when IOPTION: EQ.1: Final (target) welding curve file name; the curve is defined using *DEFINE_CURVE_TRIM_3D. EQ.-1:Initial weld curve file name; the curve is defined using *DEFINE_CURVE_TRIM_3D.

General Remarks:

For metal forming of tailor welded blanks, an initial straight weld line could become a curve on the formed part. The amounts of deviation of the formed weld curve from its initial line depend on the part shape and forming conditions. Sometimes the formed weld curve is not desirable; so a correction to the initial weld curve is needed. Or, given an initial welding curve, without performing another simulation, what would the final weld curve be like? This keyword addresses these concerns.

Mesh with adaptivity for the initial blank and final part is supported.

Final (formed) Weld Curve:

The final formed weld curve should be projected onto the final blank mesh if it does not exactly lie on the mesh surface. This can be done with *LS-PrePost4.2* via the menu option *GeoTol* → *Project* → *Project*, select *Closest Projection*, select *Project to Elements*, then define the destination mesh and source curves, and hit *Apply*. Sometimes the target curve may need enough points before projection; the points may be added via menu option *Curve* → *Spline* → *Method* (Respace) → *by number*. To write the curve out in *DEFINE_CURVE_TRIM_3D format, use *Curve* → *Convert* → *Method* (To DEFINE_CURVE_TRIM) → *To Key*, then write out the keyword using *FILE* → *Save Keyword*.

Computed Weld Curve (and IGES):

Computed weld curves are written with *DEFINE_CURVE_TRIM_3D keyword into a file called weldline.ibo (or, weldline_f.ibo, depending on the IOPTION). The format of this file follows the keyword's specification. *LS-PrePost4.0* can convert the computed curve to IGES, see procedures in manual pages under *INTERFACE_BLANKSIZE_DEVELOPMENT. After hitting *Apply*, the curves will show up in the graphics window, and *File* → *Save as* → *Save Geom as* can be used to write the curves out in IGES format.

Example:

As shown in Figure 25-41, given the initial (initialblank.k) and final blank (finalblank.k) configuration and a final formed weld curve (finalweldingcurve.k), the following input calculates a new initial weld curve on the initial blank. In this case, the final weld curve is specified as straight in the drawn panel.

```
*KEYWORD
*INTERFACE_WELDLINE_DEVELOPMENT
$OPTION
1
*INITIAL_BLANK
```

```
initialblank.k
*FINAL_PART
finalblank.k
*WELDING_CURVE
finalweldingcurve.k
*END
```

The output is the initial weld curve in the file weldline.ibo, Figure 25-41. Nodes of element edges that intersect the initial weld curve are output in affectednd_i.ibo; while nodes of element edges that intersect the final formed weld curve are output in affectednd_f.ibo, Figure 25-42.

To verify the predicted weld curve, initial blank can be re-meshed according to the curve. A draw simulation can be performed again to confirm the final weld curve as straight, Figure 25-43.

Likewise, if given an initial weld curve (initialweldingcurve.k) and a final weld curve (weldline_f.ibo) can be calculated with the keyword inputs below:

```
*KEYWORD
*INTERFACE_WELDLINE_DEVELOPMENT
$OPTION
-1
*INITIAL_BLANK
initialblank.k
*FINAL_PART
finalblank.k
*WELDING_CURVE
initialweldingcurve.k
*END
```

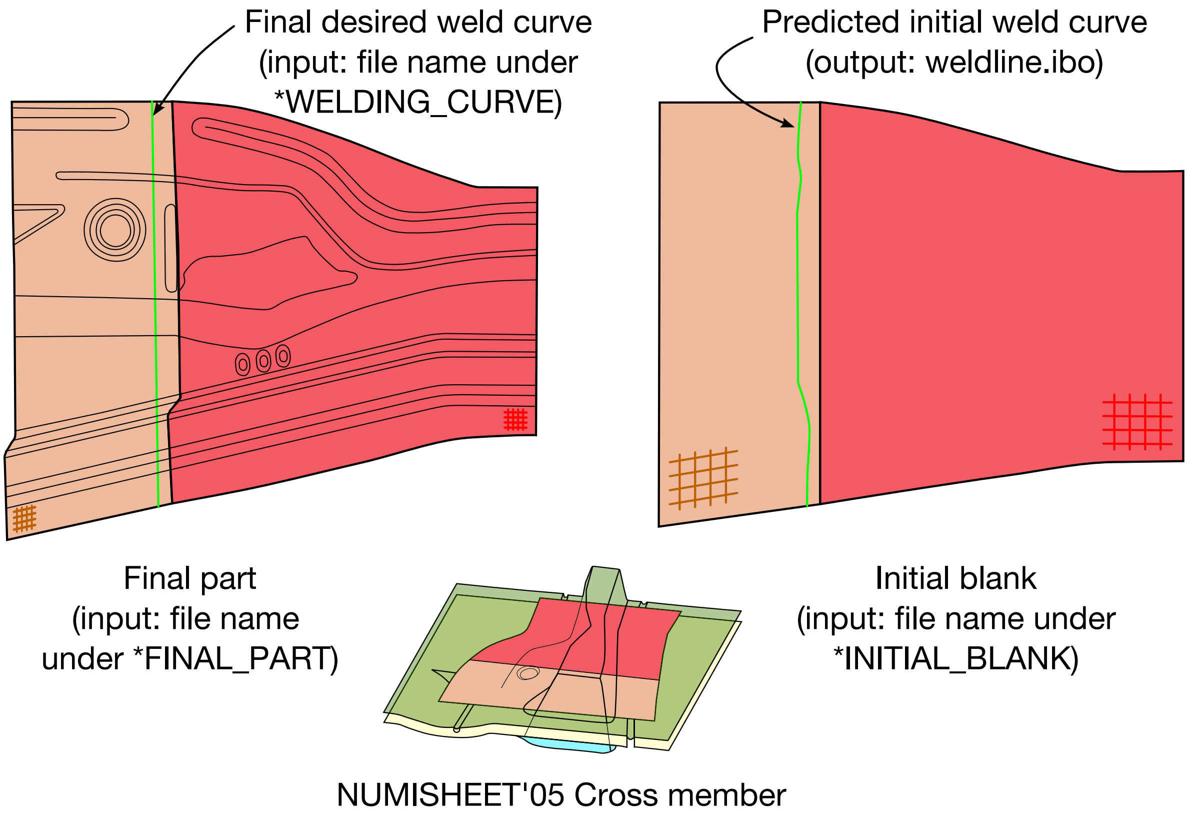


Figure 25-41. An example for a welding curve development.

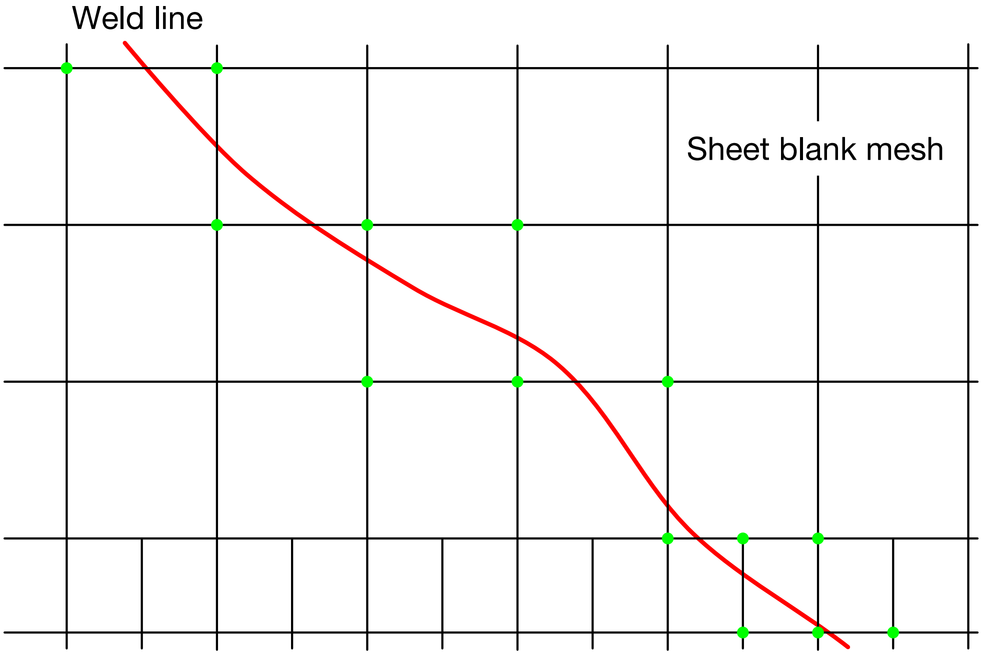


Figure 25-42. Nodes (in green) of element edges that intersect the weld curve are output in affectednd_i.ibo and affectednd_f.ibo, for initial and deformed mesh, respectively.

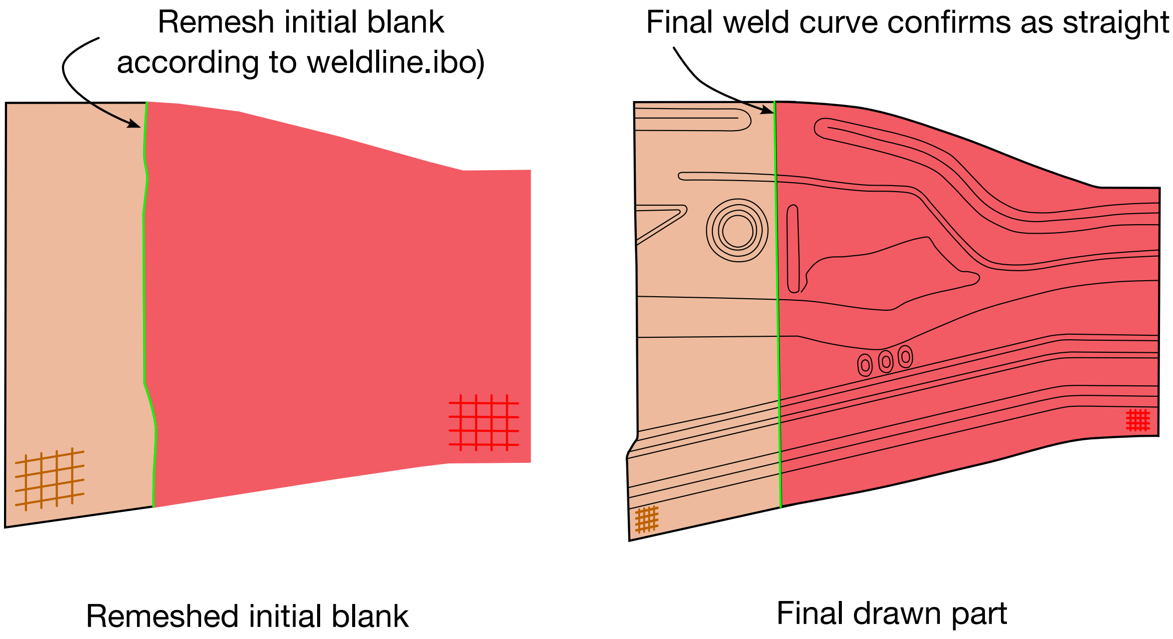

Remesh initial blank
according to weldline.ibo)
Final weld curve confirms as straight
Remeshed initial blank
Final drawn part

Figure 25-43. Verification run

Revision information:

1. IOPTION of "1": Revision 105189 in both double precision versions of SMP and MPP.

2. IOPTION of "-1": Revision 105727.

3. Output of nodes of element edges that intersect the weld curve: Revision 105727.

4. Later revisions may include improvements.

*KEYWORD

KEYWORD_{OPTION} {memory} {memory2 = j} {NCPU = n}*

Available options include:

<BLANK>

ID

JOBID

Purpose: The keyword, *KEYWORD, flags LS-DYNA that the input deck is a keyword deck rather than the structured format, which has a strictly defined format. This must be the first card in the input file. Alternatively, by typing "keyword" on the execution line, keyword input formats are assumed and this beginning "*KEYWORD" line is not required.

There are 3 optional parameters that can be specified on the *KEYWORD line. If a number *{memory}* is specified, it defines the memory size in units of words to be allocated. For MPP, if the parameter {memory2 = j} is given, it defines the memory allocation for rest of the MPP ranks. Note that if the memory size is specified on the execution line, it will override the memory size specified on the *KEYWORD line.

If the parameter {NCPU = *n*} is specified it defines the number of CPUs "*n*" to be used during the analysis. This only applies to the Shared Memory Parallel (SMP) version of LS-DYNA. For the Distributed Memory Version (MPP), the number of CPUs is always defined with the "mpirun" command. Defining the number of CPUs on the execution line overrides what is specified on the *KEYWORD line and both override the number of CPUs specified by *CONTROL_PARALLEL. An example of the {memory} and {NCPU = *n*} options would be as follows:

```
*KEYWORD 12000000 NCPU=2
```

This *KEYWORD command is requesting 12 million words of memory and 2 CPUs to be used for the analysis with the consistency flag (see CONST in *CONTROL_PARALLEL) turned off. To run with the consistency flag turned on (recommended), set NCPU to a negative value, e.g., "NCPU = -2" runs with 2 CPUs with the consistency flag turned on.

The ID and JOBID command line options are available to add a prefix to all output and scratch file names, i.e., not the input filenames. This allows multiple simulations in a directory since a different prefix prevents files from being overwritten. If the ID option is used, the prefix is constructed of three user specified strings separated by "_f" characters.

ID Card. Additional Card if the ID option is active.

Card 1	1	2	3	4	5	6	7	8
Variable	PROJECT		NUM		STAGE			
Type	A		A		A			
Default	none		none		none			

VARIABLE	DESCRIPTION
PROJECT	First part of the output file name prefix.
NUM	Second part of the output file name prefix.
STAGE	Third part of the output file name prefix.

By using the ID option of *KEYWORD, an output file name prefix may be specified as a combination of the variables PROJECT, NUM and STAGE as defined on Card 1 above. For example, if these variables were set literally to "PROJECT", "NUM", and "STAGE", the first d3plot would be named:

```
PROJECT_NUM_STAGE.d3plot
```

Alternatively, an output file name prefix can be assigned by including "jobid=" on the execution line. For example,

```
lsdyna i=input.k jobid=PROJECT_NUM_STAGE
```

A third way to define an output file name prefix is by using the JOBID option of the *KEYWORD command, in which case Card 1 is defined as shown below and the variable JBID acts as the output prefix.

JOBID Card. Additional card if the JOBID option is active.

Card 1	1	2	3	4	5	6	7	8
Variable	JBID							
Type	A							
Default	none							

*LOAD

The keyword *LOAD provides a way of defining applied forces. The keyword control cards in this section are defined in alphabetical order:

LOAD_ALE_CONVECTION_*{OPTION}

LOAD_BEAM_*OPTION

*LOAD_BLAST

*LOAD_BLAST_ENHANCED

*LOAD_BLAST_SEGMENT

*LOAD_BLAST_SEGMENT_SET

LOAD_BODY_*OPTION

*LOAD_BODY_GENERALIZED

*LOAD_BODY_POROUS

*LOAD_BRODE

*LOAD_DENSITY_DEPTH

*LOAD_ERODING_PART_SET

*LOAD_GRAVITY_PART

*LOAD_HEAT_CONTROLLER

LOAD_HEAT_GENERATION_*OPTION

*LOAD_MASK

*LOAD_MOTION_NODE

*LOAD_MOVING_PRESSURE

LOAD_NODE_*OPTION

*LOAD_REMOVE_PART

*LOAD_RIGID_BODY

LOAD_SEGMENT_*{OPTION}

*LOAD_SEGMENT_CONTACT_MASK

*LOAD_SEGMENT_FILE

*LOAD_SEGMENT_FSILNK

LOAD_SEGMENT_NONUNIFORM_*{OPTION}

LOAD_SEGMENT_SET_*{OPTION}

*LOAD_SEGMENT_SET_ANGLE

LOAD_SEGMENT_SET_NONUNIFORM_*{OPTION}

*LOAD_SEISMIC_SSI_*OPTION1*_*{OPTION2}*

*LOAD_SHELL_*{OPTION1}*_*{OPTION2}*

*LOAD_SPCFORC

*LOAD_SSA

*LOAD_STEADY_STATE_ROLLING

*LOAD_STIFFEN_PART

*LOAD_SUPERPLASTIC_FORMING

LOAD_SURFACE_STRESS_*OPTION

LOAD_THERMAL_*OPTION

*LOAD_THERMAL_CONSTANT

*LOAD_THERMAL_CONSTANT_ELEMENT

*LOAD_THERMAL_CONSTANT_NODE

*LOAD_THERMAL_D3PLOT

*LOAD_THERMAL_LOAD_CURVE

*LOAD_THERMAL_TOPAZ

*LOAD_THERMAL_VARIABLE

LOAD_THERMAL_VARIABLE_BEAM_*{OPTION}

LOAD_THERMAL_VARIABLE_ELEMENT_*{OPTION}

*LOAD_THERMAL_VARIABLE_NODE

LOAD_THERMAL_VARIABLE_SHELL_*{OPTION}

*LOAD_VOLUME_LOSS

LOAD_ALE_CONVECTION_*{OPTION}

Purpose: This card is used to define the convection thermal energy transfer from a hot ALE fluid to the surrounding Lagrangian structure (remark 1). It is associated with a corresponding coupling card defining the interaction between the ALE fluid and the Lagrangian structure. It is only used when thermal energy transfer from the ALE fluid to the surrounding Lagrangian structure is significant. This is designed specifically for airbag deployment application where the heat transfer from the inflator gas to the inflator compartment can significantly affect the inflation potential of the inflator.

Available options include:

<BLANK>

ID

To define an ID number for each convection heat transfer computation in an optional card preceding all other cards for this command. This ID number can be used to output the part temperature and temperature change as functions of time in the *DATABASE_-FSI card. To do this, set the CONVID parameter in the *DATABASE_FSI card equal to this ID.

ID Card. Additional card for ID keyword option.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	TITLE						
Type	I	A70						
Default	none	none						

Include as many cards as necessary. This input terminates at the next keyword (“*”) card.

Card 2	1	2	3	4	5	6	7	8
Variable	LAGPID	LAGT	LAGCP	H	LAGMAS			
Type	I	F	F	F	F			
Default	none	none	none	none	none			

VARIABLE	DESCRIPTION
LAGPID	Lagrangian PID (slave PID) from a corresponding coupling card which receives the thermal energy in the convection heat transfer.
LAGT	Initial temperature of this Lagrangian slave part.
LAGCP	Constant-pressure heat capacity of this Lagrangian slave part. It has a per-mass unit (for example, J/[Kg*K]).
H	Convection heat transfer coefficient on this Lagrangian slave part surface. It is the amount of energy (J) transferred per unit area, per time, and per temperature difference. For example, its units may be J/[m^2*s*K]
LAGMAS	The mass of the Lagrangian slave part receiving the thermal energy. This is in absolute mass unit (for example, Kg).

Remarks:

1. The only application of this card so far has been for the transfer of thermal energy from the ALE hot inflator gas to the surrounding Lagrangian structure (inflator canister and airbag-containing compartment) in an airbag deployment model.

2. The heat transferred is taken out of the inflator gas thermal energy thus reducing its inflating potential.

3. This is not a precise heat transfer modeling attempt. It is simply one mechanism for taking out excessive energy from the inflating potential of the hot inflator gas.

4. The heat transfer formulation may roughly be represented as following. Some representative units are shown just for clarity.

$$[\dot{Q}] = [\mathrm{H} \times A \times \Delta T] = \left(\frac{[\mathrm{E}]}{[L]^2[\mathrm{t}][\mathrm{T}]}\right) \times [\mathrm{L}]^2 \times [\mathrm{T}] = [\mathrm{Power}]$$

$$[\dot{Q}] = \left[\dot{M}C_p\left(T_{\mathrm{Lag\ New}} - T_{\mathrm{Lag\ Orig}}\right)\right] = \left(\frac{[\mathrm{M}]}{[t]}\right) \times \left(\frac{[\mathrm{E}]}{[\mathrm{M}][\mathrm{T}]}\right) \times [\mathrm{T}] = \frac{[\mathrm{E}]}{[\mathrm{t}]}$$

LOAD_BEAM_*OPTION

Available options include:

ELEMENT

SET

Purpose: Apply the distributed traction load along any local axis of beam or a set of beams. The local axes are defined in Figure 27-1, see also *ELEMENT_BEAM.

Beam Cards. Include as many as necessary. This input stops at the next keyword ("*") card.

Card	1	2	3	4	5	6	7	8
Variable	EID/ESID	DAL	LCID	SF				
Type	I	I	I	F				
Default	none	none	none	1.				

VARIABLE	DESCRIPTION
EID/ESID	Beam ID (EID) or beam set ID (ESID), see *ELEMENT_BEAM or *SET_BEAM.

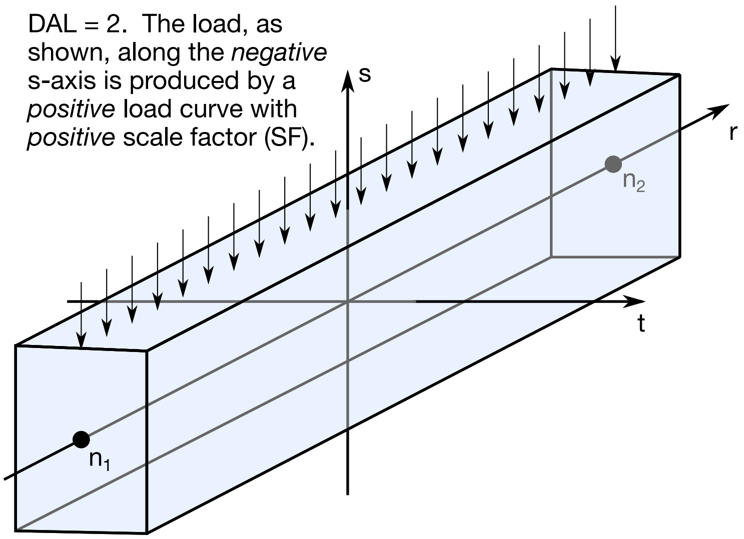

DAL = 2. The load, as shown, along the *negative* s-axis is produced by a *positive* load curve with *positive* scale factor (SF).
s
r
n_2
t
n_1

Figure 27-1. Applied traction loads are given in force per unit length. The s and t directions are defined on the *ELEMENT_BEAM keyword.

VARIABLE **DESCRIPTION**

DAL

Direction of applied load:

EQ.1: parallel to r-axis of beam,

EQ.2: parallel to s-axis of beam,

EQ.3: parallel to t-axis of beam.

LCID

Load curve ID (see *DEFINE_CURVE) or function ID (see *DEFINE_FUNCTION).

SF

Load curve scale factor. This is for a simple modification of the function values of the load curve.

Remark:

1.The function defined by LCID has 7 arguments: time, the 3 current coordinates, and the 3 reference coordinates. For example, using *DEFINE_FUNCTION,

f(t,x,y,z,x0,y0,z0)= -10.*sqrt ((x-x0)*(x-x0)+(y-y0)*(y-y0)+(z-z0)*(z-z0)).

applies a force proportional to the distance from the initial coordinates.

*LOAD_BLAST

Purpose: Define an airblast function for the application of pressure loads from the detonation of conventional explosives. The implementation is based on a report by Randers-Pehrson and Bannister [1997] where it is mentioned that this model is adequate for use in engineering studies of vehicle responses due to the blast from land mines. This option determines the pressure values when used in conjunction with the keywords: *LOAD_SEGMENT, *LOAD_SEGMENT_SET, or *LOAD_SHELL.

Card 1	1	2	3	4	5	6	7	8
Variable	WGT	XBO	YBO	ZBO	TBO	IUNIT	ISURF	
Type	F	F	F	F	F	I	I	
Default	none	0.0	0.0	0.0	0.0	2	2	

Card 2	1	2	3	4	5	6	7	8
Variable	CFM	CFL	CFT	CFP	DEATH			
Type	F	F	F	F	F			
Default	0.0	0.0	0.0	0.0	0.0			

VARIABLE	DESCRIPTION
WGT	Equivalent mass of TNT.
XBO	x-coordinate of point of explosion.
YBO	y-coordinate of point of explosion.
ZBO	z-coordinate of point of explosion.
TBO	Time-zero of explosion.

VARIABLE	DESCRIPTION
IUNIT	Unit conversion flag. EQ.1: feet, pound-mass, seconds, psi EQ.2: meters, kilograms, seconds, Pascals (default) EQ.3: inch, dozens of slugs, seconds, psi EQ.4: centimeters, grams, microseconds, Megabars EQ.5: user conversions will be supplied (see Card 2)
ISURF	Type of burst. EQ.1: surface burst - is located on or very near the ground surface (see Remark 5) EQ.2: air burst - spherical charge (default)
CFM	Conversion factor - pounds per LS-DYNA mass unit.
CFL	Conversion factor - feet per LS-DYNA length units.
CFT	Conversion factor - milliseconds per LS-DYNA time unit.
CFP	Conversion factor - psi per LS-DYNA pressure unit
DEATH	Death time. Blast pressures are deactivated at this time.

Remarks:

1. A minimum of two load curves, even if unreferenced, must be present in the model.

2. Segment normals should point away from the structure and nominally toward the charge.

3. Several methods can be used to approximate the equivalent mass of TNT for a given explosive. The simplest involves scaling the mass by the ratio of the Chapman-Jouguet detonation velocities given the by relationship.

$$M_{\mathrm{TNT}} = M \frac{DCJ^2}{DCJ_{\mathrm{TNT}}^2}$$

where M_{TNT} is the equivalent TNT mass and DCJ_{TNT} is the Chapman-Jouguet detonation velocity of TNT. M and DCJ are, respectively, the mass and C-J velocity of the explosive under consideration. "Standard" TNT is considered to be cast with a density of 1.57 gm/cm^3 and DCJ_{TNT} = 0.693 cm/microsecond.

4. The empirical equations underlying the spherical air burst are valid for the range of scaled distance 0.37 ft/lbm$^{1/3}$ < Z < 100 ft/lbm$^{1/3}$ (0.147 m/kg$^{1/3}$ < Z < 40 m/kg$^{1/3}$) where $Z = R/M^{1/3}$, R is the distance from the charge center to the target and M is the TNT equivalent mass of the charge.. The range of applicability for the hemispherical surface burst is 0.45 ft/lbm$^{1/3}$ < Z < 100 ft/lbm$^{1/3}$ (0.178 m/kg$^{1/3}$ < Z < 40 m/kg$^{1/3}$).

5. When a charge is located on or very near the the ground surface it is considered to be a surface burst. Under this circumstance the initial blast wave is immediately reflected and reinforced by the nearly unyielding ground to produce a reflected wave that moves out hemispherically from the point of burst. This reflected wave merged with the initial incident wave produces overpressures which are greater than those produced by the initial wave alone. In LS-DYNA this wave moves out spherically from the burst point so no distinction of the ground orientation is made. Target points equidistant from the burst point are loaded identically with the surface burst option.

*LOAD_BLAST_ENHANCED

Purpose: Define an airblast function for the application of pressure loads due the detonation of a conventional explosive. While similar to *LOAD_BLAST this feature includes enhancements for treating ground-reflected waves, moving warheads and multiple blast sources. The loads are applied to facets defined with the keyword *LOAD_BLAST_SEGMENT. A database containing blast pressure history is also available (see *DATABASE_BINARY_BLSTFOR).

Card Sets. Include as many sets of the following cards as necessary. This input terminates at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	BID	M	XBO	YBO	ZBO	TBO	UNIT	BLAST
Type	I	F	F	F	F	F	I	I
Default	none	0.0	0.0	0.0	0.0	0.0	2	2
Remarks		1				3	4	7

Card 2	1	2	3	4	5	6	7	8
Variable	CFM	CFL	CFT	CFP	NIDBO	DEATH	NEGPHS	
Type	F	F	F	F	I	F	I	
Default	0.0	0.0	0.0	0.0	none	1.e+20	0	

VARIABLE	DESCRIPTION
BID	Blast ID. A unique number must be defined for each blast source (charge). Multiple charges may be defined, however, interaction of the waves in air is not considered.
M	Equivalent mass of TNT (see Remark 1).
XBO	x-coordinate of charge center.

VARIABLE	DESCRIPTION
YBO	y-coordinate of charge center.
ZBO	z-coordinate of charge center.
TBO	Time of detonation. See Remark 3.
UNIT	Unit conversion flag. See Remark 4. EQ.1: pound-mass, foot,second, psi EQ.2: kilogram, meter,second, Pascal (default) EQ.3: dozen slugs (i.e., lbf-s2/in), inch, second, psi EQ.4: centimeters, grams, microseconds, Megabars EQ.5: user conversions will be supplied (see Card 2) EQ.6: kilogram, millimeter, millisecond, GPa EQ.7: metric ton, millimeter, second, MPa EQ.8: gram, millimeter, millisecond, MPa
BLAST	Type of blast source. EQ.1: hemispherical surface burst – charge is located on or very near the ground surface (see Remark 7) EQ.2: spherical air burst (default) – no amplification of the initial shock wave due to interaction with the ground surface EQ.3: air burst – moving non-spherical warhead EQ.4: air burst with ground reflection – initial shock wave impinges on the ground surface and is reinforced by the reflected wave to produce a Mach front (see Remark 8).
CFM	Conversion factor - pounds per LS-DYNA mass unit.
CFL	Conversion factor - feet per LS-DYNA length units.
CFT	Conversion factor - milliseconds per LS-DYNA time unit.
CFP	Conversion factor - psi per LS-DYNA pressure unit.
NIDBO	Optional node ID representing the charge center. If non-zero then XBO, YBO and XBO are ignored.
DEATH	Death time. Blast pressures are deactivated at this time.

VARIABLE	DESCRIPTION
NEGPHS	Treatment of negative phase. EQ.0: negative phase dictated by the Friedlander equation. EQ.1: negative phase ignored as in ConWep.

Moving non-spherical warhead Card. Additional Card for BLAST = 3.

Card 3	1	2	3	4	5	6	7	8
Variable	VEL	TEMP	RATIO	VID				
Type	F	F	F	F				
Default	0.0	70.0	1.0	none				

VARIABLE	DESCRIPTION
VEL	Speed of warhead.
TEMP	Ambient air temperature, Fahrenheit.
RATIO	Aspect ratio of the non- spheroidal blast front. This is the longitudinal axis radius divided by the lateral axis radius. Shaped charge and EFP warheads typically have significant lateral blast resembling an oblate spheroid with RATIO < 1. Cylindrically cased explosives produce more blast in the longitudinal direction so RATIO > 1, rendering a prolate spheroid blast front, is more appropriate..
VID	Vector ID representing the longitudinal axis of the warhead (see *DEFINE_VECTOR). This vector is parallel to the velocity vector when a non-zero velocity VEL is defined.

Spherical air burst with ground reflect Card. Additional card for BLAST = 4.

Card 3	1	2	3	4	5	6	7	8
Variable	GNID	GVID						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
GNID	ID of node residing on the ground surface.
GVID	ID of vector representing the vertically upward direction, i.e., normal to the ground surface (see *DEFINE_VECTOR).

Remarks:

1. Several methods can be used to approximate the equivalent mass of TNT for a given explosive. The simplest involves scaling the mass by the ratio of the Chapman-Jouguet detonation velocities given the by relationship.

$$M_{\mathrm{TNT}} = M\frac{DCJ^2}{DCJ_{\mathrm{TNT}}^2}$$

where M_{TNT} is the equivalent TNT mass and DCJ_{TNT} is the Chapman-Jouguet detonation velocity of TNT. M and DCJ are, respectively, the mass and C-J velocity of the explosive under consideration. “Standard” TNT is considered to be cast with a density of 1.57 gm/cm^3 and DCJ_{TNT} = 0.693 cm/microsecond.

2. Segment normals should point away from the structure and nominally toward the charge unless it is the analyst’s intent to apply pressure to the leeward side of a structure. The angle of incidence is zero when the segment normal points directly at the charge. Only incident pressure is applied to a segment when the angle of incidence is greater than 90 degrees.

3. The blast time offset TBO can be used to adjust the detonation time of the charge relative to the start time of the LS-DYNA simulation. The detonation time is delayed when TBO is positive. More commonly, TBO is set negative so that the detonation occurs before time-zero of the LS-DYNA calculation. In this manner, computation time is not wasted while “waiting” for the blast wave to reach the structure. The following message, written to the messag and d3hsp files as well as the screen, is useful in setting TBO.

```
    Blast wave reaches structure at 2.7832E-01 milliseconds
```

As an example, one might run LS-DYNA for one integration cycle and record the arrival time listed in the message above. Then TBO is set to a negative number slightly smaller in magnitude than the reported arrival time, for example TBO = -0.275 milliseconds. Under this circumstance the blast wave would reach the structure shortly after the start of the simulation.

4. Computation of blast pressure relies on an underlying method which uses base units of lbm-foot-millisecond-psi; note that this internal unit system is inconsistent. Calculations require that the system of units in which the LS-DYNA model is constructed must be converted to this internal set of units. Predefined and user-defined unit conversion factors are available (see the parameter UNIT) and these unit conversion factors are echoed back in the d3hsp file. Below is an example of user-defined (UNIT = 5) conversion factors for the gm-mm-millisecond-Mpa unit system.

$$1 = \left[\frac{\text{CFM} \times \text{lb}}{\text{LS-DYNA mass unit}}\right] = \left[\underbrace{2.2 \times 10^{-3}}_{= \text{CFM}} \frac{\text{lbm}}{\text{gm}}\right]$$

$$1 = \left[\frac{\text{CFL} \times \text{ft}}{\text{LS-DYNA length unit}}\right] = \left[3.28 \times 10^{-3} \frac{\text{ft}}{\text{mm}}\right]$$

$$1 = \left[\frac{\text{CFT} \times \text{ms}}{\text{LS-DYNA time unit}}\right] = \left[1.0 \frac{\text{ms}}{\text{ms}}\right]$$

$$1 = \left[\frac{\text{CFP} \times \text{psi}}{\text{LS-DYNA pressure unit}}\right] = \left[145.0 \frac{\text{psi}}{\text{MPa}}\right]$$

5. The empirical equations underlying the spherical air burst are valid for the range of scaled distance 0.37 ft/lbm$^{1/3}$ < Z < 100 ft/lbm$^{1/3}$ (0.147 m/kg$^{1/3}$ < Z < 40 m/kg$^{1/3}$) where Z = R/M$^{1/3}$, R is the distance from the charge center to the target and M is the TNT equivalent mass of the charge. The range of applicability for the hemispherical surface burst is 0.45 ft/lbm$^{1/3}$ < Z < 100 ft/lbm$^{1/3}$ (0.178 m/kg$^{1/3}$ < Z < 40 m/kg$^{1/3}$).

6. Blast loads can be used in 2D axisymmetric analyses. Repeat the second node for the third and fourth nodes of the segment definition in *LOAD_BLAST_SEGMENT and *LOAD_BLAST_SEGMENT_SET.

7. When a charge is located on or very near the the ground surface it is considered to be a surface burst. Under this circumstance the initial blast wave is immediately reflected and reinforced by the nearly unyielding ground to produce a reflected hemispherical wave that moves out from the point of burst. This reflected wave merged with the initial incident wave produces overpressures which are greater than those produced by the initial wave alone. In LS-DYNA this wave moves out spherically from the burst point so no distinction of the

ground orientation is made. Target points equidistant from the burst point are loaded identically with the surface burst option.

8. The empirical equations underlying the spherical air burst with ground reflection (BLAST = 4) are valid for the range of scaled height of burst 1.0 ft/lbm$^{1/3}$ < Hc//M$^{1/3}$ < 7.0 ft/lbm$^{1/3}$ (0.397 m/kg$^{1/3}$ < Z < 2.78 m/kg$^{1/3}$) where Hc is the height of the charge center above the ground and M is the TNT equivalent mass of the charge.

*LOAD_BLAST_SEGMENT

Purpose: Apply blast pressure loading over a triangular or quadrilateral segment for 3D geometry or line segment for 2D geometry (see *LOAD_BLAST_ENHANCED).

Segment Cards. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	BID	N1	N2	N3	N4	ALEPID	SFNRB	SCALEP
Type	I	I	I	I	I	I	F	F
Default	none	none	none	none	none	none	0.	1.

VARIABLE	DESCRIPTION
BID	Blast source ID (see *LOAD_BLAST_ENHANCED).
N1	Node ID.
N2	Node ID.
N3	Node ID. For line segments on two-dimensional geometries set N3 = N2.
N4	Node ID. For line segments on two-dimensional geometries set N4 = N3 = N2 or for triangular segments in three diemensions set N4 = N3.
ALEPID	Part ID of ALE ambient part underlying this segment to be loaded by this blast (see *PART and *SECTION_SOLID, AET = 5). This applies only when the blast load is coupled to an ALE air domain.
SFNRB	Scale factor for the ambient element non-reflecting boundary condition. Shocks waves reflected back to the ambient elements can be attenuated with this feature. A value of 1.0 works well for most situations. The feature is disabled when a value of zero is specified
SCALEP	Pressure scale factor.

*LOAD_BLAST_SEGMENT_SET

Purpose: Apply blast pressure loading over each segment in a segment set (see *LOAD_BLAST_ENHANCED).

Segment Set Cards. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	BID	SSID	ALEPID	SFNRB	SCALEP			
Type	I	I	I	F	F			
Default	none	none	none	0.	1.			

VARIABLE	DESCRIPTION
BID	Blast source ID (see *LOAD_BLAST_ENHANCED).
SSID	Segment set ID (see *SET_SEGMENT).
ALEPID	Part ID of ALE ambient part underlying this segment to be loaded by this blast (see *PART and *SECTION_SOLID, AET = 5). This applies only when the blast load is coupled to an ALE air domain.
SFNRB	Scale factor for the ambient element non-reflecting boundary condition. Shocks waves reflected back to the ambient elements can be attenuated with this feature. A value of 1.0 works well for most situations.
SCALEP	Pressure scale factor.

Remarks:

1. Triangular segments are defined by setting N4 = N3.

2. Line segments for two-dimensional geometries are defined by setting N4 = N3 = N2.

LOAD_BODY_*OPTION

Available options include for base accelerations:

X

Y

Z

for angular velocities:

RX

RY

RZ

for loading in any direction, specified by vector components:

VECTOR

and to specify a part set:

PARTS

Purpose: Define body force loads due to a prescribed base acceleration or angular velocity using global axes directions. This option applies nodal forces only: it cannot be used to prescribe translational or rotational motion. These body forces do not take into account non-physical mass added via mass scaling; see *CONTROL_TIMESTEP.

> **NOTE:** This data applies to all nodes in the complete problem unless a part subset is specified via the *LOAD_BODY_PARTS keyword.

If a part subset with *LOAD_BODY_PARTS then all nodal points belonging to the subset will have body forces applied.

> **NOTE:** Only one *LOAD_BODY_PARTS card is permitted per deck. To specify, for instance, one body load on one part and another body load on another part use *LOAD_BODY_GENERALIZED instead.

For options X, Y, Z, RX, RY, RZ and VECTOR.

Card 1	1	2	3	4	5	6	7	8
Variable	LCID	SF	LCIDDR	XC	YC	ZC	CID	
Type	I	F	I	F	F	F	I	
Default	none	1.	0	0.	0.	0.	0	

For option PARTS.

Card 1	1	2	3	4	5	6	7	8
Variable	PSID							
Type	I							
Default	none							

For option VECTOR.

Card 2	1	2	3	4	5	6	7	8
Variable	V1	V2	V3					
Type	F	F	F					
Default	0.0	0.0	0.0					

VARIABLE	DESCRIPTION
LCID	Load curve ID, see *DEFINE_CURVE.
SF	Load curve scale factor

VARIABLE	DESCRIPTION
LCIDDR	Load curve ID for dynamic relaxation phase (optional). This is needed when dynamic relaxation is defined and a different load curve to LCID is required during the dynamic relaxation phase. Note if LCID is undefined then no body load will be applied during dynamic relaxation regardless of the value LCIDDR. See *CONTROL_DYNAMIC_RELAXATION
XC	x-center of rotation, define for angular velocities.
YC	y-center of rotation, define for angular velocities.
ZC	z-center of rotation, define for angular velocities.
CID	Coordinate system ID to define acceleration in local coordinate system. The accelerations (LCID) are with respect to CID. EQ.0: global
PSID	Part set ID.
V1, V2, V3	Vector components of vector **V**.

General remarks:

Translational base accelerations allow body force loads to be imposed on a structure. Conceptually, base acceleration may be thought of as accelerating the coordinate system in the direction specified, and, thus, the inertial loads acting on the model are of opposite sign. For example, if a cylinder were fixed to the y-z plane and extended in the positive x-direction, then a positive x-direction base acceleration would tend to shorten the cylinder, i.e., create forces acting in the negative x-direction.

Base accelerations are frequently used to impose gravitational loads during dynamic relaxation to initialize the stresses and displacements. During the analysis, in this latter case, the body forces loads are held constant to simulate gravitational loads. When imposing loads during dynamic relaxation, it is recommended that the load curve slowly ramp up to avoid the excitation of a high frequency response.

Body force loads due to the angular velocity about an axis are calculated with respect to the deformed configuration and act radially outward from the axis of rotation. Torsional effects which arise from changes in angular velocity are neglected with this option. The angular velocity is assumed to have the units of radians per unit time.

The body force density is given at a point **P** of the body by:

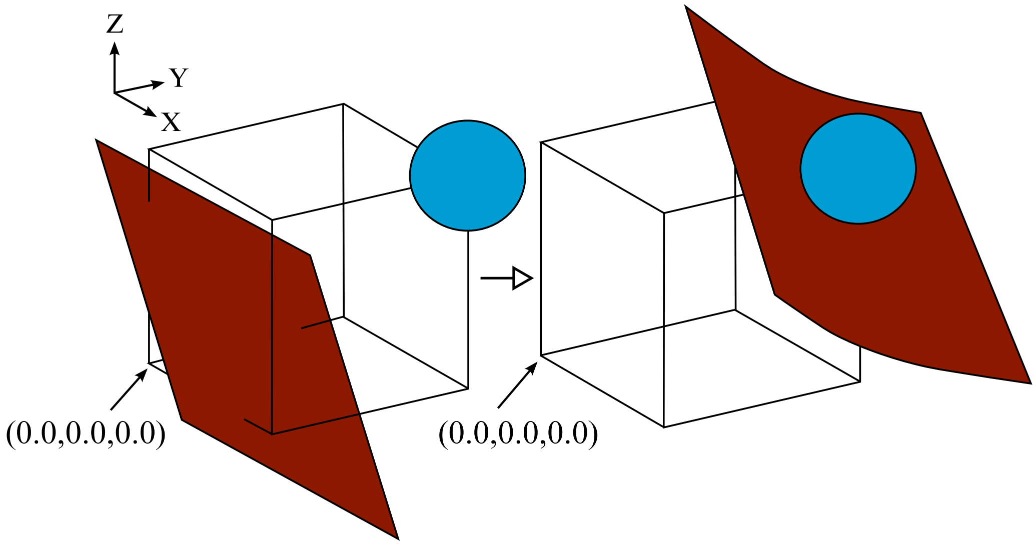


Figure 27-2. A validation example for option **VECTOR**.

$$\mathbf{b} = \rho[\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r})]$$

where ρ is the mass density, $\boldsymbol{\omega}$ is the angular velocity vector, and $\mathbf{r}$ is a position vector from the origin to point **P**. Although the angular velocity may vary with time, the effects of angular acceleration are not included.

Angular velocities are useful for studying transient deformation of spinning three-dimensional objects. Typical applications have included stress initialization during dynamic relaxation where the initial rotational velocities are assigned at the completion of the initialization, and this option ceases to be active.

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$  *LOAD_BODY_Z
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Add gravity such that it acts in the negative Z-direction.
$  Use units of mm/ms2.  Since gravity is constant, the load
$  curve is set as a constant equal to 1.  If the simulation
$  is to exceed 1000 ms, then the load curve needs to be
$  extended.
$
$$$  Note: Positive body load acts in the negative direction.
$
*LOAD_BODY_Z
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$     lcid        sf    lciddr        xc        yc        zc
         5   0.00981
$
$
*DEFINE_CURVE
$     lcid      sidr      scla      sclo      offa      offo
         5
```

```
$
$           abscissa            ordinate
                0.00               1.000
             1000.00               1.000
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

About option VECTOR:

The vector V defines the direction of the body force. Body forces act in the negative direction of the vector V.

In an example shown in Figure 27-2, a rectangular sheet metal blank is loaded with gravity into a ball defined as a fixed rigid body. Given the global coordinate system shown, if the part set ID of the blank is 1, the keywords responsible for specifying the body force (in units of mm, second, tonne and Newton) in positive direction of (1.0, 1.0, 1.0) will be as follows,

```
*LOAD_BODY_PARTS
1
*LOAD_BODY_VECTOR
101, 9810.0
-1.0, -1.0, -1.0
*DEFINE_CURVE
101
0.0, 1.0
10.0, 1.0
```

It is note that straight lines represent a cube with each edge length of 500.0mm.

Revision information:

The_**VECTOR** option is available in LS-DYNA R5 Revision 59290 and later releases.

LOAD_BODY_GENERALIZED_*OPTION

Available options include:

SET_NODE

SET_PART

Purpose: Define body force loads due to a prescribed base acceleration or a prescribed angular motion over a subset of the complete problem. The subset is defined by using nodes or parts. Warning: Nodes, which belong to rigid bodies, should not be specified. Rigid bodies must be included within the part sets definitions.

The body forces defined using this command do not take into account non-physical mass added via mass scaling; see *CONTROL_TIMESTEP.

Card Sets. Include as many sets of Cards 1 and 2 as necessary. This input terminates at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	N1/SID	N2/0	LCID	DRLCID	XC	YC	ZC	
Type	I	I	I	I	F	F	F	
Default	none	none	none	0	0.	0.	0.	
Remarks								

Card 2	1	2	3	4	5	6	7	8
Variable	AX	AY	AZ	OMX	OMY	OMZ	CID	ANGTYP
Type	F	F	F	F	F	F	I	A
Default	0.	0.	0.	0.	0.	0.	0	CENT
Remarks	1, 2	1, 2	1, 2	3, 4, 5	3, 4, 5	3, 4, 5	optional	3

VARIABLE	DESCRIPTION
N1/SID	Beginning node ID for body force load or the node or part set ID.
N2	Ending node ID for body force load. Set to zero if a set ID is defined.
LCID	Load curve ID, see *DEFINE_CURVE.
DRLCID	Load curve ID for dynamic relaxation phase. Only necessary if dynamic relaxation is defined. See *CONTROL_DYNAMIC_RELAXATION.
XC	*x*-center of rotation. Define only for angular motion.
YC	*y*-center of rotation. Define only for angular motion.
ZC	*z*-center of rotation. Define only for angular motion.
AX	Scale factor for acceleration in *x*-direction
AY	Scale factor for acceleration in *y*-direction
AZ	Scale factor for acceleration in *z*-direction
OMX	Scale factor for *x*-angular velocity or acceleration
OMY	Scale factor for *y*-angular velocity or acceleration
OMZ	Scale factor for *z*-angular velocity or acceleration
CID	Coordinate system ID to define acceleration in the local coordinate system. The coordinate (XC, YC, ZC) is defined with respect to the local coordinate system if CID is nonzero. The accelerations, LCID and their scale factors are with respect to CID. EQ.0: global

VARIABLE	DESCRIPTION
ANGTYP	Type of body loads due to angular motion EQ.CENT: body load from centrifugal acceleration, $\rho[\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r})]$. EQ.CORI: body load from Coriolis-type acceleration, $2\rho(\boldsymbol{\omega} \times \mathbf{v})$. EQ.ROTA: body load from rotational acceleration, $\rho(\boldsymbol{\alpha} \times \mathbf{r})$, where $\boldsymbol{\omega}$ is the angular velocity, $\boldsymbol{\alpha}$ is the angular acceleration, $\mathbf{r}$ is the position vector relative to center of rotation and $\mathbf{v}$ is the velocity vector

Remarks:

1. Translational base accelerations allow body forces loads to be imposed on a structure. Conceptually, base acceleration may be thought of as accelerating the coordinate system in the direction specified, and, thus, the inertial loads acting on the model are of opposite sign. For example, if a cylinder were fixed to the *y*-*z* plane and extended in the positive x-direction, then a positive *x*-direction base acceleration would tend to shorten the cylinder, i.e., create forces acting in the negative *x*-direction.

2. Base accelerations are frequently used to impose gravitational loads during dynamic relaxation to initialize the stresses and displacements. During the analysis, in this latter case, the body forces loads are held constant to simulate gravitational loads. When imposing loads during dynamic relaxation, it is recommended that the load curve slowly ramp up to avoid the excitation of a high frequency response.

3. Body force loads due to the angular motion about an axis are calculated with respect to the deformed configuration. When ANGYP = CENT or CORI, torsional effects which arise from changes in angular velocity are neglected. Such torsional effects can be taken into account by setting ANGTYP = ROTA. The angular velocity is assumed to have the units of radians per unit time, accordingly angular acceleration has the units of radians/time^2.

4. The body force density is given at a point **P** of the body by:

$$\boldsymbol{b} = \rho[\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r})]$$

where ρ is the mass density, $\boldsymbol{\omega}$ is the angular velocity vector, and $\mathbf{r}$ is a position vector from the origin to point **P**. Although the angular velocity may vary with time, the effects of angular acceleration are included.

5. Angular velocities are useful for studying transient deformation of spinning three-dimensional objects. Typical applications have included stress initialization during dynamic relaxation where the initial rotational velocities are assigned at the completion of the initialization, and this option ceases to be active.

*LOAD_BODY_POROUS

Purpose: Define the effects of porosity on the flow with body-force-like loads applied to the ALE element nodes. Ergun porous flow assumptions are used. This only applies to non-deformable (constant-porosity), fully saturated porous media. This model only works with a non-zero and constant viscosity fluid defined via either *MAT_NULL or *MAT_ALE_VISCOUS card.

Card Sets. Include as many sets of Cards 1 and 2 as necessary. This input terminates at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	SIDTYP	AX	AY	AZ	BX	BY	BZ
Type	I	I	F	F	F	F	F	F
Default	0	0	0.0	0.0	0.0	0.0	0.0	0.0

Card 2	1	2	3	4	5	6	7	8
Variable	AOPT							
Type	I							
Default	0							

VARIABLE **DESCRIPTION**

SID Set ID of the ALE fluid part subjected to porous flow condition.

SIDTYP Set ID type of the SID above. If SIDTYP = 0 (default), then the SID = PSID (part set ID). If SIDTYP = 1, then SID = PID (part ID).

AX, AY, AZ Viscous coefficients for viscous terms in global x, y, and z directions (please see equation below). If $A_x \neq 0$ and $A_y = A_z = 0$ then an isotropic viscous permeability condition is assumed for the porous medium.

VARIABLE	DESCRIPTION
BX, BY, BZ	Inertial coefficients for inertia terms in global x, y, and z directions (please see equation below). If $B_x \neq 0$, and $B_y = B_z = 0$ then an isotropic inertial permeability condition is assumed for the porous medium.
AOPT	Material axis option: EQ.0: inactive. EQ.1: The forces are applied in a local system attached to the ALE solid (see CTYPE = 12 and DIREC = 1 in *CONSTRAINED_LAGRANGE_IN_SOLID).

Remarks:

1. Consider the basic general Ergun equation for porous flow in one direction:

$$\frac{\Delta P}{\Delta L} = \frac{\mu}{k_1} V_s + \frac{\rho}{k_2} {V_s}^2$$

Where

$$\begin{aligned}
\rho &= \text{Fluid Density} \\
\mu &= \text{Fluid dynamic vicosity} \\
V_s = \frac{4Q}{\pi D^2} &= \text{Superficial fluid velocity} \\
Q &= \text{Overall volume flow rate } \left(\frac{\mathrm{m}^3}{\mathrm{s}}\right) \\
D &= \text{Porous channel characteristic width (perpendicular to } \Delta\text{L)} \\
k_1 = \frac{\varepsilon^3 {d_p}^2}{150(1-\varepsilon)^2} &= \text{Permeability parameter} \\
k_2 = \frac{\varepsilon^3 d_p}{1.75(1-\varepsilon)} &= \text{Passability parameter} \\
\epsilon &= \text{Porosity} = \frac{\text{total pore volume}}{\text{total media volume}} \\
d_p &= \text{Effective particle diameter}
\end{aligned}$$

2. The above equation can be generalized into 3 dimensional flows where each component may be written as

$$-\frac{dP}{dx_i} = A_i \mu V_i + B_i \rho |V_i| V_i$$

where $i = 1,2,3$ refers to the global coordinate directions (no summation intended for repeated indices), μ is the constant dynamic viscosity, ρ is the fluid density, V_i is the fluid velocity components, A_i is analogous to k_1 above, and B_i is analogous to k_2 above. A matrix version can be defined by ALE elements with *DEFINE_POROUS_ALE.

3. If $B_i = 0$, the equation is reduced to simple Darcy Law for porous flow (may be good for sand-like flow). For coarse grain (rocks) media, the inertia term will be important and the user needs to input these coefficients.

*LOAD_BRODE

Purpose: Define Brode function for application of pressure loads due to explosion, see Brode [1970], also see *LOAD_SEGMENT, *LOAD_SEGMENT_SET, or *LOAD_SHELL.

Card 1	1	2	3	4	5	6	7	8
Variable	YLD	BHT	XBO	YBO	ZBO	TBO	TALC	SFLC
Type	F	F	F	F	F	F	I	I
Default	0.0	0.0	0.0	0.0	0.0	0.0	0	0
Remarks							1	1

Card 2	1	2	3	4	5	6	7	8
Variable	CFL	CFT	CFP					
Type	F	F	F					
Default	0.0	0.0	0.0					

VARIABLE	DESCRIPTION
YLD	Yield (Kt, equivalent tons of TNT).
BHT	Height of burst.
XBO	x-coordinates of Brode origin.
YBO	y-coordinates of Brode origin.
ZBO	z-coordinates of Brode origin.
TBO	Time offset of Brode origin.
TALC	Load curve number giving time of arrival versus range relative to Brode origin (space, time), see *DEFINE_CURVE and remark below.

VARIABLE	DESCRIPTION
SFLC	Load curve number giving yield scaling versus scaled time (time relative to Brode origin divided by [yield$^{(**1/3)}$]) origin (space, time), see *DEFINE_CURVE and remark below.
CFL	Conversion factor - kft to LS-DYNA length units.
CFT	Conversion factor - milliseconds to LS-DYNA time units.
CFP	Conversion factor - psi to LS-DYNA pressure units.

Remarks:

1. If these curves are defined a variable yield is assumed. Both load curves must be specified for the variable yield option. If this option is used, the shock time of arrival is found from the time of arrival curve. The yield used in the Brode formulas is computed by taking the value from the yield scaling curve at the current time/[yield$^{(**1/3)}$] and multiplying that value by yield.

*LOAD_DENSITY_DEPTH

Purpose: Define density versus depth for gravity loading. This option has been occasionally used for analyzing underground and submerged structures where the gravitational preload is important. The purpose of this option is to initialize the hydrostatic pressure field at the integration points in the element.

This card should be only defined once in the input deck.

Card 1	1	2	3	4	5	6	7	8
Variable	PSID	GC	DIR	LCID				
Type	I	F	I	I				
Default	0	0.0	1	none				
Remarks	1,2			3				

VARIABLE **DESCRIPTION**

PSID Part set ID, see *SET_PART. If a PSID of zero is defined then all parts are initialized.

GC Gravitational acceleration value.

DIR Direction of loading:

EQ.1: global x,

EQ.2: global y,

EQ.3: global z.

LCID Load curve ID defining density versus depth, see *DEFINE_CURVE.

Remarks:

1. Density versus depth curves are used to initialize hydrostatic pressure due to gravity acting on an overburden material. The hydrostatic pressure acting at a material point at depth, d, is given by:

$$p = -\int_{d}^{d_{\text{surface}}} \rho(z) g dz$$

where p is pressure, d_{surface}, is the depth of the surface of the material to be initialized (usually zero), $\rho\ (z)$ is the mass density at depth z, and g is the acceleration of gravity. This integral is evaluated for each integration point. Depth may be measured along any of the global coordinate axes, and the sign convention of the global coordinate system should be respected. The sign convention of gravity also follows that of the global coordinate system. For example, if the positive z axis points "up", then gravitational acceleration should be input as a negative number.

2. For this option there is a limit of 12 parts that can be defined by PSID, unless all parts are initialized.

3. Depth is the ordinate of the curve and is input as a descending x, y, or z coordinate value. Density is the abscissa of the curve and must vary (increase) with depth, i.e., an infinite slope is not allowed.

4. See also GRAV in *PART.

*LOAD_ERODING_PART_SET

Purpose: Apply a pressure load to the exposed surface composed of solid elements that may erode.

Card Sets. Include as many sets of Cards 1 and 2 as necessary. This input terminates at the next keyword ("*") card.

Card	1	2	3	4	5	6	7	8
Variable	ID	LCID	SF	AT	PSID	BOXID	MEM	ALPHA
Type	I	I	F	F	I	I	I	F
Default	none	none	1	0.0	none	0	50	80

Card 2	1	2	3	4	5	6	7	8
Variable	IFLAG	X	Y	Z	BETA			
Type	I	F	F	F	F			
Default	0	0.0	0.0	0.0	90			

VARIABLE	DESCRIPTION
ID	ID number.
LCID	Load curve ID defining pressure as a function of time, see *DEFINE_CURVE.
SF	Scale factor.
AT	Arrival time.
PSID	Part set ID, see *SET_PART.
BOXID	Box ID, see *DEFINE_BOX. Any segment that would otherwise be loaded but whose centroid falls outside of this box is not loaded.

VARIABLE DESCRIPTION

MEM Extra memory, in percent, to be allocated above the initial memory for storing the new load segments exposed by the erosion.

ALPHA The maximum angle (in degrees) permitted between the normal of a segment at its centroid and the average normal at its nodes. This angle is used to eliminate interior segments.

IFLAG Flag for choosing a subset of the exposed surface that is oriented towards a blast or other loading source. The vector from the center of the element to the source location must be within an angle of BETA of the surface normal. If IFLAG > 0, then the subset is chosen, otherwise if IFLAG = 0, the entire surface is loaded.

X, Y, Z Optional source location.

BETA Maximum permitted angle (in degrees) between the surface normal and the vector to the source. The exposed segment is not loaded if the calculated angle is greater than BETA.

Remarks:

1. If LCID is input as -1, then the Brode function is used to determine the pressure for the segments, see *LOAD_BRODE.

2. If LCID is input as -2, then an empirical air blast function is used to determine the pressure for the segments, see *LOAD_BLAST.

3. The load curve multipliers may be used to increase or decrease the pressure. The time value is not scaled.

4. The activation time, AT, is the time during the solution that the pressure begins to act. Until this time, the pressure is ignored. The function value of the load curves will be evaluated at the offset time given by the difference of the solution time and AT i.e., (solution time-AT).

5. For proper evolution of the loaded surface, it is a requirement that DTMIN in *CONTROL_TERMINATION be greater trhan zero and ERODE in *CONTROL_TIMESTEP be set to 1.

LOAD_GRAVITY_PART_*{OPTION}

Available options are:

<BLANK>

SET

Purpose: Define gravity for individual parts. This feature is intended for use with *LOAD_STIFFEN_PART to simulate staged construction. This keyword is available for solids and shells, and also beam element types 1, 2, 6, and 9. It is not currently available for thick shells.

Part Cards. Include this card as many times as necessary. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	DOF	LC	ACCEL	LCDR	STGA	STGR	
Type	I	I	I	F	I	I	I	
Default	none	none	none	0	none	0	0	

VARIABLE	DESCRIPTION
PID/PSID	Part ID (or Part Set ID for the_SET option) for application of gravity load
DOF	Direction: enter 1, 2 or 3 for x, y or z
LC	Load curve defining factor vs. time (or zero if STGA, STGR are defined)
ACCEL	Acceleration (will be multiplied by factor from curve
LCDR	Load curve defining factor vs. time during dynamic relaxation
STGA	Construction stage at which part is added (optional)
STGR	Construction stage at which part is removed (optional)

Remarks:

1. There are 3 options for defining how the gravity load on a part varies with time.

a) Curve LC gives factor vs time. This overrides the other methods if LC is non-zero.

b) STGA, STGR refer to stages at which part is added and removed – the stages are defined in *DEFINE_CONSTRUCTION_STAGES. If STGA is zero, the gravity load starts at time zero. If not, it ramps up from the small factor FACT (on *CONTROL_STAGED_CONSTRUCTION) up to full value over the ramp time at the start of stage STGA. If STGR is zero, the gravity load continues until the end of the analysis. If not, it ramps down from full value to FACT over the ramp time at the start of stage STGR.

c) *DEFINE_STAGED_CONSTRUCTION_PART can be used instead of *LOAD_GRAVITY_PART to define this loading. During initialization, a LOAD_GRAVITY_PART card will be created and the effect is the same as using the STGA, STGR method described above; ACCEL is then taken from *CONTROL_STAGED_CONSTRUCTION.

2. This feature calculates the loading from the mass of the elements of the referenced parts only (density × volume). This mass does not include any attached lumped mass elements. Only solid, beam and shell elements can be loaded.

*LOAD_HEAT_CONTROLLER

Purpose: Used to define a thermostat control function. The thermostat controls the heat generation within a material by monitoring a remote nodal temperature. Control can be specified as on-off, proportional, integral, or proportional with integral.

Sensor Node Cards. Include up to 20 cards. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	NODE	PID	LOAD	TSET	TYPE	GP	GI	
Type	I	I	F	F	I	F	F	
Default	none	none	none	none	none	none	none	

VARIABLE	DESCRIPTION
NODE	Sensor is located at this node number.
PID	Part ID assigned to the elements modeling the heater or cooler being controlled.
LOAD	Heater output q_0. [typical units: W/m^3]
TSET	Controller set point temperature at the location identified by NODE.
TYPE	Type of control function. EQ.1: on-off EQ.2: proportional + integral
GP	Proportional gain.
GI	Integral gain.

Remarks:

The thermostat control function is

$$\dot{q}''' = \dot{q}_0''' \left[G_P (T_{\text{set}} - T_{\text{node}}) + G_I \int_{t=0}^{t} (T_{\text{set}} - T_{\text{node}})\, dt \right]$$

LOAD_HEAT_GENERATION_*OPTION

Available options include:

SOLID

SET_SOLID

SHELL

SET_SHELL

Purpose: Define elements or element sets with heat generation.

Generation Cards. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	LCID	MULT	WBLCID	CBLCID	TBLCID		
Type	I	I	F	I	I	I		
Default	none	none	0.	0	0	0		

VARIABLE **DESCRIPTION**

SID

Element ID or element set ID, see *ELEMENT_SOLID, *SET_SOLID, *ELEMENT_SHELL, and *SET_SHELL, respectively.

LCID

Volumetric heat generation rate, $\dot{q}'''$, specification. SI units are W/m^3. This parameter can reference a load curve ID (see *DEFINE_CURVE) or a function ID (see *DEFINE_FUNCTION and Remark 1). When the reference is to a curve, LCID has the following interpretation:

GT.0: $\dot{q}'''$ is defined by a curve consisting of (time, $\dot{q}'''$) data pairs.

EQ.0: $\dot{q}'''$ is a constant defined by the value MULT.

LT.0: $\dot{q}'''$ is defined by a curve consisting of (temperature, $\dot{q}'''$) data pairs. Enter |-LCID| on the DEFINE_CURVE keyword.

MULT

Volumetric heat generation, $\dot{q}'''$, curve multiplier.

VARIABLE	DESCRIPTION
WBLCID	Load curve ID defining the blood perfusion rate [e.g., kg/m^3 sec] as a function of time.
CBLCID	Load curve ID defining the blood specific heat [e.g., J/kg C] as a function of the blood temperature.
TBLCID	Load curve ID defining the blood temperature [e.g., C] as a function of time.

Remarks:

1. If LCID references a DEFINE_FUNCTION, the following function arguments are allowed $f(x, y, z, v_x, v_y, v_z, T, t)$ where:

   x, y, z = element centroid coordinates

   v_x, v_y, v_z = element centroid velocity components

   T = element integration point temperature

   t = solution time

2. Rate of heat transfer from blood to tissue $= W_b C_b (T_b - T)$ [units: J/m^3 sec]

*LOAD_MASK

Purpose: Apply a distributed pressure load over a three-dimensional shell part. The pressure is applied to a subset of elements that are within a fixed global box and lie either outside or inside of a closed curve in space which is projected onto the surface.

Card Sets. Include as many sets of Cards 1 and 2 as necessary. This input terminates at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	LCID	VID1	OFF	BOXID	LCIDM	VID2	INOUT
Type	I	I	F	F	I	I	I	I
Default	none	none	1.	0.	0	0	none	0
Remarks	1		2					

Card 2	1	2	3	4	5	6	7	8
Variable	ICYCLE							
Type	I							
Default	200							
Remarks								

VARIABLE	DESCRIPTION
PID	Part ID (PID). This part must consist of 3D shell elements. To use this option with solid element the surface of the solid elements must be covered with null shells. See *MAT_NULL.
LCID	Curve ID defining the pressure time history, see *DEFINE_-CURVE.

VARIABLE **DESCRIPTION**

VID1

Vector ID normal to the surface on which the applied pressure acts. Positive pressure acts in a direction that is in the opposite direction. This vector may be used if the surface on which the pressure acts is relatively flat. If zero, the pressure load depends on the orientation of the shell elements as shown in Figure 27-4.

OFF

Pressure loads will be discontinued if $|VID1 \cdot n_{shell}| < OFF$ where n_{shell} is the normal vector to the shell element.

BOXID

Only elements inside the box with part ID, PID, are considered. If no ID is given all elements of part ID, PID, are included. When the active list of elements are updated, elements outside the box will no longer have pressure applied, i.e., the current configuration is always used.

LCIDM

Curve ID defining the mask. This curve gives (x,y) pairs of points in a local coordinate system defined by the vector ID, VID2. Generally, the curve should form a closed loop, i.e., the first point is identical to the last point, and the curve should be flagged as a DATTYP = 1 curve in the *DEFINE_CURVE section. If no curve ID is given, all elements of part ID, PID, are included with the exception of those deleted by the box. The mask works like the trimming option, i.e., see DEFINE_CURVE_TRIM and Figure15-18.

VID2

Vector ID used to project the masking curve onto the surface of part ID, PID. The origin of this vector determines the origin of the local system that the coordinates of the PID are transformed into prior to determining the pressure distribution in the local system. This curve must be defined if LCIDM is nonzero. See Figure15-18.

INOUT

If 0, elements whose center falls inside the projected curve are considered. If 1, elements whose center falls outside the projected curve are considered.

ICYCLE

Number of time steps between updating the list of active elements (default = 200). The list update can be quite expensive and should be done at a reasonable interval. The default is not be appropriate for all problems.

Remarks:

1. The part ID must consist of 3D shell elements.

*LOAD_MOTION_NODE

Purpose: Apply a concentrated nodal force or moment to a node based on the motion of another node.

Node Cards. This input continues until the next keyword ("*") card.

Card	1	2	3	4	5	6	7	8
Variable	NODE1	DOF1	LCID	SF	CID1	NODE2	DOF2	CID2
Type	I	I	I	F	I	I	I	I
Default	none	none	none	1.	0	0	0	0
Remarks					1			1

VARIABLE	DESCRIPTION
NODE1	Node ID for the concentrated force.
DOF1	Applicable degrees-of-freedom: EQ.1: x-direction of load action, EQ.2: y-direction of load action, EQ.3: z-direction of load action, EQ.4: moment about the x-axis, EQ.5: moment about the y-axis, EQ.6: moment about the z-axis.
LCID	Load curve ID (see *DEFINE_CURVE) or function ID (see *DEFINE_FUNCTION). The applied force is a function of the applicable degree-of-freedom of NODE2.
SF	Load curve scale factor.
CID1	Coordinate system ID (optional), see remark 1 on next page.
NODE2	Node ID for calculating the force.

VARIABLE	DESCRIPTION
DOF2	Applicable degrees-of-freedom: EQ.1: x-coordinate EQ.2: y-coordinate, EQ.3: z-coordinate, EQ.4: x-translational displacement, EQ.5: y-translational displacement, EQ.6: z-translational displacement, EQ.7: rotational displacement about the x-axis, EQ.8: rotational displacement about the y-axis, EQ.9: rotational displacement about the z-axis. EQ.10: x-translational velocity, EQ.11: y-translational velocity, EQ.12: z-translational velocity, EQ.13: rotational velocity about the x-axis, EQ.14: rotational velocity about the y-axis, EQ.15: rotational velocity about the z-axis.
CID2	Coordinate system ID (optional), see Remark 1.

Remarks:

1. The global coordinate system is the default. The local coordinate system ID's are defined in the *DEFINE_COORDINATE_SYSTEM section.

*LOAD_MOVING_PRESSURE

Purpose: Apply moving pressure loads to a nondisjoint surface. The pressure loads approximate a jet of high velocity fluid impinging on the surface. Multiple surfaces may be defined each acted on by a set of nozzles.

Card 1	1	2	3	4	5	6	7	8
Variable	LOADID							
Type	I							
Default	none							

Nozzle Cards. Define the following cards for each nozzle. Include as many cards as desired. This input ends at the first card with the second field (NODE2) <= 3.

Card 2	1	2	3	4	5	6	7	8
Variable	NODE1	NODE2	LCID	CUTOFF	LCIDT	LCIDD		
Type	I	I	I	F	I	I		
Default	none	none	none	none	0	0		

The following card defines the surface where the nozzles act.

Card 3	1	2	3	4	5	6	7	8
Variable	ID	IDTYPE	NIP					
Type	I	I	I					
Default	none	none	3x3					

VARIABLE **DESCRIPTION**

LOADID Loading ID.

NODE1 Node located at the origin of the nozzle.

VARIABLE **DESCRIPTION**

NODE2 Node located at the head of the nozzle

LCID Load curve or function (see *DEFINE_FUNCTION) ID defining pressure versus radial distance from the center of the jet.

CUTOFF Outer radius of jet. The pressure acting outside this radius is set to zero.

LCIDT Load curve or function (see *DEFINE_FUNCTION) ID, which scales the pressure as a function of time. If a load curve isn't specified, the scale factor defaults to 1.0.

LCIDD Load curve or function (see *DEFINE_FUNCTION) ID, which scales the pressure as a function of distance from the nozzle. If a load curve isn't specified, the scale factor defaults to 1.0.

ID Segment set ID, shell element set ID, part set ID, or part ID. See IDT below.

IDT Slave segment or node set type. The type must correlate with the number specified for SSID:

EQ.0: segment set ID for surface-to-surface contact,

EQ.1: shell element set ID for surface-to-surface contact,

EQ.2: part set ID,

EQ.3: part ID,

NIP Number of integration in segment used to compute pressure loads.

LOAD_NODE_*OPTION

Available options include:

POINT

SET

Purpose: Apply a concentrated nodal force to a node or each node in a set of nodes.

Node/Node set Cards. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card	1	2	3	4	5	6	7	8
Variable	NID/NSID	DOF	LCID	SF	CID	M1	M2	M3
Type	I	I	I	F	I	I	I	I
Default	none	none	none	1.	0	0	0	0
Remarks					1	2		

VARIABLE	DESCRIPTION
NID/NSID	Node ID or nodal set ID (NSID), see *SET_NODE_*OPTION*.
DOF	Applicable degrees-of-freedom: EQ.1: x-direction of load action, EQ.2: y-direction of load action, EQ.3: z-direction of load action, EQ.4: follower force (see Remark 2), EQ.5: moment about the x-axis (see Remark 4), EQ.6: moment about the y-axis axis (see Remark 4), EQ.7: moment about the z-axis axis (see Remark 4), EQ.8: follower moment (see Remarks 2 and 4).
LCID	Load curve ID (see *DEFINE_CURVE) or function ID (see *DEFINE_FUNCTION and Remark 5 below).

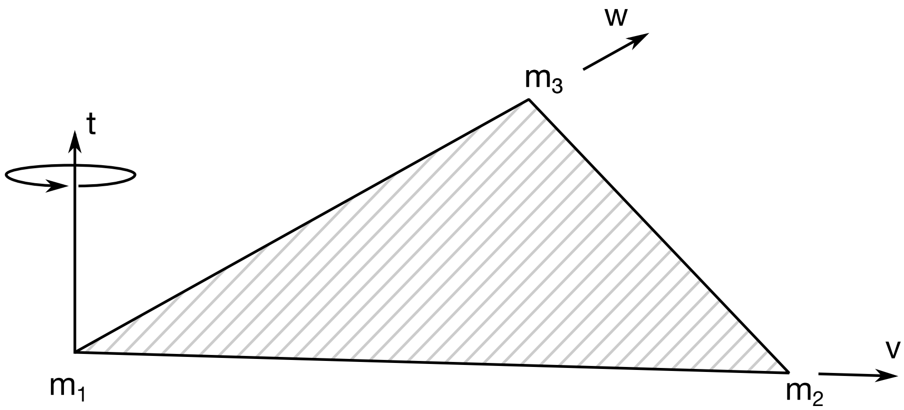


Figure 27-3. Nodes M1, M2, and M3 define a plane. A positive follower force acts in the positive t-direction of that plane, i.e., along the normal vector of the plane. A positive follower moment puts a counterclockwise torque about the normal vector. The normal vector is found by the cross product $\mathbf{t} = \mathbf{v} \times \mathbf{w}$ where $\mathbf{v}$ and $\mathbf{w}$ are vectors as shown.

VARIABLE	DESCRIPTION
SF	Load curve scale factor.
CID	Coordinate system ID (optional), see Remark 1 on next page.
M1	Node 1 ID. Only necessary if DOF.EQ.4 or 8, see Remark 2 below.
M2	Node 2 ID. Only necessary if DOF.EQ.4 or 8, see Remark 2 below.
M3	Node 3 ID. Only necessary if DOF.EQ.4 or 8, see Remark 2 below.

Remarks:

1. **Coordinate Systems.** The global coordinate system is the default. The local coordinate system ID's are defined in the *DEFINE_COORDINATE_SYSTEM section.

2. **Follower Forces.** The current position of nodes M_1, M_2, M_3 are used to control the direction of a follower force. A positive follower force acts normal to the plane defined by these nodes, and a positive follower moment puts a counterclockwise torque about the t-axis. These actions are depicted in Figure 27-3. An alternative way to define the force direction is by setting M_3 to any nonpositive value, in which case the follower force is in the M_1 *to* M_2 direction.

3. **Axisymmetric Elements with Area and Volume Weighting.** For shell formulations 14 and 15, the axisymmetric solid elements with area and volume weighting, respectively, the specified nodal load is per unit length (type14) and per radian (type 15).

4. **Moments.** Moments can only be applied to nodes that have rotational degrees of freedom. Element type and formulation determine the degrees of freedom for a node, e.g., the nodes of solid formulation 1 have only 3 translational degrees of freedom and no rotational degrees of freedom.

5. ***DEFINE_FUNCTION for LCID.** The function defined by LCID has 7 arguments: time, the 3 current coordinates, and the 3 reference coordinates. A function that applies a force proportional to the distance from the initial coordinates would be:

$$f(t,x,y,z,x0,y0,z0)= -10.*\mathrm{sqrt} ((x-x0)*(x-x0)+(y-y0)*(y-y0)+(z-z0)*(z-z0))$$

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$   *LOAD_NODE_SET
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  A cantilever beam (made from shells) is loaded on the two end nodes
$  (nodes 21 & 22).  The load is applied in the y-direction (dof=2).
$  Load curve number 1 defines the load, but is scaled by sf=0.5 in the
$  *LOAD_NODE_SET definition.
$
*LOAD_NODE_SET
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$     nsid       dof      lcid        sf       cid        m1        m2        m3
        14         2         1       0.5
$
$
*SET_NODE_LIST
$      sid
        14
$
$     nid1      nid2      nid3      nid4      nid5      nid6      nid7      nid8
        21        22
$
$
*DEFINE_CURVE
$     lcid      sidr      scla      sclo      offa      offo
         1
$
$           abscissa           ordinate
                 0.0                 0.0
                10.0               100.0
                20.0                 0.0
$
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

LOAD_REMOVE_PART_*{OPTION}

Available options include:

<BLANK>

SET

Purpose: Delete the elements of a part in a staged construction simulation. Shock effects are prevented by gradually reducing the stresses prior to deletion. Available only for solid and shell elements.

Note: This keyword card will be available starting in release 3 of version 971.

Part Cards. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID/PSID	TIME0	TIME1	STGR				
Type	I	F	F	I				
Default	none	0	0	0				

VARIABLE	DESCRIPTION
PID	Part ID (or Part Set ID for the_SET option) for deletion
TIME0	Time at which stress reduction starts
TIME1	Time at which stresses become zero and elements are deleted
STGR	Construction stage at which part is removed (optional)

Remarks:

There are 3 methods of defining the part removal time:

1. TIME0, TIME1 override all the other methods if non-zero

2. STGR refers to the stage at which the part is removed – the stages are defined in *DEFINE_CONSTRUCTION_STAGES. This is equivalent to setting TIME0 and

TIME1 equal to the start and end of the ramp time at the beginning of stage STGR.

3. *DEFINE_STAGED_CONSTRUCTION_PART can be used instead of *LOAD_-REMOVE_PART to define this loading. During initialization, a STIFFEN_-PART card will be created and the effect is the same as using the STGA, STGR method described above.

*LOAD_RIGID_BODY

Purpose: Apply a concentrated nodal force to a rigid body. The force is applied at the center of mass or a moment is applied around a global axis. As an option, local axes can be defined for force or moment directions.

Rigid Body Cards. Include as many Rigid Body Cards as necessary. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	DOF	LCID	SF	CID	M1	M2	M3
Type	I	I	I	F	I	I	I	I
Default	none	none	none	1.	0	0	0	0
Remark					1	2		

VARIABLE	DESCRIPTION
PID	Part ID of the rigid body, see *PART_*OPTION*.
DOF	Applicable degrees-of-freedom: EQ.1: x-direction of load action, EQ.2: y-direction of load action, EQ.3: z-direction of load action, EQ.4: follower force, see Remark 2, EQ.5: moment about the x-axis, EQ.6: moment about the y-axis, EQ.7: moment about the z-axis. EQ.8: follower moment, see Remark 2.
LCID	Load curve ID (see *DEFINE_CURVE) or function ID (see *DEFINE_FUNCTION). GT.0: force as a function of time, LT.0: force as a function of the absolute value of the rigid body displacement. This option only applies to load curves.

VARIABLE	DESCRIPTION
SF	Load curve scale factor
CID	Coordinate system ID
M1	Node 1 ID. Only necessary if DOF.EQ.4 or 8, see Remark 2.
M2	Node 2 ID. Only necessary if DOF.EQ.4 or 8, see Remark 2.
M3	Node 3 ID. Only necessary if DOF.EQ.4 or 8, see Remark 2.

Remarks:

1. The global coordinate system is the default. The local coordinate system ID's are defined in the *DEFINE_COORDINATE_SYSTEM section. This local axis is fixed in inertial space, i.e., it does not move with the rigid body.

2. Nodes M_1, M_2, M_3 must be defined for a follower force or moment. The follower force acts normal to the plane defined by these nodes as depicted in Figure 27-3. The positive t-direction is found by the cross product $t = v \times w$ where v and w are vectors as shown. The follower force is applied at the center of mass. A positive follower moment puts a counterclockwise torque about the t-axis.

3. When LCID defines a function, the function has seven arguments: time, the 3 current coordinates for the center of mass, and the 3 reference coordinates. A function that applies a force proportional to the distance from the initial coordinates would be

   f(t,x,y,z,x0,y0,z0)= -10.*sqrt ((x-x0)*(x-x0)+(y-y0)*(y-y0)+(z-z0)*(z-z0)).

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$   *LOAD_RIGID_BODY
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  From a sheet metal forming example.  A blank is hit by a punch, a binder is
$  used to hold the blank on its sides.  The rigid holder (part 27) is held
$  against the blank using a load applied to the cg of the holder.
$
$  The direction of the load is in the y-direction (dof=2) but is scaled
$  by sf = -1 so that the load is in the correct direction.  The load
$  is defined by load curve 12.
$
*LOAD_RIGID_BODY
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      pid       dof      lcid        sf       cid        m1        m2        m3
        27         2        12      -1.0
$
$
*DEFINE_CURVE
$     lcid      sidr      scla      sclo      offa      offo
        12
$
$           abscissa            ordinate
           0.000E+00           8.000E-05
           1.000E+04           8.000E-05
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

LOAD_SEGMENT_*{OPTION}

To define an ID for the segment loading, the following option is available:

ID

If the ID is defined an additional card is required.

Purpose: Apply the distributed pressure load over one triangular or quadrilateral segment defined by four, six, or eight nodes, or in the case of two-dimensional geometries, over one two-noded line segment. The pressure and node numbering convention follows the Figure 27-4 shown in the remarks below.

ID Card. Additional card for the ID keyword option.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	HEADING						
Type	I	A70						

Card 2	1	2	3	4	5	6	7	8
Variable	LCID	SF	AT	N1	N2	N3	N4	N5
Type	I	F	F	I	I	I	I	I
Default	none	1.	0.	none	none	none	none	none
Remarks	1	2	3	4,5,6,7				

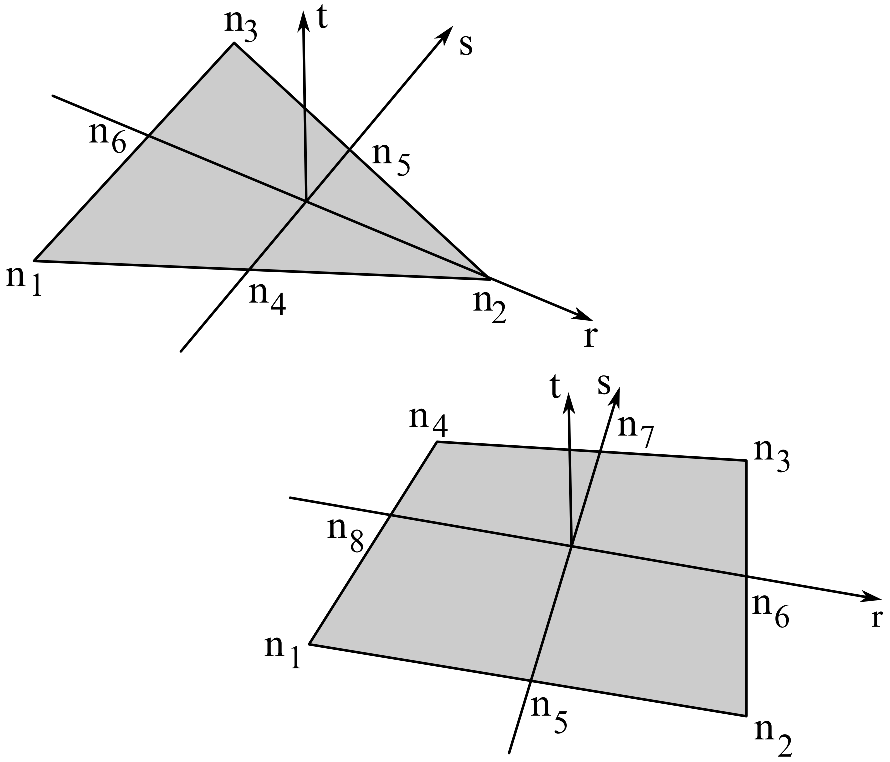


Figure 27-4. Nodal numbering for pressure segments in three-dimensional geometries. Positive pressure acts in the negative t-direction.

Addition card for when N5 ≠ 0.

Card 3	1	2	3	4	5	6	7	8
Variable	N6	N7	N8					
Type	I	I	I					
Default	none	none	none					

VARIABLE	DESCRIPTION
ID	Loading ID
HEADING	A description of the loading.
LCID	Load curve ID (see *DEFINE_CURVE) or function ID (see *DEFINE_FUNCTION).
SF	Load curve scale factor

VARIABLE	DESCRIPTION
AT	Arrival time for pressure or birth time of pressure.
N1	Node ID
N2	Node ID
N3	Node ID
N4	Node ID. Repeat N3 for 3-node triangular segments.
N5	Mid-side node ID, if applicable (see Figure 27-4 and remarks).
N6	Mid-side node ID, if applicable (see Figure 27-4 and remarks).
N7	Mid-side node ID, if applicable (see Figure 27-4 and remarks).
N8	Mid-side node ID, if applicable (see Figure 27-4 and remarks).

Remarks:

1. If LCID is input as -1, then the Brode function is used to determine the pressure for the segments, see *LOAD_BRODE. If LCID is input as -2, then an empirical airblast function is used to determine the pressure for the segments, see *LOAD_BLAST.

2. The load curve multipliers may be used to increase or decrease the pressure. The time value is not scaled.

3. The activation time, AT, is the time during the solution that the pressure begins to act. Until this time, the pressure is ignored. The function value of the load curves will be evaluated at the offset time given by the difference of the solution time and AT i.e., (solution time-AT).

4. Triangular segments without mid-side nodes are defined by setting N4 = N3. To apply a uniform pressure to type 17 tetrathedral elements, 4 triangular segments should be defined for each loaded face of an element. Three of the segments should each have one corner node and the two adjacent mid-side node. The 4th segment should be made from the 3 mid-side nodes.

5. To apply a uniform pressure to type 16 tetrahedral elements or type 24 type triangular shell elements, one 6 node segment may be defined for each loaded face. However, LS-DYNA will accept and properly treat "any" other segment definition. In other words, if the preprocessor happens to create *one* 3-noded segment corresponding to local node numbering convention {1,2,3,3} in Figure

27-4, or permutations thereof, then this will internally be detected as a 6-noded segment and the nodal forces will be consistently distributed over the 6 involved nodes. Likewise, if four 3-noded segments corresponding to local node numbering conventions {1,4,6,6}, {2,5,4,4}, {3,6,5,5} and {4,5,6,6}, or permutations thereof, are created, then these will be detected to belong to the same 6-noded segment and again result in the correct nodal forces. The difference between the two latter approaches is that using four "sub-segments" will provide four different normal directions ***t*** instead of just one, and for curved faces this will provide a more accurate representation of the pressure load.

6. To apply a uniform pressure to type 23 hexahedral elements or type 23 quadrilateral shell elements, one 8 node segment may be defined for each loaded face. However, LS-DYNA will accept and properly treat a segment excluding the 4 mid side nodes. In other words, if the preprocessor happens to create just one 4-noded segment corresponding to local node numbering convention {1,2,3,4} in Figure 27-4, or permutations thereof, then this will internally be detected as a 8-noded segment and the nodal forces will be consistently distributed over the 8 involved nodes.

7. To apply a uniform pressure to type 24 hexahedral elements, either one 4 node segment corresponding to the numbering convention {1,2,3,4} in Figure 27-4, or permutations thereof, may be defined for each loaded face. But LS-DYNA will also accept and properly treat *four* 4 node segments corresponding to {1,5,9,8}, {2,6,9,5}, {3,7,9,6} and {4,8,9,7}, where local node number 9 is located in the center of the face. The difference between these two approaches is that using four "sub-segments" will provide four different normal directions ***t*** instead of just one, and for curved faces this will provide a more accurate representation of the pressure load.

8. Segments for two-dimensional geometries are defined by two nodes, N1 and N2. Leave N3 and N4 as zero or else set both equal to N2. A positive pressure acts on the segment in the Z x (N1-N2) direction where Z is the global Z-axis and (N1-N2) is the vector from N1 to N2.

9. The function defined by LCID has 7 arguments: time, the 3 current coordinates, and the 3 reference coordinates. A function that applies a pressure proportional to the distance from the initial coordinates would be:

```
f(t,x,y,z,x0,y0,z0)= sqrt ( (x-x0)*(x-x0)+(y-y0)*(y-y0)+(z-z0)*(z-z0) )
```

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$   *LOAD_SEGMENT
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  A block of solid elements is pressed down onto a plane as it moves along
$  that plane.  This pressure is defined using the *LOAD_SEGMENT keyword.
$
$  The pressure is applied to the top of the block.  This top is defined
$  by the faces on top of the appropriate solid elements.  The faces are
$  defined with segments.  For example, nodes 97, 106, 107 & 98 define
$  a top face on one of the solids (and thus, one of the faces to apply the
$  pressure too).  This "face" is referred to as a single segment.
$
$  The load is defined with load curve number 1.  The curve starts at zero,
$  ramps to 100 in 0.01 time units and then remains constant.  However,
$  the curve is then scaled by sclo = 2.5.  Thus, raising the load to 250.
$  Note that the load is NOT scaled in the *LOAD_SEGMENT keyword, but
$  could be using the sf variable.
$
*LOAD_SEGMENT
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$     lcid        sf        at        n1        n2        n3        n4
         1      1.00       0.0        97       106       107        98
         1      1.00       0.0       106       115       116       107
         1      1.00       0.0        98       107       108        99
         1      1.00       0.0       107       116       117       108
$
$
*DEFINE_CURVE
$
$     lcid      sidr      scla      sclo      offa      offo
         1         0       0.0       2.5
$
$           abscissa            ordinate
               0.000                 0.0
               0.010               100.0
               0.020               100.0
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

*LOAD_SEGMENT_CONTACT_MASK

Purpose: Mask the pressure from a *LOAD_SEGMENT_SET when the pressure segments are in contact with another material. This keyword is currently only supported in the MPP version. Note that the Heading Card is required.

Heading Card.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	HEADING						
Type	I	A70						

Load Set Card.

Card 2	1	2	3	4	5	6	7	8
Variable	LSID	P1	P2	CID1	CID2	CID3	CID4	CID5
Type	I	F	F	I	I	I	I	I

Optional Cards. This data ends at the next "*" card.

Card 2	1	2	3	4	5	6	7	8
Variable	CID6	CID7	CID8	CID9	CID10	CID11N	CID12	CID13
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
LSID	Load set ID to mask, which must match a *LOAD_SEGMENT_-SET. See Remark 2.
P1	Lower pressure limit. When the surface pressure due to contact is below P1, no masking is done and the full load defined in *LOAD_SEGMENT_SET is applied. For pressures between P1 and P2 see Remark 1.

VARIABLE	DESCRIPTION
P2	Upper pressure limit. When the surface pressure due to contact is above P2, no load is applied due to the *LOAD_SEGMENT_SET. For pressures between P1 and P2 see Remark 1.
CID*n*	The IDs of contacts that can mask the pressure loads. The specified contact definitions must all be of the same type. Furthermore, only non-automatic SURFACE_TO_SURFACE (two way) or AUTOMATIC_SURFACE_TO_SURFACE_TIEBREAK are supported. For TIEBREAK contacts, pressure is masked until the tie fails. Once the tie fails, the full pressure will be applied for the remainder of the simulation. The values P1 and P2 are ignored. For other contact types, the contact forces, along with the nodal contact surface areas, are used to compute the contact pressure at each node to determine any masking effect.

Remarks:

1. **Intermediate Pressures.** If the contact pressure, p_{contact} is between P1 and P2, the pressure load is scaled by a factor of

$$f = \frac{\mathrm{P2} - p_{\mathrm{contact}}}{P2 - P1}.$$

P1 may be set equal to P2 if desired.

2. **LSID.** The LSID values must be unique. Having two instances of this referencing the same*LOAD_SEGMENT_SET *is not supported*. However, a contact ID *may* appear in two different instances of this keyword.

*LOAD_SEGMENT_FILE

Purpose: To define *time-varying* distributed pressure loads over triangular or quadrilateral segments defined by four, six, or eight nodes *via a binary file*.

Card 1	1
Variable	FILENAME
Type	A80

Card 2 is required but may be left blank.

Card 2	1	2	3	4	5	6	7	8
Variable	LCID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
FILENAME	Filename of binary database containing segment pressures versus time. There are three sections in this database.
LCID	Optional Load curve ID defining segment pressure scale factor versus time.

Remarks:

Each database is assumed to be written with a block size equal to a multiple of 512 words, with each block containing 1 or more states. If the data does not complete the last block it is padded with zeros. The code will open and read next family file if it detects phony word or EOF. If the IO returns a zero length read, the code assumes end of IO and strop reading.

Linear interpolation is used if IO can find current time between two states. Therefore, the given database can have more states than simulation time steps.

Linear extrapolation will only be used when IO reaches end of the database. The last two states will be used and code will issue warning message "MSG_SOL+1323" to d3hsp and messag files.

Section	Words	Description
Control Section 64 words	10	Title
	1	NSEG: The absolute value, |"NSEG" | specifies the number of segments contained within the file. If NSEG < 0, then the file contains mid-side nodes.
Segment Data 4 words	1	N1; Node ID.
	1	N2: Node ID.
	1	N3: Node ID. Repeat N2 for two-dimensional geometries.
	1	N4: Node ID. Repeat N2 for two-dimensional geometries or repeat N3 for 3-node triangular segments.
Mid-side Nodes 4 Words. Ommitted unless NSEG < 0.	1	N5: Optional mid-side node ID.
	1	N6: Optional mid-side node ID.
	1	N7: Optional mid-side node ID.
	1	N8: Optional mid-side node ID.
⋮	4 or 8	⋮
NSEGth segment data	4 or 8	Last set of segment data
"State" Section 1 + |NSEG| words	1	Time
	1	Segment Pressure
	⋮	⋮

	1	Pressure of last segment
⋮	1	⋮
	NSEG	⋮

*LOAD_SEGMENT_FSILNK

Purpose: Apply distributed pressure loads from a previous ALE analysis to a specified segment set in the current analysis. This capability trades some of the model's accuracy for a large reduction in model size.

> NOTE: The deck for the "previous" run must include a *DATABASE_BINARY_FSILNK card to activate the creation of the fsilink file. Either the *LOAD_SEGMENT_FSILNK card (this card) or the *DATABASE_BINARY_FSILNK card may be in an input deck, *but not both*.

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	A80							

Card 2	1	2	3	4	5	6	7	8
Variable	NINT	LCID						
Type	I	I						
Default	none	0						

Coupling ID Cards. Read in NINT coupling IDs. Repeat this card as many times as necessary to input all NINT values.

Card 3	1	2	3	4	5	6	7	8
Variable	ID1	ID2	ID3	ID4	ID5	ID6	ID7	ID8
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
FILENAME	Filename of the interface linking file.
NINT	Number of couplings for which the previous run provides pressure data.
LCID	Load curve ID (see *DEFINE_CURVE) or function ID (see *DEFINE_FUNCTION). The curve referred to by LCID provides a scale factor as a function of time. The pressure data that is read in from the fsilnk file is scaled according to this value.
ID*i*	These must match COUPIDs from the *CONSTRAINED_LAGRANGE_IN_SOLID card from the previous runs. These IDs specify which of the first run's couplings are propagated into this run through pressure data read from the fsilnk file.

The algorithm:

This feature provides a method for using pressure time history data from *CONSTRAINED_LAGRANGE_IN_SOLID Lagrangian-to-ALE couplings in one calculation as pressure data for the same segment in subsequent calculations. The time range covered by the subsequent calculation *must* overlap the time range of the initial calculation.

First calculation: Write out pressure data.

1. Add a *DATABASE_BINARY_FSILNK card to the first run.
2. Specify filename for the fsilnk file by adding a command line argument to ls-dyna.

```
ls-dyna … fsilnk=fsi_filename …
```

   Without a *DATABASE_BINARY_FSILNK card this command line argument will have no effect.

Format of fsilnk File:

```
WRITE: Job title (character*80 TITLE)
WRITE: Number of interfaces (integer NINTF)
For $i$ = 1 to NINTF {
    WRITE: Number of segments in the $i^{th}$ interface (integer NSEG[$i$])
    For $j$ = 1 to NSEG[$i$] {
            WRITE: Connectivities of $j^{th}$ segment in the $i^{th}$ interface (integer*4)
        }
}
For $n$ = 1 to number of time steps {
        WRITE: time value for the $n^{th}$ time step (real)
        For $i$ = 1 to NINTF
            For $j$ = 1 to NSEG[$i$] {
                WRITE: Pressure for the $j^{th}$ segment of the $i^{th}$ interface (real)
            }
        }
}
```

Subsequent calculations: Read fsilnk File.

Include this keyword, *LOAD_SEGMENT_FSILNK, and be careful to remove the *DATABASE_BINARY_FSILNK keyword. Specify the name of the fsilnk file from the previous run on *LOAD_SEGMENT_FSILNK's first data card.

Then, at each time step, the pressure of the specified coupling IDs is set on the Lagrangian-mesh side from data in the fsilnk file. For times outside of the fsilnk file's range LS-DYNA extrapolates.

1. If current time is before the 1st fsilnk time, then pressure is set to 0.
2. If the current time is in the range of times in the fsilnk file, then the pressure is linearly interpolated from the data at the two time states in the fsilnk file bracketing the current time.
3. If the current time is after the last fsilnk time, then the pressure is set to the fsilnk pressure at last time step.

LOAD_SEGMENT_NONUNIFORM_*{OPTION}

To define an ID for the non-uniform segment loading the following option is available:

ID

If the ID is defined an additional card is required.

Purpose: Apply a distributed load over one triangular or quadrilateral segment defined by three, four, six, or eight nodes. The loading and node numbering convention follows Figure 27-3.

ID Card. Additional card for ID keyword option.

Optional	1	2	3	4	5	6	7	8
Variable	ID	HEADING						
Type	I	A70						

Card Sets. Each segment is specified by a set of the following 3 cards. Include as many sets as necessary. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	LCID	SF	AT	DT	CID	V1	V2	V3
Type	I	F	F	F	I	F	F	F
Default	none	1.	0.	1.E+16	0	none	none	none
Remarks	1	2	3	3	4			

Card 2	1	2	3	4	5	6	7	8
Variable	N1	N2	N3	N4	N5	N6	N7	N8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

Card 3	1	2	3	4	5	6	7	8
Variable	P1	P2	P3	P4	P5	P6	P7	P8
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
ID	Loading ID
HEADING	A description of the loading.
LCID	Load curve ID (see *DEFINE_CURVE) or function ID (see *DEFINE_FUNCTION). For a load curve ID the load curve must provide pressure as a function of time. For a function ID, the function is expected to have seven arguments: current time minus the birth time, the current x, y, and z coordinates, and the initial x, y, and z coordinates. LT.0: Applies to 3, 4, 6 and 8-noded segments. With this option the load becomes a follower load, meaning that the direction of the load is constant with respect to the local segment coordinate system.
SF	Load curve scale factor
AT	Arrival/birth time for the traction load.
DT	Death time for the traction load.
CID	Coordinate system ID

VARIABLE	DESCRIPTION
V1, V2, V3	Vector direction cosines relative to coordinate system CID defining the direction of the traction loading. Note that for LCID.LT.0 this vector rotates with the geometry of the segment.
N1	Node ID
N2	Node ID
N3	Node ID. Repeat N2 for two-dimensional geometries.
N4	Node ID. Repeat N2 for two-dimensional geometries or repeat N3 for 3-node triangular segments.
N5	Optional mid-side node ID (see Fig. 27.3).
N6	Optional mid-side node ID (see Fig. 27.3).
N7	Optional mid-side node ID (see Fig. 27.3).
N8	Optional mid-side node ID (see Fig. 27.3).
P1	Scale factor at node ID, N1.
P2	Scale factor at node ID, N2.
P3	Scale factor at node ID, N3.
P4	Scale factor at node ID, N4.
P5	Scale factor at node ID, N5.
P6	Scale factor at node ID, N6.
P7	Scale factor at node ID, N7.
P8	Scale factor at node ID, N8.

LOAD_SEGMENT_SET_*{OPTION}

To define an ID for the segment loading, the following option is available:

ID

If the ID is defined an additional card is required.

Purpose: Apply the distributed pressure load over each segment in a segment set. See *LOAD_SEGMENT for a description of the pressure sign convention and remarks on high order segment definitions.

ID Card. Additional card for the ID keyword option.

Optional	1	2	3	4	5	6	7	8
Variable	ID	HEADING						
Type	I	A70						

Segment Set Cards. Include as many segment set cards as necessary. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	SSID	LCID	SF	AT				
Type	I	I	F	F				
Default	none	none	1.	0.				
Remarks		1	2	3				

VARIABLE	DESCRIPTION
SSID	Segment set ID, see *SET_SEGMENT.

VARIABLE **DESCRIPTION**

LCID Load curve ID (see *DEFINE_CURVE) or function ID (see *DEFINE_FUNCTION). For a load curve ID the load curve must provide pressure as a function of time. For a function ID the function is expected to have seven arguments: current time minus the birth time, the current x, y, and z coordinates, and the initial x, y, and z coordinates.

SF Load curve scale factor

AT Arrival time for pressure or birth time of pressure.

Remarks:

1. If LCID is input as -1, then the Brode function is used to determine pressure for the segment set, also see *LOAD_BRODE. If LCID is input as -2, then an empirical airblast function is used to determine the pressure for the segments, see *LOAD_BLAST.

2. The load curve multipliers may be used to increase or decrease the pressure. The time value is not scaled.

3. The activation time, AT, is the time during the solution that the pressure begins to act. Until this time, the pressure is ignored. The function value of the load curves will be evaluated at the offset time given by the difference of the solution time and AT i.e., (solution time-AT).

*LOAD_SEGMENT_SET_ANGLE

Purpose: Apply the traction load over a segment set that is dependent on the orientation of a vector. An example application is applying a pressure to a cylinder as a function of the crank angle in an automobile engine. The pressure and node numbering convention follows Figure 27-4.

Card Sets. Include as many sets of Cards 1 and 2 as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	IDSS	LCID	SCALE	IOPTP	IOPTD		
Type	I	I	I	F	I	I		
Default	none	none	none	1.	0	0		

Card 2	1	2	3	4	5	6	7	8
Variable	N1	N2	NA	NI				
Type	I	I	I	I				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
ID	Loading ID
IDSS	Segment set ID.
LCID	Load curve or function ID defining the traction as a function of the angle. If IOPT = 0 below, define the abscissa between 0 and 2π radians or 0 and 360 degrees if IOPD = 1.
SCALE	Scale factor on value of the load curve or function.

VARIABLE DESCRIPTION

IOPTP Flag for periodicity. The default (IOPTP = 0) requires the load curve to be defined between 0 and 2π. This is useful, for example, for modeling an engine that is running at a steady state since each rotation will experience the same loading. To model a transient response, IOPTP = 1 uses a load curve defined over the full range of angles, permitting a different response on the second and subsequent revolutions.

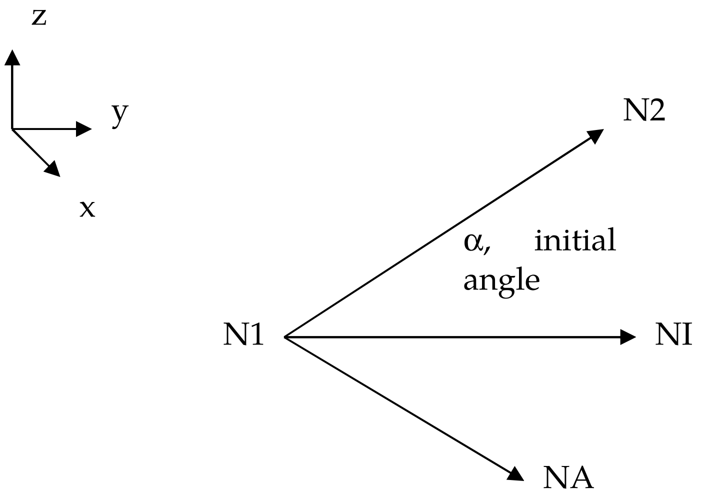
IOPTD Flag for specifying if the load curve or function argument is in radians (IOPTD = 0, the default) or degrees (IOPTD = 1).

N1 The node specifying the tail of the rotating vector.

N2 The node specifying the head of the rotating vector.

NA The node specifying the head of the vector defining the axis of rotation. The node N1 specifies the tail.

NI The node specifying the orientation of the vector at an angle of zero. If the initial angle is zero, NI should be equal to N2.



LOAD_SEGMENT_SET_NONUNIFORM_*{OPTION}

To define an ID for the non-uniform segment loading the following option is available:

ID

If the ID is defined an additional card is required.

Purpose: Apply the traction load over one triangular or quadrilateral segment defined by three or four nodes. The pressure and node numbering convention follows Figure 27-4.

ID Card. Additional card for ID keyword option.

Optional	1	2	3	4	5	6	7	8
Variable	ID	HEADING						
Type	I	A70						

Card Sets. Include as many pairs of Cards 1 and 2 as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	SSID	LCID	SF	AT	DT	ELTYPE		
Type	I	I	F	F	F	A		
Default	none	none	1.	0.	10^{16}	none		

Card 2	1	2	3	4	5	6	7	8
Variable	CID	V1	V2	V3				
Type	I	F	F	F				
Default	0	none	none	none				

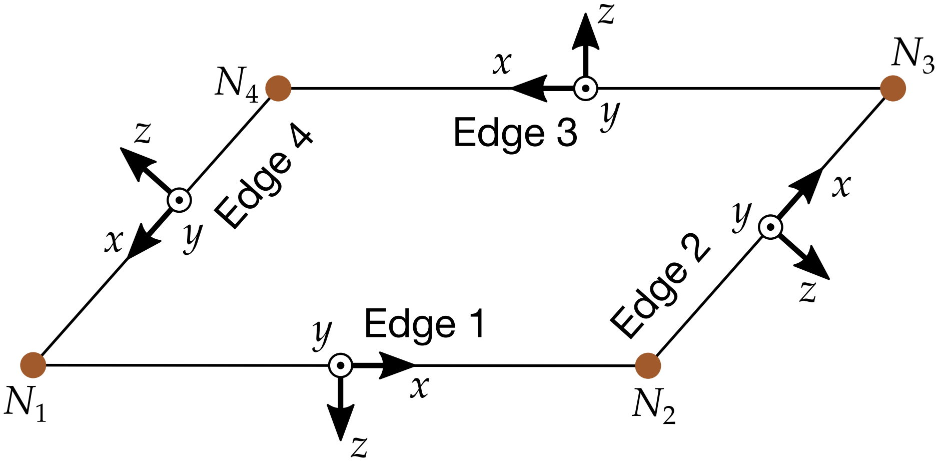


Figure 27-5. Local coordinates system for edge load.

VARIABLE **DESCRIPTION**

ID Loading ID

HEADING A description of the loading.

SSID Segment set ID.

LCID Load curve ID (see *DEFINE_CURVE) or function ID (see *DEFINE_FUNCTION). The seven arguments for the function are current time minus the birth time, the current x, y, and z coordinates, and the initial x, y, and z coordinates.

LT.0: Applies to 3, 4, 6 and 8-noded segment sets. With this option the load becomes a follower load, meaning that the direction of the load is constant with respect to the local segment coordinate system. The local coordinate system for edge load, see ELTYPE, is shown in Figure 27-5.

SF Load curve scale factor

AT Arrival/birth time for pressure.

DT Death time for pressure.

VARIABLE	DESCRIPTION
ELTYPE	Optional edge loading type. If left blank, pressure on the segment will be applied. EQ.EF1: Distributed force per unit length along edge 1, Figure 27-5. EQ.EF2: Distributed force per unit length along edge 2, Figure 27-5. EQ.EF3: Distributed force per unit length along edge 3, Figure 27-5. EQ.EF4: Distributed force per unit length along edge 4, Figure 27-5.
CID	Coordinate system ID
V1, V2, V3	Vector direction cosines relative to the coordinate system CID defining the direction of the traction loading. Note that for LCID.LT.0 this vector rotates with the geometry of the segment.

*LOAD_SEISMIC_SSI_*OPTION1*_*{OPTION2}*

Available options for *OPTION1* include:

NODE

SET

POINT

OPTION2 allows an optional ID to be given:

ID

Purpose: Apply earthquake load due to free-field earthquake ground motion at certain locations — defined by either nodes or coordinates — on a soil-structure interface, for use in earthquake soil-structure interaction analysis. The specified motions are used to compute a set of effective forces in the soil elements adjacent to the soil-structure interface, according to the effective seismic input–domain reduction method [Bielak and Christiano (1984)].

ID Card. Additional card for the ID keyword option.

Optional	1	2	3	4	5	6	7	8
Variable	ID	HEADING						
Type	I	A70						

Card Sets. Include as many pairs of Cards 1 and 2 as desired. This input ends at the next keyword ("*") card.

Node and set Cards. Card 1 for keyword options NODE or SET:

Card 1	1	2	3	4	5	6	7	8
Variable	SSID	typeID	GMX	GMY	GMZ			
Type	I	I	I	I	I			
Default	none	none	none	none	none			

Point Cards. Card 1 for keyword option POINT.

Card 1	1	2	3	4	5	6	7	8	9	10
Variable	SSID	XP		YP		ZP		GMX	GMY	GMZ
Type	I	F		F		F		I	I	I
Default	none	0.		0.		0.		none	none	none

Card 2	1	2	3	4	5	6	7	8
Variable	SF	CID	BIRTH	DEATH	ISG	IGM		
Type	F	I	F	F	I	I		
Default	1.	0	0.	10^{28}	0	0		

VARIABLE	DESCRIPTION
ID	Optional ID. This ID does not need to be unique.
HEADING	An optional descriptor for the given ID.
SSID	Soil-structure interface ID.
typeID	Node ID (NID in *NODE) or nodal set ID (SID in *SET_NODE).
XP	*x* coordinate of ground motion location on soil-structure interface.
YP	*y* coordinate of ground motion location on soil-structure interface.
ZP	*z* coordinate of ground motion location on soil-structure interface.
GMX	Acceleration load curve or ground motion ID for motion in the (local) *x*-direction.
GMY	Acceleration load curve or ground motion ID for motion in the (local) *y*-direction.
GMZ	Acceleration load curve or ground motion ID for motion in the (local) *z*-direction.

VARIABLE	DESCRIPTION
SF	Ground motion scale factor. (default = 1.0)
CID	Coordinate system ID, see *DEFINE_COORDINATE_SYSTEM.
BIRTH	Time at which specified ground motion is activated.
DEATH	Time at which specified ground motion is removed: EQ.0.0: default set to 10^{28}
ISG	Definition of soil-structure interface: EQ.0: SSID is ID for soil-structure interface defined by *INTERFACE_SSI_ID for non-matching mesh between soil and structure. EQ.1: SSID is segment set ID identifying soil-structure interface for merged meshes between soil and structure.
IGM	Specification of ground motions GMX, GMY, GMZ: EQ.0: ground motions are specified as acceleration load curves. See *DEFINE_CURVE EQ.1: Both ground accelerations and velocities specified using *DEFINE_GROUND_MOTION.

Remarks:

1. The ground motion at any node on a soil-structure interface is computed as follows:

   a) If the node coincides with a location where ground motion is specified, that ground motion is used for that node.

   b) If the node does not coincide with a location where ground motion is specified, the ground motion at that node along a particular degree-of-freedom is taken as a weighted average of all the ground motions specified on the interface along that degree-of-freedom, where the weights are inversely proportional to the distance of the node from the ground motion location.

2. Multiple ground motions specified at the same location are added together to obtain the resultant ground motion at that location.

3. Spatially-uniform ground motion may be specified on a soil-structure interface by specifying the ground motion at only one location on that interface. Specifying the ground motion at more than one point on a soil-structure interface results in spatially-varying ground motion on that interface.

*LOAD_SHELL_*OPTION1*_*{OPTION2}*

Available options for *OPTION1* include:

ELEMENT

SET

Available options for *OPTION2* include:

ID

If the ID is defined an additional card is required.

Purpose: Apply the distributed pressure load over one shell element or shell element set. The numbering of the shell nodal connectivities must follow the right hand rule with positive pressure acting in the negative t-direction. See Figure 27-4. This option applies to the three-dimensional shell elements only.

ID Card. Additional card for ID keyword option.

Optional	1	2	3	4	5	6	7	8
Variable	ID	HEADING						
Type	I	A70						

Shell Cards. Include as many of these cards as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	EID/ESID	LCID	SF	AT				
Type	I	I	F	F				
Default	none	none	1.	0.				
Remarks	1	1	2					

VARIABLE	DESCRIPTION
EID/ESID	Shell ID (EID) or shell set ID (ESID), see *ELEMENT_SHELL or *SET_SHELL.
LCID	Load curve ID, see *DEFINE_CURVE.
SF	Load curve scale factor
AT	Arrival time for pressure or birth time of pressure.

Remarks:

1. If LCID is input as -1, then the Brode function is used to determine the pressure for the segments, see also *LOAD_BRODE.
2. If LCID is input as -2, then the ConWep function is used to determine the pressure for the segments, see *LOAD_BLAST.
3. The load curve multipliers may be used to increase or decrease the pressure. The time value is not scaled.

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$   *LOAD_SHELL_ELEMENT
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  From a sheet metal forming example.  A blank is hit by a punch, a holder is
$  used to hold the blank on its sides.  All shells on the holder are given a
$  pressure boundary condition to clamp down on the blank.  The pressure
$  follows load curve 3, but is scaled by -1 so that it applies the load in the
$  correct direction.  The load starts at zero, but quickly rises to 5 MPa
$  after 0.001 sec.  (Units of this model are in: ton, mm, s, N, MPa, N-mm)
$
*LOAD_SHELL_ELEMENT
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      eid      lcid        sf        at
     30001         3 -1.00E+00       0.0
     30002         3 -1.00E+00       0.0
     30003         3 -1.00E+00       0.0
     30004         3 -1.00E+00       0.0
     30005         3 -1.00E+00       0.0
     30006         3 -1.00E+00       0.0
     30007         3 -1.00E+00       0.0
$
$  Note: Just a subset of all the shell elements of the holder is shown above,
$        in practice this list contained 448 shell element id's.
$
$
*DEFINE_CURVE
$     lcid      sidr      scla      sclo      offa      offo
         3
$
```

```
$           abscissa            ordinate
               0.000                 0.0
               0.001                 5.0
               0.150                 5.0
$
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

*LOAD_SPCFORC

Purpose: When used in a full-deck restart run, this card will apply the SPC constraint forces from the initial run on the corresponding degrees of freedom in the current run. This is useful when modeling unbounded domains using a non-reflecting boundary while incorporating static stresses computed in the initial run: the fixed constraints on the outer boundary in the initial static analysis are removed in the transient analysis and replaced by equivalent static forces.

While *BOUNDARY_NON_REFLECTING acts similarly if dynamic relaxation is used for the static analysis, this approach works for any method used to preload the model.

No parameters are necessary for this card.

*LOAD_SSA

Purpose: The Sub-Sea Analysis (SSA) capability allows a simple and efficient way of loading the structure to account for the effects of the primary shock wave and the subsequent bubble oscillations of an underwater explosion. It achieves its efficiency by approximating the pressure scattered by air and water-backed plates and the pressure transmitted through a water-back plate. The loading incorporates the plane wave approximation for direct shock response and the virtual mass approximation for bubble response. *LOAD_SSA does not implement a doubly asymptotic approximation of transient fluid-structure interaction.

Card 1	1	2	3	4	5	6	7	8
Variable	VS	DS	REFL	ZB	ZSURF	FPSID	PSID	NPTS
Type	F	F	F	F	F	I	I	I
Default	none	none	0.	0.	0.	0	0	1

Card Sets. Include as many pairs of Cards 1 and 2 as necessary. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	A	ALPHA	GAMMA	KTHETA	KAPPA			
Type	F	F	F	F	F			
Default	none	none	none	none	none			

Card 2	1	2	3	4	5	6	7	8
Variable	XS	YS	ZS	W	TDELY	RAD	CZ	
Type	F	F	F	F	F	F	F	
Default	none	none	none	none	none	none	none	

VARIABLE	DESCRIPTION
VS	Sound speed in fluid
DS	Density of fluid
REFL	Consider reflections from sea floor. EQ.0: off EQ.1: on
ZB	Z coordinate of sea floor if REFL = 1, otherwise, not used.
ZSURF	Z coordinate of sea surface
FPSID	Part set ID of parts subject to flood control. Use the *PART_SET_COLUMN option where the parameters A1 and A2 must be defined as follows: Parameter A1: Flooding status: EQ.1.0: Fluid on both sides. EQ.2.0: Fluid outside, air inside. EQ.3.0: Air outside, fluid inside. EQ.4.0: Material or part is ignored. Parameter A2: Tubular outer diameter of beam elements. For shell elements this input must be greater than zero for loading.
PSID	Part set IDs of parts defining the wet surface. The elements defining these parts must have their outward normals pointing into the fluid. See Figure 27-6. EQ.0: all parts are included. GT.0: the part set id.
NPTS	Number of integration points for computing pressure (1 or 4)
A	Shock pressure parameter
ALPHA	α, shock pressure parameter
GAMMA	γ, time constant parameter

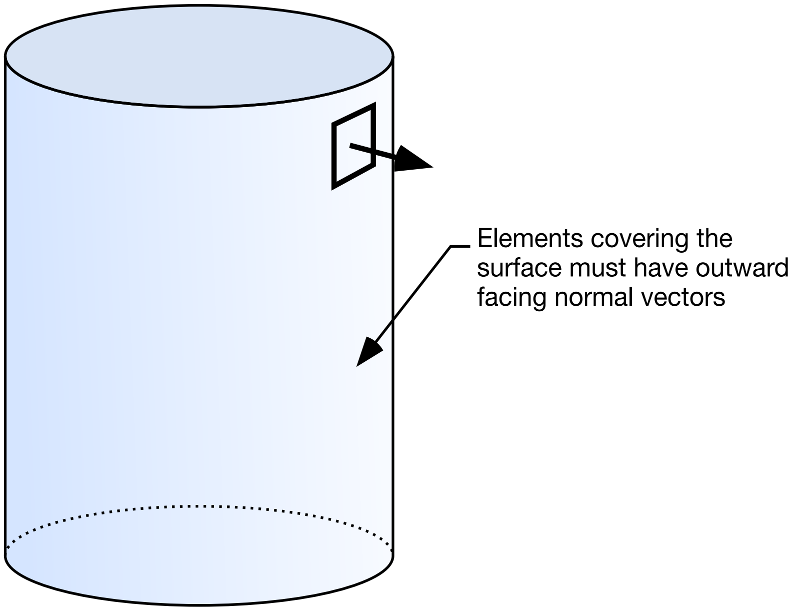


Figure 27-6. The shell elements interacting with the fluid must be numbered such that their outward normal vector points into the fluid media.

VARIABLE	DESCRIPTION
KTHETA	K_θ, time constant parameter
KAPPA	κ, ratio of specific heat capacities
XS	X coordinate of charge
YS	Y coordinate of charge
ZS	Z coordinate of charge
W	Weight of charge
TDELY	Time delay before charge detonates
RAD	Charge radius
CZ	Charge depth

Remarks:

1. SSA assumes the model is in MKS units. If it is in another system of units, *control_coupling should be used to account for the conversion.

2. The "flooding status" is instrumental in determining how the model parts are loaded. If A1 = 1, the front of the plate as defined by the outward normal is exposed to the incident pressure. The back of the plate is not exposed to the incident pressure but feels a transmitted pressure that resists plate motion. If A1 = 2, then the plating has fluid on the outside as determined by the outward normal. It is exposed to the incident pressure and feels the scattered pressure. No loading is applied to the back side. If A1 = 3, then air is on the front of the plate and water is on the back. Neither the front nor the back of the plate is exposed to the incident pressure, but the motion of the plate is resisted by pressure generated on the back of the plate when it moves. Transmitted pressures are assumed not to strike another plate.

3. The pressure history of the primary shockwave at a point in space through which a detonation wave passes is given as:

$$P(t) = P_m e^{\frac{-t}{\theta}}$$

where P_m and the time constant θ below are functions of the type and weight W of the explosive charge and the distance Q from the charge.

$$P_{peak} = A\left[\frac{W^{1/3}}{Q}\right]^{\alpha}$$

$$\theta = K_\theta W^{1/3}\left[\frac{W^{1/3}}{Q}\right]^{\gamma}$$

where A, α, γ, and K_θ are constants for the explosive being used.

*LOAD_STEADY_STATE_ROLLING

Steady state rolling analysis is a generalization of *LOAD_BODY, allowing the user to apply body loads to part sets due to translational and rotational accelerations in a manner that is more general than the *LOAD_BODY capability. *LOAD_STEADY_STATE_ROLLING may be invoked multiple times as long as no part has the command applied more than once. Furthermore, the command may be applied to arbitrary meshes, i.e., axisymmetric meshes are not required.

Card Sets. Include as many sets consisting of the following four cards as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	PSID						
Type	I	I						
Default	none	none						

Card 2	1	2	3	4	5	6	7	8
Variable	N1	N2	LCD1	LCD1R				
Type	I	I	I	I				
Default	0	0	0	0				

Card 3	1	2	3	4	5	6	7	8
Variable	N3	N4	LCD2	LCD2R				
Type 4	I	I	I	I				
Default	0	0	0	0				

Card 4	1	2	3	4	5	6	7	8
Variable	N5	N6	LCD3	LCD3R				
Type	I	I	I	I				
Default	0	0	0	0				

VARIABLE	DESCRIPTION
ID	ID
PSID	Part set ID
N1	Node 1 defining rotational axis
N2	Node 2 defining rotational axis
LCD1	Load curve defining angular velocity around rotational axis.
LCD1R	Optional load curve defining angular velocity around rotational axis for dynamic relaxation. LCD1 is used during dynamic relaxation if LCD1R is not defined.
N3	Node 3 defining turning axis
N4	Node 4 defining turning axis
LCD2	Load curve defining angular velocity around turning axis.
LCD2R	Optional load curve defining angular velocity around turning axis for dynamic relaxation. LCD2 is used during dynamic relaxation if LCD2R is not defined.
N5	Node 5 defining translational direction
N6	Node 6 defining translational direction
LCD3	Load curve defining translational velocity in translational direction.
LCD3R	Optional load curve defining translational velocity in translational direction. LCD3 is used during dynamic relaxation if LCD3R is not defined.

Remarks:

The steady state rolling capability adds inertial body loads in terms of a moving reference defined by the user input. The current coordinates are defined in terms of the displacement, u, and the moving reference frame, Y,

$$x_{SSR} = u + Y \quad \dot{x}_{SSR} = \dot{u} + \dot{Y} \quad \ddot{x}_{SSR} = \ddot{u} + \ddot{Y}$$
$$Y = R(\omega_2 t)[R(\omega_1 t)(X - X_O) - X_C] + Y_T(t)$$

where R is the rotation matrix obtained by integrating the appropriate angular velocity, the magnitude of the angular velocities ω_1 and ω_2 are defined by load curves LCD1 and LCD2 respectively, and the directions are defined by the current coordinates of the node pairs N1-N2 and N3-N4 (see Figure XX). The velocity corresponding to the translational term, $Y_T(t)$, is defined in magnitude by LCD3 and in direction by the node pair N5-N6. The initial coordinates of the nodes are X, X_O is the initial coordinate vector of node N1 and X_C is the initial coordinate vector of node N3. If data defining an angular velocity is not specified, the velocity is defaulted to zero, and R is the identity matrix. In a similar manner, if the translational velocity is not specified, it is defaulted to zero.

This capability is useful for initializing the stresses and velocity of tires during dynamic relaxation, and rolling processes in manufacturing. It is available for solid formulations 1, 2, 10, 13, and 15, and for shell formulations 2, 4, 5, 6, 16, 25, 26, and 27. It is not available for beams and tshells. It is available for implicit and explicit simulations and is invoked for dynamic relaxation by specifying that the load curves are used during dynamic relaxation. At the end of the dynamic relaxation, the velocities of the parts are set to $\dot{x}_{SSR}$and the remaining parts are initialized according to the input file.

Users must ensure that the appropriate load curves are turned on during the relaxation process, and if implicit dynamic relaxation is used, that sufficient constraints are applied during the initialization to remove any rigid body motion and that they are removed at the end of the dynamic relaxation. The implicit iteration convergence rate is often improved by adding the geometric stiffness matrix using *CONTROL_IMPLICIT_GENERAL. A consistent tangent matrix is available by using *CONTROL_IMPLICIT_GENERAL, and while it improves the convergence rate with problems with small strains, it is often unstable for problems with large strains. The *CONTROL_STEADY_STATE_ROLLING options should be used to ramp up the frictional forces to obtain smooth solutions and good convergence rates.

To obtain the free-rolling angular velocity, the tire should be first inflated, then brought into contact with the road while the frictional force is ramped up with a load curve and a large value of SCL_K specified in *CONTROL_STEADY_STATE_ROLLING. The angular velocity of the tire is then slowly varied over a range that covers the free rolling velocity. The free rolling velocity is obtained when either the frictional force in the direction of rolling or the moment about the tire axis is near zero. For a tire with an initial radius of R and a translational velocity of V, the approximate value for the free rolling value of the rolling velocity is $\omega = \frac{V}{(1+\varepsilon)R}$, where ε is the hoop strain of the rolling tire. For a first guess, the hoop strain can be set to 0.0, and the rolling velocity will be within 10% of the actual value. After the first calculation, a smaller range bracketing the free rolling velocity should be used in a second calculation to refine the free rolling velocity. An accurate value of the free rolling velocity is necessary for subsequent analyses, such as varying the slip angle of the tire.

A time varying slip angle can be specified by moving one of the nodes defining the direction vector of the translational velocity. To check that the stiffness scale factor in *CONTROL_STEADY_STATE_ROLLING is high enough, a complete cycle from a zero slip angle to a maximum value, then back to zero, should be performed. If the loading and unloading values are reasonably close, then the stiffness scale factor is adequate.

LOAD_STIFFEN_PART_*{OPTION}

Available options include:

<BLANK>

SET

Purpose: Staged construction. Available for solid, shell, and beam elements.

Note: This keyword card is available starting in release 3 of version 971.

Id Cards. Include as many of these cards as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID/PSID	LC	(blank)	STGA	STGR			
Type	I	I		I	I			
Default	none	0		0	0			

VARIABLE	DESCRIPTION
PID	Part ID (or Part Set ID for the_SET option)
LC	Load curve defining factor vs. time
STGA	Construction stage at which part is added (optional)
STGR	Construction stage at which part is removed (optional)

Remarks:

1. In many cases it is more convenient to use *DEFINE_STAGED_CONSTRUCTION_PART which automatically generates *LOAD_STIFFEN_PART data.

2. For parts that are initially present but are excavated (removed) during the analysis, the stiffness factor starts at 1.0. During the excavation time, it ramps down to a small value such as 1.0E-6. The excavation time should be sufficiently long to avoid introducing shock or dynamic effects. For parts that are introduced during the construction, e.g. retaining walls, the elements are initially present in the model but the factor is set to a low value such as 1.0e-6. During

the construction time the factor should be ramped up to 1.0. The construction time should be sufficiently long to avoid shock or dynamic effects. A factor that ramps up from 1.0E-6 to 1.0, then reduces back to 1.0E-6, can be used for temporary retaining walls, props, etc.

3. When the factor is increasing, it applies only to the stiffness and strength of the material in response to subsequent strain increments, not to any existing stresses.

4. When the factor is decreasing, it applies also to existing stresses as well as to the stiffness and strength.

5. This feature works with all material models when used only to reduce the stiffness (e.g. parts that are excavated, not parts that are added during construction). It works for most material types in all other cases, except those few materials that re-calculate stresses each time step from total strains (elastic, SOIL_BRICK, rubber models, orthotropic elastic, fabric, etc). There is no error check at present to detect STIFFEN_PART being used with an inappropriate material model. Symptoms of resulting problems would include non-physical large stresses when a part stiffens, due to the accumulated strains in the "dormant" material since the start of the analysis.

6. This feature is generally used with *LOAD_GRAVITY_PART. The same curve is often used for the stiffness factor and the gravity factor.

7. There are 3 methods of defining the factor-versus-time:

   a) LC overrides all the other methods if non-zero

   b) STGA, STGR refer to stages at which the part is added and removed – the stages are defined in *DEFINE_CONSTRUCTION_STAGES. If STGA is zero, the part has full stiffness at time zero. If not, it ramps up from the small factor FACT (on *CONTROL_STAGED_CONSTRUCTION) up to 1.0 over the ramp time at the start of stage STGA. If STGR is zero, the stiffness factor continues at 1.0 until the end of the analysis. If not, it ramps down from 1.0 to FACT over the ramp time at the start of stage STGR.

   c) *DEFINE_STAGED_CONSTRUCTION_PART can be used instead of *LOAD_STIFFEN_PART to define this loading. During initialization, a *LOAD_STIFFEN_PART card will be created and the effect is the same as using the STGA, STGR method described above.

*LOAD_SUPERPLASTIC_FORMING

Purpose: Perform superplastic forming (SPF) analysis. This option can be applied to 2D and 3D solid elements and to 3D shell elements, and has been implemented for both explicit and implicit analyses. The pressure loading controlled by the load curve ID given below is scaled to maintain a constant maximal strain rate or other target value.

This option must be used with material model 64, *MAT_RATE_SENSITIVE_POWER-LAW_PLASTICITY, for strain rate sensitive, powerlaw plasticity. For the output of data, see *DATA-BASE_SUPERPLASTIC_FORMING. Mass scaling is recommended in SPF applications.

New options to compute the target value with various averaging techniques and autojump options to control the simulation are implemented. Strain-rate speedup is also available. See Remarks 5-7 for details.

Card Sets. Include as many sets consisting of the following four cards as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	LCP1	CSP1	NCP1	LCP2	CSP2	NCP2	PCTS1	PCTS2
Type	I	I	F	I	I	F	F	F
Default	none	none	none.	none	none	none	100.0	100.0
Remarks				1	1	1		

Card 2	1	2	3	4	5	6	7	8
Variable	ERATE	SCMIN	SCMAX	NCYL	NTGT	LEVEL	TSRCH	AT
Type	F	F	F	I	I	I	F	F
Default	none	none	none	0	1	0	none	0.0
Remarks				2		5		

Card 3	1	2	3	4	5	6	7	8
Variable	TPEAK	TNEG	TOSC	POSC	PDROP	RILIM	RDLIM	STR
Type	F	F	F	F	F	F	F	F
Default	10.0	5.0	10.0	1.0	2.0	1.0	1.0	0.0
Remarks								6

Card 4	1	2	3	4	5	6	7	8
Variable	THRES	LOWER	UPPER	TFACT	NTFCT	BOX		
Type	F	F	F	F	I	I		
Default	5.0	90.0	99.0	1.0	10	0		
Remarks				7	7			

VARIABLE	DESCRIPTION
LCP1	Load curve number for Phase I pressure loading, (see *DEFINE_CURVE). The scaled version of this curve as calculated by LS-DYNA is output to "curve1". See Remark 3.
CSP1	Contact surface number to determine completion of Phase I.
NCP1	Percent of nodes in contact to terminate Phase I, (see *CONTACT_*OPTION*).
LCP2	Load curve number for Phase II pressure loading (reverse), (see *DEFINE_CURVE). The scaled version of this curve as calculated by LS-DYNA is output to "curve2". See Remark 3.
CSP2	Contact surface to determine completion of Phase II, (see *CONTACT_OPTION).
NCP2	Percent of nodes in contact to terminate Phase II.

VARIABLE	DESCRIPTION
PCTS1	Percentage of nodes-in-contact to active autojump in Phase I forming.
PCTS2	Percentage of nodes-in-contact to active autojump in Phase II forming.
ERATE	Desired target value. If it's strain rate, this is the time derivative of the logarithmic strain.
SCMIN	Minimum allowable value for load curve scale factor. To maintain a constant strain rate the pressure curve is scaled. In the case of a snap through buckling the pressure may be removed completely. By putting a value here the pressure will continue to act but at a value given by this scale factor multiplying the pressure curve.
SCMAX	Maximum allowable value for load curve scale factor. Generally, it is a good idea to put a value here to keep the pressure from going to unreasonable values after full contact has been attained. When full contact is achieved the strain rates will approach zero and pressure will go to infinity unless it is limited or the calculation terminates.
NCYL	Number of cycles for monotonic pressure after reversal.
NTGT	Type of the target (controlling) variable: EQ.1: strain rate. EQ.2: effective stress.
LEVEL	Criterion to compute averaged maximum of controlling variable: EQ.0: no average used. GE.1: averaging over neighbors of element with peak value of controlling variable. This parameter determines the level of neighbor search. EQ.-1: averaging over elements within selective range of peak controlling variable.
TSRCH	Time interval to conduct neighbors search.
AT	Time when SPF Phase I simulation starts.
TPEAK	Additional run time to terminate simulation when maximum pressure is reached.

VARIABLE **DESCRIPTION**

TNEG Additional run time to terminate simulation when percentage change of nodes-in-contact is zero or negative.

TOSC Additional run time to terminate simulation when percentage change of nodes-in-contact oscillates within a specific value.

POSC Percentage change to define the oscillation of percentage of nodes-in-contact.

PDROP Drop in percentage of nodes-in-contact from the maximum to terminate simulation after the specified termination percentage has been reached.

STR Autojump option or strike-through time (period of time without autojump check):

EQ.0: no autojump

EQ.-1: autojump controlled by peak pressure

EQ.-2: autojump controlled by percentage of nodes in contact

EQ.-3: autojump controlled by both above

GT.0: strike-through time, then same as STR = -3

THRES Threshold percentage that gives the threshold value above which elements are considered for average.

LOWER Lower percentile of elements above the threshold value to be included for average.

UPPER Upper percentile of elements above the threshold value to be included for average.

RILIM Maximum percentage change for pressure increment.

RDLIM Maximum percentage change for pressure decrement.

TFACT Strain rate speedup factor.

NTFCT Number of computing cycles to ramp up speedup.

BOX Box ID or box set ID. See Remark 8.

GT.0: box ID, see *DEFINE_BOX.

LT.0: |BOX| is box set ID, see *SET_BOX.

Remarks:

1. Optionally, a second phase can be defined. In this second phase a unique set of pressure segments must be defined whose pressure is controlled by load curve 2. During the first phase, the pressure segments of load curve 2 are inactive, and likewise, during the second phase the pressure segments of the first phase are inactive. When shell elements are used the complete set of pressure segments can be repeated in the input with a sign reversal used on the load curve. When solid elements are used the pressure segments for each phase will, in general, must be unique.

2. This is an ad hoc parameter which should probably not be used.

3. There are three output files "pressure", "curve1", and "curve2" from the SPF simulation, and they may be plotted using ASCII > superpl in LS-PrePost. The file "curve2" is created only if the second phase is active. The time interval for writing out these files is controlled by *DATABASE_SUPERPLASTIC_FORMING. The files "curve1" and "curve2" contain the adjusted pressure histories calculated by SPF solver. File "pressure" contains time histories of scaling factor, maximal target value, averaged target value, and percentage of contact.

4. The constraint method contact, *CONTACT_CONSTRAINT_NODES_TO_SURFACE, is recommended for superplastic forming simulations since the penalty methods are not as reliable when mass scaling is applied. Generally, in superplastic simulations mass scaling is used to enable the calculation to be carried out in real time.

5. In order to reduce the oscillation in pressure, the maximal target value used to adjust the pressure load is calculated by special averaging algorithm. There are two options available:

   a) *Averaging over neighbors of element with maximum target value*: In this method, the element that has the maximum strain rate or other controlling variable is stored in each cycle of the computation. The elements close to the element with the maximum value are searched and stored in an array. The averaged maximal target value is computed over the neighboring elements. The user can input an integer number to control the level of neighbor search, which will affect the total number of elements for average. Because the neighbor search is time consuming, the user can input a time interval to limit the occurrence of searching. The neighbor search is conducted only when the simulation time reaches the specified time or the element with maximum target value falls out of the array of neighbors.

   b) *Averaging over elements within selective range of target value:* In this method, all elements that have target value above a threshold value (a threshold percentage of maximum target value) are sorted according to their target value and the elements between the user specified lower percentile and upper percentile are selected to compute the average of the maximal target value.

6. The SPF simulation can be controlled by various autojump options. When autojump conditions are met, the SPF simulation will be either terminated or continued from phase I to phase II simulation. The autojump check can be held inactive by setting a strikethrough time. In this case the SPF simulation will continue for that period of time and only be interrupted when the percentage of nodes-in-contact reaches 100% for a specified time. The available autojump conditions are:

   a) *Peak pressure is reached and stays for certain time:* The peak pressure is determined by the maximum allowable scale factor and the load curve. The simulation will continue for a user specified time before termination.

   b) *User specified percentage of nodes-in-contact is reached:* The simulation will be terminated or continued to Phase II automatically if one of the following conditions is met:

      i) If the change of the percentage of nodes-in-contact is zero or negative for a specified time.

      ii) If the percentage of nodes-in-contact oscillates in a specified range for a specified time.

      iii) If the percentage of nodes-in-contact drops more than a specified value from the maximum value recorded.

      iv) If the percentage of nodes-in-contact reaches a user specified stop value.

7. In order to speed up the simulation of the superplastic forming process, we scale down the computation time. By doing this we increase the strain rate allowed in the SPF process, resulting in reduced simulation time. However, caution should be utilized with this speedup as it may affect the accuracy of the results. We recommend no or small strain rate speedup for simulations with complex geometry or tight angles.

8. If the user knows the area(s) in the workpiece that are critical in the SPF process, he can use the box option to limit the region(s) where the elements are checked for computing the average of the maximal target value.

*LOAD_SURFACE_STRESS_{OPTION}

Available options include:

<BLANK>

SET

Purpose: This keyword modifies the behavior of shell elements causing them to pass pressure-type loads to material models 37 and 125 (see Remarks 1 and 2); shells usually omit such effects.

With this keyword, LS-DYNA calculates segment pressures from contact and applied pressure loads on both the upper and lower surfaces of the shell and applies them as negative local *z*-stresses during the simulation. It is found in some cases this capability can improve the accuracy of metal forming simulations.

Card Sets. Include as many sets consisting of the following three cards as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID/PSID							
Type	I							

Card 2	1	2	3	4	5	6	7	8
Variable	LSCID1	LSCID2	LSCID3	LSCID4	LSCID5	LSCID6	LSCID7	LSCID8
Type	I	I	I	I	I	I	I	I

Card 3	1	2	3	4	5	6	7	8
Variable	USCID1	USCID2	USCID3	USCID4	USCID5	USCID6	USCID7	USCID8
Type	I	I	I	I	I	I	I	I

VARIABLE **DESCRIPTION**

PID/PSID

Part ID or if option set is active, part set ID.

LSCID*n*

Lower surface contact IDs. Up to eight IDs can be defined. These contacts definitions contribute to the pressure acting on the lower surface of the shell. If the pressure on the lower surface is due to applied pressure loads, specify a value -1 instead of a contact ID. Only one of the LSCID*n* may be set to -1.

Lower surface of a part is on the opposite side of the shell element normals, which must be made consistent (in *LS-PrePost*).

USCID*n*

Upper surface contact IDs. Up to eight IDs can be defined. These contacts definitions contribute to the pressure acting on the upper surface of the shell. . If the pressure on the upper surface is due to applied pressure loads, specify a value of -1 instead of a contact ID. Only one of the USCID*n* may be set to -1.

Upper surface of a part is on the same side of the shell element normals, which must be made consistent (in *LS-PrePost*).

Remarks:

1. **MAT_TRANSVERSLY_ANISOTROPIC_ELASTIC_PLASTIC (37).** This keyword can be used with *MAT_037 when ETAN is set to a negative value (see *MAT_037 manual pages), which triggers normal stresses (local *z*-stresses) resulting either from sliding contact or applied pressure to be considered in the shell formulation. The normal stresses can be significant in male die radius and corners in forming of Advanced High Strength Steels (AHSS). The negative local *z*-stresses are written to the d3plot files after Revision 97158, and can be plotted in *LS-PrePost* by selecting *z*-stress under *FCOMP* → *Stress* and select *local* under *FCOMP*.

2. **MAT_KINEMATIC_HARDENING_TRANSVERSLY_ANISOTROPIC (125).** This keyword can also be used in a simulation with *MAT_125 to account for the normal stresses (see *MAT_125 manual pages). For this material inserting this keyword *anywhere* in the input deck will invoke the shell normal stress calculation.

LOAD_THERMAL_*OPTION

Available options include:

CONSTANT

CONSTANT_ELEMENT_*OPTION*

CONSTANT_NODE

LOAD_CURVE

TOPAZ

VARIABLE

VARIABLE_ELEMENT_*OPTION*

VARIABLE_NODE

VARIABLE_SHELL_*OPTION*

Purpose: To define nodal temperatures that thermally load the structure. Nodal temperatures defined by the *LOAD_THERMAL_*OPTION* method are all applied in a structural only analysis. They are ignored in a thermal only or coupled thermal/structural analysis, see *CONTROL_THERMAL_*OPTION*.

All the *LOAD_THERMAL options cannot be used in conjunction with each other. Only those of the same thermal load type, as defined below in column 2, may be used together.

*LOAD_THERMAL_CONSTANT	- Thermal load type 1
*LOAD_THERMAL_ELEMENT	- Thermal load type 1
*LOAD_THERMAL_CONSTANT_NODE	- Thermal load type 1
*LOAD_THERMAL_LOAD_CURVE	- Thermal load type 2
*LOAD_THERMAL_TOPAZ	- Thermal load type 3
*LOAD_THERMAL_VARIABLE	- Thermal load type 4
*LOAD_THERMAL_VARIABLE_ELEMENT	- Thermal load type 4
*LOAD_THERMAL_VARIABLE_NODE	- Thermal load type 4
*LOAD_THERMAL_VARIABLE_SHELL	- Thermal load type 4

*LOAD_THERMAL_CONSTANT

Purpose: Define nodal sets giving the temperature that remains constant for the duration of the calculation. The reference temperature state is assumed to be a null state with this option. A nodal temperature state, read in above and held constant throughout the analysis, dynamically loads the structure. Thus, the temperature defined can also be seen as a relative temperature to a surrounding or initial temperature.

Card Sets. Include as many sets consisting of the following two cards as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	NSID	NSIDEX	BOXID					
Type	I	I	I					
Default	none	0.	0.					

Card 2	1	2	3	4	5	6	7	8
Variable	T	TE						
Type	F	F						
Default	0.	0.						

VARIABLE	DESCRIPTION
NSID	Nodal set ID containing nodes for initial temperature EQ.0: all nodes are included:
NSIDEX	Nodal set ID containing nodes that are exempted from the imposed temperature (optional).
BOXID	All nodes in box which belong to NSID are initialized. Others are excluded (optional).
T	Temperature

VARIABLE	DESCRIPTION
TE	Temperature of exempted nodes (optional)

LOAD_THERMAL_CONSTANT_ELEMENT_*OPTION

Available options include:

BEAM

SHELL

SOLID

TSHELL

Purpose: Define a uniform element temperature that remains constant for the duration of the calculation. The reference temperature state is assumed to be a null state. An element temperature, read in above and held constant throughout the analysis, dynamically loads the structure. The defined temperature can also be seen as a relative temperature to a surrounding or initial temperature.

Element Cards. Include as many cards in this format as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	EID	T						
Type	I	F						
Default	none	0.						

VARIABLE	DESCRIPTION
EID	Element ID
T	Temperature, see remark below.

Remarks:

1. The temperature range for the constitutive constants in the thermal materials must include the reference temperature of zero. If not termination will occur with a temperature out-of-range error immediately after the execution phase is entered.

*LOAD_THERMAL_CONSTANT_NODE

Purpose: Define nodal temperature that remains constant for the duration of the calculation. The reference temperature state is assumed to be a null state with this option. A nodal temperature state, read in above and held constant throughout the analysis, dynamically loads the structure. Thus, the temperature defined can also be seen as a relative temperature to a surrounding or initial temperature.

Node Cards. Include as many cards in this format as desired. This input ends at the next keyword ("*") card.

Card	1	2	3	4	5	6	7	8
Variable	NID	T						
Type	I	F						
Default	none	0.						

VARIABLE	DESCRIPTION
NID	Node ID
T	Temperature, see remark below.

Remarks:

1. The temperature range for the constitutive constants in the thermal materials must include the reference temperature of zero. If not termination will occur with a temperature out-of-range error immediately after the execution phase is entered.

*LOAD_THERMAL_D3PLOT

Purpose: Temperatures computed in a prior thermal-only analysis are used to load a mechanical-only analysis. The rootname of the d3plot database from the thermal-only analysis is specified on the execution line of the mechanical-only analysis using T = *tpf*, where *tpf* is that rootname, e.g., T = d3plot.

Warnings:

1. If using a double precision LS-DYNA executable in making the two runs, do not write the d3plot data using 32ieee format in the thermal-only run, i.e., the environment variable LSTC_BINARY must not be set.
2. The rootnames of the d3plot databases from the two runs must not conflict. Such conflict can be avoided, for example, by using jobid = *jobname* on the execution line of the second (mechanical-only) run.

*LOAD_THERMAL_LOAD_CURVE

Purpose: Nodal temperatures will be uniform throughout the model and will vary according to a load curve. The temperature at time = 0 becomes the reference temperature for the thermal material. The reference temperature is obtained from the optional curve for dynamic relaxation if this curve is used. The load curve option for dynamic relaxation is useful for initializing preloads.

Thermal Load Curve Cards. Include as many cards in this format as desired. This input ends at the next keyword ("*") card.

Card	1	2	3	4	5	6	7	8
Variable	LCID	LCIDDR						
Type	I	I						
Default	none	0						

VARIABLE	DESCRIPTION
LCID	Load curve ID, see *DEFINE_CURVE, to define temperature versus time.
LCIDDR	An optional load curve ID, see *DEFINE_CURVE, to define temperature versus time during the dynamic relaxation phase.

*LOAD_THERMAL_TOPAZ

Purpose: Nodal temperatures will be read in from the TOPAZ3D database. This file is defined on the execution line by the specification: T = tpf, where tpf is a binary database file (e.g., T3PLOT).

*LOAD_THERMAL_VARIABLE

Purpose: Define nodal temperature using node set(s) and temperature vs. time curve(s).

Card Sets. Include as many sets consisting of the following two cards as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	NSID	NSIDEX	BOXID					
Type	I	I	I					
Default	none	0.	0.					

Card 2	1	2	3	4	5	6	7	8
Variable	TS	TB	LCID	TSE	TBE	LCIDE	LCIDR	LCIDEDR
Type	F	F	I	F	F	I	I	I
Default	0.	0.	none	0.	0.	none	none	none
Remark	1	1	1	1	1			

VARIABLE	DESCRIPTION
NSID	Nodal set ID containing nodes (see *SET_NODE_*OPTION*): EQ.0: all nodes are included.
NSIDEX	Nodal set ID containing nodes that are exempted (optional), (see *SET_NODE_OPTION).
BOXID	All nodes in box which belong to NSID are initialized. Others are excluded.
TS	Scaled temperature.
TB	Base temperature.

VARIABLE	DESCRIPTION
LCID	Load curve ID that multiplies the scaled temperature, (see *DEFINE_CURVE).
TSE	Scaled temperature of the exempted nodes (optional).
TBE	Base temperature of the exempted nodes (optional).
LCIDE	Load curve ID that multiplies the scaled temperature of the exempted nodes (optional), (see *DEFINE_CURVE).
LCIDR	Load curve ID that multiplies the scaled temperature during the dynamic relaxation phase
LCIDEDR	Load curve ID that multiplies the scaled temperature of the exempted nodes (optional) during the dynamic relaxation phase.

Remarks:

1. The total temperature is defined as

$$T = \mathrm{TB} + \mathrm{TS} \times f(t)$$

where $f(t)$ is the current value of the load curve, TS is the scaled temperature, and TB is the base temperature.

The rate of thermal strain is based on the rate of temperature change. In other words, the thermal load arises from change in total temperature. Furthermore, the calculation of the thermal strain from the coefficient of thermal expansion depends on the material model, and some material models, e.g., *MAT_255, may offer multiple options. Temperature-dependent material properties are based on the total temperature.

LOAD_THERMAL_VARIABLE_BEAM_*{OPTION}

Available options include:

<BLANK>

SET

Purpose: Define a known temperature time history as a function of the section coordinates for beam elements. To set the temperature for the whole element see *LOAD_THERMAL_VARIABLE_ELEMENT_BEAM.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	EID/SID	IPOLAR					
Type	I	I	1					
Default	none	none	0					

Temperature Cards. Include as many cards in the following format as desired. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	TBASE	TSCALE	TCURVE	TCURDR	SCOOR	TCOOR		
Type	F	F	I	I	F	F		
Default	0	1.0	0	TCURVE	none	none		

VARIABLE **DESCRIPTION**

ID Load case ID

EID/SID Beam ID or beam set ID

IPOLAR GT.0: the coordinates SCOOR and TCOOR are given in polar coordinates (see notes)

TBASE Base temperature

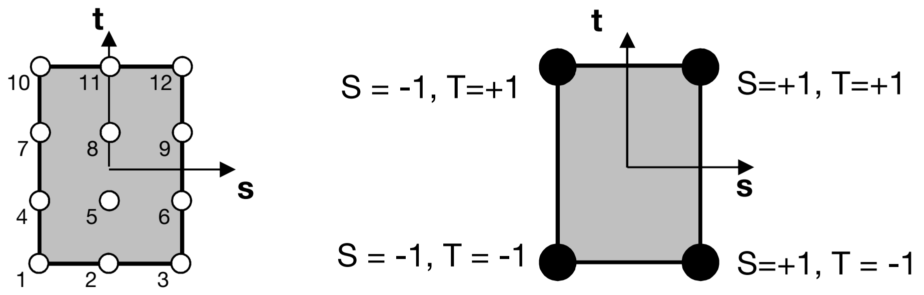


Figure 0-1. Figure illustrating point ordering.

VARIABLE	DESCRIPTION
TSCALE	Constant scale factor applied to temperature from curve
TCURVE	Curve ID for temperature vs. time
TCURDR	Curve ID for temperature vs. time used during dynamic relaxation
SCOOR	Normalized coordinate in local s-direction (-1.0 to +1.0)
TCOOR	Normalized coordinate in local t-direction (-1.0 to +1.0)

Remarks:

1. The temperature T is defined as:

$$\mathrm{T} = \mathrm{TBASE} + \mathrm{TSCALE} \times f(t)$$

   where $f(t)$ is the current ordinate value of the curve. If the curve is undefined, then T = TBASE at all times.

2. At least four points (four Card 2's) must be defined in a rectangular grid. The required order of the points is as shown in Figure 0-1. First, define the bottom row of points (most negative t), left to right in order of increasing s. Then increment *t* to define the next row of points, left-to-right in order of increasing s, and so on. The *s*-*t* axes are in the plane of the beam cross-section with the *s*-axis in the plane of nodes N1, N2, N3 defined in *ELEMENT_BEAM.

3. For the polar option, SCOOR is the non-dimensional radius R/R_0 where R_0 is the outer radius of the section; and TCOOR is defined as θ/π, where θ is the angle in radians from the s-axis, defined in the range $-\pi < \theta < \pi$.

4. Temperatures will be assigned to the integration points by linear interpolation from the points defined using this command.

LOAD_THERMAL_VARIABLE_ELEMENT_*OPTION

Available options include:

BEAM

SHELL

SOLID

TSHELL

Purpose: Define element temperature that is variable during the calculation. The reference temperature state is assumed to be the temperature at time = 0.0 with this option.

Element Cards. Include as many cards in the following format as desired. This input ends at the next keyword ("*") card.

Card	1	2	3	4	5	6	7	8
Variable	EID	TS	TB	LCID				
Type	I	F	F	I				
Default	none	0.	0.	none				

VARIABLE	DESCRIPTION
EID	Element ID
TS	Scaled temperature
TB	Base temperature
LCID	Load curve ID defining a scale factor that multiplies the scaled temperature as a function of time, (see *DEFINE_CURVE).

Remarks:

1. The temperature is defined as:

$$T = \mathrm{TB} + \mathrm{TS} \times f(t)$$

where $f(t)$ is the current value of the load curve, TS is the scaled temperature, and TB is the base temperature

*LOAD_THERMAL_VARIABLE_NODE

Purpose: Define nodal temperature that is variable during the calculation. The reference temperature state is assumed to be a null state with this option. A nodal temperature state read in and varied according to the load curve dynamically loads the structure. Thus, the defined temperatures are relative temperatures to an initial reference temperature.

Node Cards. Include as many cards in the following format as desired. This input ends at the next keyword ("*") card.

Card	1	2	3	4	5	6	7	8
Variable	NID	TS	TB	LCID				
Type	I	F	F	I				
Default	none	0.	0.	none				

VARIABLE	DESCRIPTION
NID	Node ID
TS	Scaled temperature
TB	Base temperature
LCID	Load curve ID that multiplies the scaled temperature, (see *DEFINE_CURVE).

Remarks:

1. The temperature is defined as:

$$T = \mathrm{TB} + \mathrm{TS} \times f(t)$$

where $f(t)$ is the current value of the load curve, TS is the scaled temperature, and TB is the base temperature

LOAD_THERMAL_VARIABLE_SHELL_*{OPTION}

Available options include:

<BLANK>

SET

Purpose: Define a known temperature time history as a function of the through-thickness coordinate for the shell elements.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	EID/SID						
Type	I	I						
Default	none	none						

Temperature Cards. Include as many cards of this type as desired. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	TBASE	TSCALE	TCURVE	TCURDR	ZCO			
Type	F	F	I	I	F			
Default	0	1.0	0	TCURVE	-1/+1			

VARIABLE	DESCRIPTION
ID	Load case ID
EID/SID	Shell ID or shell set ID
TBASE	Base temperature
TSCALE	Constant scale factor applied to temperature from curve
TCURVE	Curve ID for temperature vs. time

VARIABLE	DESCRIPTION
TCURDR	Curve ID for temperature vs. time used during dynamic relaxation
ZCO	Normalized through-thickness coordinate (-1.0 to +1.0)

Remarks:

1. The temperature T is defined as:

$$T = \text{TBASE} + \text{TSCALE} \times f(t)$$

where $f(t)$ is the current ordinate value of the curve. If the curve is undefined, then T = TBASE at all times.

2. Through-thickness points must be defined in order of increasing ZCO (-1.0 to +1.0). ZCO=+1.0 is the top surface of the element, i.e. the element surface in the positive outward normal vector direction from the mid-plane.

3. At least two points (two Card 2′s) must be defined. Temperatures will be assigned in the through-thickness direction by linear interpolation from the points defined using this command.

4. If the element has multiple in-plane integration points, the same temperature distribution is used at each in-plane integration point.

5. If a shell′s temperature distribution is defined using this card, any values defined by *LOAD_THERMAL_NODE are ignored for that shell.

*LOAD_VOLUME_LOSS

Purpose: To represent the effect of tunneling on surrounding structures, it is common to assume that a pre-defined fraction (e.g., 2%) of the volume occupied by the tunnel is lost during the construction process. This feature is available for solid elements only.

Part Set Cards. Include as many of these cards as desired. This input ends at the next keyword ("*") card.

Card	1	2	3	4	5	6	7	8
Variable	PSID	COORD	LCUR	FX	FY	FZ	PMIN	FACTOR
Type	I	I	I	F	F	F	F	F
Default	none	0	0	1	1	1	-1.e20	.01

VARIABLE	DESCRIPTION
PSID	Part Set ID
COORD	Coordinate System ID (default - global coordinate system)
LCUR	Curve ID containing volume fraction lost vs. time
FX	Fraction of strain occurring in *x*-direction
FY	Fraction of strain occurring in *y*-direction
FZ	Fraction of strain occurring in *z*-direction
PMIN	(Leave blank)
FACTOR	Feedback factor

Remarks:

Volume loss is modeled by a process similar to thermal contraction: if the material is unrestrained it will shrink while remaining unstressed; if restrained, stresses will become more tensile. Typically the material surrounding the tunnel offers partial restraint; the volume loss algorithm adjusts the applied "thermal" strains to attempt to achieve the desired volume loss. Optionally, FX, FY and FZ may be defined: these will be treated as ratios for the x, y and z strains; this feature can be used to prevent contraction parallel to the tunnel axis.

The total volume of all the parts in the part set is monitored and output at the time-history interval (on *DATABASE_BINARY_D3THDT) to a file named vloss_output. This file contains lines of data (*time, volume1, volume2, volume3…*) where *volume1* is the total volume of elements controlled by the first *LOAD_VOLUME_LOSS card, *volume2* is the total volume of elements controlled by the second *LOAD_VOLUME_LOSS card, etc.

This feature works only with material types that use incremental strains to compute stresses. Thus, hyperelastic materials (e.g. MAT_027) are excluded, as are certain foam material types (e.g. MAT_083).

The feedback factor (see FACTOR field) controls how strongly the algorithm tries to impose the desired change of volumetric strain. The default value is recommended. If the volumetric response appears noisy or unstable, it may be necessary to reduce FACTOR. Alternatively, if the actual volumetric strain changes much more slowly than the input curve, it may be necessary to increase FACTOR.

*MODULE

The keyword *MODULE provides a way to load user compiled libraries at runtime, to support user defined capabilities such as material models, equations of state, element formulations, etc. The following keywords implement this capability:

*MODULE_LOAD

*MODULE_PATH

*MODULE_USE

*MODULE_LOAD

Purpose: Load a dynamic library for user subroutines.

Card Sets. Repeat as many sets data cards as desired (cards 1 and 2) to load multiple libraries. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	MDLID		TITLE					
Type	A20		A60					
Default	none		none					

Card 2	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	C							
Default	none							

VARIABLE	DESCRIPTION
MDLID	Module identification. A unique string label must be specified.
TITLE	Description of the module.
FILENAME	File name of the library to be loaded, 80 characters maximum. If the filename has no path component, LS-DYNA will search in all directories specified in *MODULE_PATH first. If not found and the filename starts with "+" (a plus sign), LS-DYNA will continue to search all directories specified in the system environment variable LD_LIBRARY_PATH.

Remarks:

1. The MODULE capability described here and in *MODULE_USE is still under development and is considered experimental. As such, it is currently only supported in specially compiled executables on Linux systems

2. If only one dynamic library is loaded and no rules are required (*MODULE_USE), this dynamic library can be specified through the execution line option "module = *dll*". The system environment variable LD_LIBRARY_PATH is also used for searching the dynamic library if the filename starts with "+". This execution line option provides the support to the classic user subroutine subroutines without modifying the input deck.

MODULE_PATH_*{OPTION}

Available options:

<BLANK>

RELATIVE

LS-DYNA's default behavior is search for modules in the current working directory. If a module file is not found and the filename has no path component, LS-DYNA will search in all directories specified on the cards following a *MODULE_PATH keyword. Multiple paths can specified using one *MODULE_PATH keyword card, i.e.,

```
*MODULE_PATH
Directory_path1
Directory_path2
Directory_path3
```

Directory paths are read until the next "*" card is encountered. A directory path can have up to 236 characters. See Remark 3.

When the RELATIVE option is used, all directories are relative to the location of the input file. For example, if "i=/home/test/problems/input.k" is given on the command line, and the input contains

```
*MODULE_PATH_RELATIVE
lib
../lib
```

then the two directories /home/test/problems/lib and /home/test/lib will be searched for module files.

Remarks:

1. Filenames and pathnames are limited to 236 characters spread over up to three 80 character lines. When 2 or 3 lines are needed to specify the filename or pathname, end the preceding line with "␣+" (space followed by a plus sign) to signal that a continuation line follows. Note that the "␣+" combination is, itself, part of the 80 character line; hence the maximum number of allowed characters is 78 + 78 + 80 = 236.

*MODULE_USE

Purpose: Define the rules for mapping the user subroutines loaded in dynamic libraries to the model. The rules can be applied to:

*MAT_USER_DEFINED_MATERIAL_MODELS	(MAT 41 - 50)
*MAT_THERMAL_USER_DEFINED	(MAT T11 - T15)
*EOS_USER_DEFINED	(EOS 21 – 30)
*SECTION_SOLID	(ELFORM 101 - 105)
*SECTION_SHELL	(ELFORM 101- 105)

and other subroutines in the LS-DYNA user subroutine package.

LS-DYNA requires that subroutines in modules (see *MODULE_LOAD) be named as if they were part of the traditional user subroutine framework. Each module, however, can contain a complete set of those subroutines, and it is, therefore, possible to import in different modules the same subroutines of the same name several times. Each module is, essentially, an independent copy of the traditional user-subroutine framework. This keyword, *MODULE_USE, deals with namespace conflicts by defining how each subroutine in the module is presented to the other keywords.

The rules defined in *MODULE_USE are applied to only one dynamic library.

Card 1	1	2	3	4	5	6	7	8
Variable	MDLID							
Type	A20							
Default	none							

Rule Cards. Card 2 defines rules for the module specified in Card 1. Include one instance of this card for each subroutine to be mapped. If two rules conflict, new rules override existing rules. Input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	TYPE		PARAM1		PARAM2			
Type	A20		A20		A20			
Default	none		blank		blank			

VARIABLE	DESCRIPTION
MDLID	Module identification defined in *MODULE_LOAD.
TYPE	Rule type. See TYPE definitions below.
PARAM1	Type dependent parameter.
PARAM2	Type dependent parameter.

User Defined Materials:

TYPE	DESCRIPTION
UMAT	Implements the user material type PARAM1 in the model via a possibly different type PARAM2 in the dynamic library. Types in the extended range from 1001 to 2000 can be used as material types in the model. For example, if PARAM1 = 1001 and PARAM2 = 42, then model material 1001 will use subroutine umat42 from this library. If PARAM1 is blank and PARAM2 is not, then all user materials in the model will use the indicated subroutine from this library. If PARAM2 is blank and PARAM1 is not, the material type PARAM1 in the model will use the traditional subroutine from this library. For example, if PARAM1 = 43 and PARAM2 is blank, then material type 43 in the model will use the subroutine umat43 from this library If both PARAM1 and PARAM2 are blank, then all user defined

TYPE **DESCRIPTION**

materials in the model will use the traditional subroutines from this library (41 → umat41, 42 → umat42, etc).

MATID

Implements the user material having ID = PARAM1 via a possibly different material type PARAM2 in the dynamic library. PARAM2 may be blank if the material type defined in the model is the same as the one in the dynamic library.

PARAM1 can be a numerical id or label of the material as defined in the model.

Materials beyond user material models can be overloaded.

Thermal User Defined Materials:

TYPE **DESCRIPTION**

TUMAT

Implements the user thermal material type PARAM1 in the model (PARAM1 = 11-15) via type PARAM2 in the dynamic library. See type UMAT for the default rules.

TMATID

Implements the user thermal material having ID = PARAM1 via thermal material type PARAM2 in the dynamic library. PARAM2 may be blank if the material type defined in the model is the same as the one in the dynamic library.

PARAM1 can be a numerical id or label of the thermal material as defined in the model.

Materials beyond user material models can be overloaded.

User Defined EOS:

TYPE **DESCRIPTION**

UEOS

Implements the user EOS model PARAM1 in the model (PARAM1 = 21-30) via EOS model PARAM2 in the dynamic library. See type UMAT for the default rules.

User Defined Elements:

TYPE **DESCRIPTION**

UELEM

Implements the user element type PARAM1 in the model (PARAM1 = 101-105) via element type PARAM2 in the dynamic library. See type UMAT for the default rules.

TYPE	DESCRIPTION
SECTIONID	Implements the user element type with section ID = PARAM1 via element type PARAM2 in the dynamic library. PARAM2 may be blank if the element type defined in the model is the same as the one in the dynamic library.

> **Note**: Solid element types are mapped to subroutine usrsld, and shell element types are mapped to subroutine usrshl. These interfaces are documented in dyn21b.f.

Other User Subroutines:

TYPE	DESCRIPTION
	The following subroutines implemented in the dynamic library are used for the model. Any subroutines with the same name in other dynamic libraries, if they exist, are ignored.
UMATFAIL	matusr_24 matusr_103
UFRICTION	usrfrc
UWEAR	userwear
UADAP	useradap uadpnorm uadpval
UWELDFAIL	uweldfail uweldfail12 uweldfail22
USPH	hdot
UTHERMAL	usrhcon usrflux
ULOAD	ujntfdrv loadsetud loadud
UELEMFAILCTL	matfailusercontrol matuserfail

TYPE	DESCRIPTION
UMATFPERT	usermatfpert
UREBAR	rebar_bondslip_get_nhisvar rebar_bondslip_get_force
ULAGPOROUS	lagpor_getab_userdef
UAIRBAG	airusr user_inflator
UALE	al2rfn_criteria5 al2rmv_criteria5 alerfn_criteria5 alermv_criteria5 shlrfn_criteria5 shlrmv_criteria5 sldrfn_criteria5 sldrmv_criteria5 f3dm9ale_userdef1
UCOUPLE2OTHER	couple2other_boxminmax couple2other_comm couple2other_dt couple2other_extra couple2other_getf couple2other_givex couple2other_reader chkusercomm usercomm usercomm1
UOUTPUT	ushout

Remarks:

1. In order to simplify the development of user defined materials via modules, the *MODULE_USE keyword can be omitted in one special case. If only a single module library will be used and no remapping of material types is desired, then a *MODULE_LOAD keyword may appear with a single library and no other *MODULE keywords. In this case, all the user defined subroutines found in this library will be used in the normal way. For example, if the library contains umat41 and umat43, those routines will be used for all materials of type 41 and 43 respectively, and the *MODULE_USE keyword describing this may be omitted.

EXAMPLES:

The following examples demonstrate the input for the MODULE keywords:

*MODULE_PATH

*MODULE_LOAD

*MODULE_USE

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Using *MODULE_PATH to define additional directories where
$  dynamic libraries are saved.
$
$------------------------------------------------------------------------------
$
*MODULE_PATH
/home/lsdynauser/lib
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Using *MODULE_LOAD to load all dynamic libraries for this model.
$
$  three libraries are loaded here for demonstration:
$
$  M_LIB: my own library which is under development.  A debug version is built
$         in my local directory.
$         it contains: UMAT41, UMAT42, and UMAT45
$
$  LIB_A: a hypoelastic model, provided by a third party, for shell & solid.
$          it contains UMAT41 only, with an optional FLUID
$
$  LIB_B: contains two material models provided by another company, with
$         UMAT41: a elasto-plastic model
$         UMAT45: a hyper-elastic model for rubber
$
$------------------------------------------------------------------------------
*MODULE_LOAD
$...>....1...>....2...>....3...>....4...>....5...>....6...>....7...>....8
M_LIB               My own library
/my_development_path/mylib_r123_dbg.so
LIB_A               library from company A
Lib_hypoelastic.so
LIB_B               library from company B
Lib_plastic.so
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Using *MODULE_USE to map to user subroutines
$
$  CASE 1:
$
$  M_LIB is used for UMAT41,UMAT42,UMAT45 in the model, as default
$  UMAT41 in LIB_A is used for MT=1001 in the model for shell, and MT=1002 for solid
$  UMAT45 in LIB_B is used for MATID=202, which also happens to hvae MT=1002
$
*MODULE_USE
M_LIB
$...>....1...>....2...>....3...>....4...>....5...>....6...>....7...>....8
UMAT
```

```
*MODULE_USE
LIB_A
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
UMAT                1001                41
UMAT                1002                41
*MODULE_USE
LIB_B
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
MATID               202                 45
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$  CASE 2:
$
$  M_LIB is used for UMAT41,UMAT42 as,
$         MATID=101 with MT=41
$         MATID=102 with MT=42
$  UMAT45 in LIB_B is used with different material properties, as,
$         MATID=201 with MT=1001
$         MATID=202 with MT=1002
$         MATID=203 with MT=1003
$
*MODULE_USE
LIB_B
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
UMAT                                    45
*MODULE_USE
M_LIB
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
MATID               101
MATID               102
```

*NODE

The keywords defined in this section include:

NODE_*{OPTION}

*NODE_MERGE_SET

*NODE_MERGE_TOLERANCE

*NODE_RIGID_SURFACE

NODE_SCALAR_*{OPTION}

*NODE_THICKNESS

*NODE_TO_TARGET_VECTOR

*NODE_TRANSFORM

NODE_*{OPTION}

Available options include:

<BLANK>

MERGE

Purpose: Define a node and its coordinates in the global coordinate system. Also, the boundary conditions in global directions can be specified. Generally, nodes are assigned to elements; however, exceptions are possible, see remark 2 below. The nodal point ID must be unique relative to other nodes defined in the *NODE section.

The MERGE option is usually applied to boundary nodes on disjoint parts and only applies to nodes defined when the merge option is invoked. With this option, nodes with identical coordinates are replaced during the input phase by the first node encountered that shares the coordinate. During the merging process a tolerance is used to determine whether a node should be merged. This tolerance can be defined using the keyword *NODE_MERGE_TOLERANCE keyword, which is recommended over the default value. See the *NODE_MERGE_TOLERANCE input description in the next section.

Node Cards. Include as many cards in the following format as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8	9	10
Variable	NID	X		Y		Z		TC	RC	
Type	I	F		F		F		F	F	
Default	none	0.		0.		0.		0.	0.	
Remarks								1	1	

VARIABLE	DESCRIPTION
NID	Node number
X	x coordinate
Y	y coordinate
Z	z coordinate

VARIABLE **DESCRIPTION**

TC Translational Constraint:

EQ.0: no constraints,

EQ.1: constrained x displacement,

EQ.2: constrained y displacement,

EQ.3: constrained z displacement,

EQ.4: constrained x and y displacements,

EQ.5: constrained y and z displacements,

EQ.6: constrained z and x displacements,

EQ.7: constrained x, y, and z displacements.

RC Rotational constraint:

EQ.0: no constraints,

EQ.1: constrained x rotation,

EQ.2: constrained y rotation,

EQ.3: constrained z rotation,

EQ.4: constrained x and y rotations,

EQ.5: constrained y and z rotations,

EQ.6: constrained z and x rotations,

EQ.7: constrained x, y, and z rotations.

Remarks:

1. Boundary conditions can also be defined on nodal points in a local (or global) system by using the keyword *BOUNDARY_SPC. For other possibilities also see the *CONSTRAINED keyword section of the manual.

2. A node without an element or a mass attached to it will be assigned a very small amount of mass and rotary inertia. Generally, massless nodes should not cause any problems but in rare cases may create stability problems if these massless nodes interact with the structure. Warning messages are printed when massless nodes are found. Also, massless nodes are used with rigid bodies to place joints, see *CONSTRAINED_EXTRA_NODES_*OPTION* and *CONSTRAINED_NODAL_RIGID_BODY.

*NODE_MERGE_SET

Purpose: The MERGE_SET option is applied to a set of boundary nodes on disjoint part. With this option, nodes with identical coordinates that are members of any node set ID defined by this keyword are replaced during the input phase by one node within the set or sets. Of the nodes sharing the same coordinates, the node chosen is the one with the smallest ID. During the merging process a tolerance is used to determine whether a node should be merged. This tolerance can be defined using the keyword *NODE_MERGE_TOLERANCE keyword, which is recommended over the default value. See the *NODE_MERGE_TOLERANCE input description in the next section. Only nodes contained within the specified sets will be merged. Nodes contained within the set are defined by the *NODE keyword. With this option, the keyword *NODE_MERGE is not needed.

Node Set Cards. Include as many cards as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	NSID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
NSID	Node set ID containing list of nodes to be considered for merging.

*NODE_MERGE_TOLERANCE

Purpose: Define a tolerance is determine whether a node should be merged for the keyword, *NODE_MERGE.

Card 1	1	2	3	4	5	6	7	8
Variable	TOLR							
Type	F							
Default	yes							

VARIABLE	DESCRIPTION
TOLR	Physical distance used to determine whether to merge a nodal pair of nearby nodes. See remark below.

Remarks:

If the tolerance, TOLR, is undefined or if it is defaulted to zero, a value is computed as:

$$\mathrm{TOLR} = 10^{-5} \cdot \frac{\mathrm{XMAX} + \mathrm{YMAX} + \mathrm{ZMAX} - \mathrm{XMIN} - \mathrm{YMIN} - \mathrm{ZMIN}}{3 \times \sqrt[3]{\mathrm{NUMNP}}}$$

where XMIN, XMAX, YMIN,YMAX, ZMIN, and ZMAX represent the minimum and maximum values of the (x,y,z) nodal point coordinates in the global coordinate system, and NUMNP is the number of nodal points.

*NODE_RIGID_SURFACE

Purpose: Define a rigid node and its coordinates in the global coordinate system. These nodes are used to define rigid road surfaces and they have no degrees of freedom. The nodal points are used in the definition of the segments that define the rigid surface. See *CONTACT_RIGID_SURFACE. The nodal point ID must be unique relative to other nodes defined in the *NODE section.

Node Cards. Include as many cards in the following format as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8	9	10
Variable	NID	X		Y		Z				
Type	I	F		F		F				
Default	none	0.		0.		0.				
Remarks										

VARIABLE	DESCRIPTION
NID	Node number
X	x coordinate
Y	y coordinate
Z	z coordinate

*NODE_SCALAR_{OPTION}

Available options include:

<BLANK>

VALUE

Purpose: Define a scalar nodal point which has one degree-of-freedom. The scalar point ID must be unique relative to other nodes defined in the *NODE section.

Node Card. Card 1 for no keyword option (option set to <BLANK>). Include as many cards in the following format as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8	9	10
Variable	NID	NDOF								
Type	I	I								
Default	none	0								

Node Card. Card 1 for the VALUE keyword option. Include as many cards in the following format as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7
Variable	NID	X1	X2	X3	NDOF		
Type	I	F	F	F	I		
Default	none	0	0	0	0		

VARIABLE	DESCRIPTION
NID	Scalar node ID.

VARIABLE	DESCRIPTION
NDOF	Number of degrees-of-freedom EQ.0: fully constrained EQ.1: one degree-of-freedom EQ.2: two degrees-of-freedom EQ.3: three degrees-of-freedom
XI	Initial value of Ith degree of freedom.

NODE_THICKNESS_*{OPTION1}_{OPTION2}

For OPTION1 the available options include:

<BLANK>

SET

For OPTION2 the available options include:

<BLANK>

GENERATE

Purpose: Define nodal thickness that overrides nodal thickness otherwise determined via *SECTION_SHELL, *PART_COMPOSITE, or *ELEMENT_SHELL_THICKNESS. The option GENERATE generates a linear thickness distribution between a starting node (or node set) and a ending node (or node set).

Card 1	1	2	3	4	5	6	7	8
Variable	ID1	THK	ID2	INC				
Type	I	F	I	I				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
ID1	Node ID, or node set ID if SET option is active. If GENERATE option is active, ID1 serves as the starting node (or node set).
THK	Thickness at node ID1, or node set ID1 if SET option is active (ignored if GENERATE option is active).
ID2	Ending node (or node set) if GENERATE option is active.
INC	Increment in node numbers if GENERATE option is active.

Remarks:

When the GENERATE option is active, both the starting and ending nodes (or node sets) must have a nodal thickness as defined by *NODE_THICKNESS or NODE_-

THICKNESS_SET. The sample commands shown below create a linear thickness distribution between node set 100 and node set 200.

```
*SET_NODE_LIST
100
1, 15, 39
*SET_NODE_LIST
200
7, 21, 45
*NODE_THICKNESS_SET
$ assign thickness of 2.0 to node 1, 15 and 39
100, 2.0
*NODE_THICKNESS_SET
$ assign thickness of 5.0 to node 7, 21 and 45
200, 5.0
*NODE_THICKNESS_SET_GENERATE
$ assign thickness of 3.  (= 2.+1.) to node 3 (=1+2), 17 (=15+2) and 41
(=39+2)
$ assign thickness of 4.  (= 2.+2.) to node 5 (=1+4), 19 (=15+4) and 43
(=39+4)
100,, 200, 2
```

*NODE_TO_TARGET_VECTOR

Purpose: Calculate vector components of the normal distance from the target to a node of a part best fitted to the target. This keyword is generated from *CONTROL_FORMING_BESTFIT_VECTOR.

Node Cards. Include as many cards in the following format as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8	9	10
Variable	NID	XDELTA		YDELTA		ZDELTA				
Type	I	F		F		F				
Default	none	0.		0.		0.				

VARIABLE	DESCRIPTION
NID	Node ID on a part best fitted to the target.
XDELTA	Difference in X-coordinates of the normal distance from the target (typically scan data) to a node of a part best fitted to the target.
YDELTA	Difference in Y-coordinates of the normal distance from the target (typically scan data) to a node of a part best fitted to the target.
ZDELTA	Difference in Z-coordinates of the normal distance from the target (typically scan data) to a node of a part best fitted to the target.

Remarks:

1. This keyword is automatically generated from *CONTROL_FORMING_BESTFIT_VECTOR and is available starting from Revision 112655.

*NODE_TRANSFORM

Purpose: Perform a transformation on a node set based on a transformation defined by the keyword *DEFINE_TRANSFORMATION.

Transformation Cards. Include as many cards in the following format as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	TRSID	NSID						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
TRSID	The ID of the transformation defined under *DEFINE_TRANS-FOR-MATION.
NSID	Node set ID of the set of nodes to be subject to the transformation.

*PARAMETER

The *PARAMETER family of commands assign numerical values or expressions to named parameters. The parameter names can be used subsequently in the input in place of numerical values.

PARAMETER_*OPTION

*PARAMETER_DUPLICATION

*PARAMETER_EXPRESSION

*PARAMETER_TYPE

PARAMETER_*{OPTION}_{OPTION}

The available options are

<BLANK>

LOCAL

MUTABLE

Purpose: Define the numerical values of parameter names referenced throughout the input file. The parameter definitions, if used, should be placed at the beginning of the input file following *KEYWORD or at the beginning of an include file if the LOCAL option is specified.

Parameter Cards. Include as many cards as necessary.

Card 1	1	2	3	4	5	6	7	8
Variable	PRMR1	VAL1	PRMR2	VAL2	PRMR3	VAL3	PRMR4	VAL4
Type	A	I, F or C	A	I, F or C	A	I, F or C	A	I, F or C
Default	none	none	none	none	none	none	none	none

VARIABLE **DESCRIPTION**

PRMR*n*

PRMR*n* sets both the *n*th parameter and its storage type.

$$\text{PRMR} = \text{T}\underbrace{\text{xxxxxxxxx}}_{\text{9 character name}}$$

The first character, “T”, is decoded as follows:

T.EQ.“R”: Parameter is a real number

T.EQ.“I”: Parameter is an integer

T.EQ.“C”: Parameter is a character

The remaining 9 characters specifiy the name of the parameter. A parameter name “time” (case insensitive) is disallowed.

For example, to define a shell thickness named, "SHLTHK", the input “RSHLTHK”, “R␣␣␣SHLTHK”, or “R␣␣SHLTHK␣” are all equivalent 10 character strings (“␣” is space). For instructions regard how to use the variable “SHLTHK” see Remark 1.

VARIABLE	DESCRIPTION
VAL*n*	Define the value of the *n*th parameter as either a real or integer number, or a character string consistent with preceding definition for PRMR*n*.

Remarks:

1. **Syntax for Using Parameters.** Parameters can be referenced anywhere in the input by placing an "&" immediately preceding the parameter name. If a minus sign "-" is placed directly before "&", i.e., "-&", with no space the sign of the numerical value will be switched.

2. **LOCAL Option.** *PARAMETER_LOCAL behaves like the *PARAMETER keyword with one difference. A parameter defined by *PARAMETER without the LOCAL option is visible and available at any later point in the input processing. Parameters defined via the LOCAL versions disappear when the input parser finishes reading the file in which they appear. LOCAL variables can temporarily mask non-LOCAL variables.

   For example, suppose you have the following input files:

   main.k:

   ```
   *PARAMETER
   R  VAL1    1.0
   *PARAMETER
   R  VAL2    2.0
   *PARAMETER
   R  VAL3    3.0
   *CONTROL_TERMINATION
      &VAL1
   *INCLUDE
   file1
   ```

   file1:

   ```
   *PARAMETER
   R  VAL1    10.0
   *PARAMETER_LOCAL
   R  VAL2    20.0
   *PARAMETER_LOCAL
   R  VAL4    40.0
   *INCLUDE
   file2
   ⋮
   ```

Then, inside file2 we will see VAL1 = 10.0, VAL2 = 20.0, VAL3 = 3.0 and VAL4 = 40.0. In main.k, after returning from file1, we will see VAL1 = 10.0, VAL2 = 2.0, and VAL3 = 3.0. VAL4 will not exist. This allows for include files that can set all their own parameters without clobbering the parameters in the rest of the input.

3. **MUTABLE Option for Redefining.** The MUTABLE option is used to indicate that an integer or real parameter may be redefined at some later point in the input processing (it is ignored for character parameters). Redefinition is allowed regardless of the setting of *PARAMETER_DUPLICATION. The MUTABLE qualifier must appear on the first definition of the parameter. It is not required on any later redefinition.

*PARAMETER_DUPLICATION

Purpose: The purpose is to control how the code behaves if a duplicate parameter definition is found in the input.

Card 1	1	2	3	4	5	6	7	8
Variable	DFLAG							
Type	I							
Default	1							

VARIABLE **DESCRIPTION**

DFLAG Flag to control treatment of duplicate parameter definitions:

EQ.1: issue a warning and ignore the new definition (default)

EQ.2: issue a warning and accept the new definition

EQ.3: issue an error and ignore (terminates at end of input)

EQ.4: accept silently

EQ.5: ignore silently

Remarks:

A LOCAL variable appearing in a file, which masks a non-LOCAL parameter, won't trigger these actions; however, a LOCAL that masks another LOCAL or a non-LOCAL that masks a non-LOCAL will

Only one *PARAMETER_DUPLICATION card is allowed. If more than one is found, a warning is issued and any after the first are ignored.

PARAMETER_EXPRESSION_*{OPTION}

The available options are

<BLANK>

LOCAL

MUTABLE

Purpose: Define the numerical values of parameter names referenced throughout the input file. Like the *PARAMETER keyword, but allows for general algebraic expressions, not simply fixed values. The LOCAL option allows for include files to contain their own unique expressions without clobbering the expressions in the rest of the input. See the *PARAMETER keyword.

Parameter Cards. Include as many cards as necessary.

Card 1	1	2	3	4	5	6	7	8
Variable	PRMR1	EXPRESSION1						
Type	A	A						
Default	none	none						

VARIABLE	DESCRIPTION
PRMR*n*	Define the nth parameter in a field of 10. Within this field the first character must be either an "R" for a real number, "I" for an integer, or "C" for a character string. Lower or upper case for "I/C/R" is okay. Following the type designation, define the name of the parameter using up to, but not exceeding nine characters. For example, when defining a shell thickness named, "SHLTHK", both inputs "RSHLTHK" or "R SHLTHK" can be used and placed anywhere in the field of 10. When referencing SHLTHK in the input file see Remark 1 below.
EXPRESSION*n*	General expression which is evaluated, having the result stored in PRMR*n*. The following functions are available: sin, cos, tan, csc, sec, ctn, asin, acos, atan, atan2, sinh, cosh, tanh, asinh, acosh, atanh, min, max, sqrt, mod, abs, sign, int, aint, nint, anint, float, exp, log, log10, float, pi, and general arithmetic expressions involving +, -, *, /, and **.

VARIABLE	DESCRIPTION
	The standard rules regarding operator precedence are obeyed, and nested parentheses are allowed. The expression can reference previously defined parameters (with or without the leading &). The expression can be continued on multiple lines simply by leaving the first 10 characters of the continuation line blank. For type “C” parameters, the expression is not evaluated in any sense, just stored as a string.

Remarks:

1. Parameters can be referenced anywhere in the input by placing an "&" immediately preceding the parameter name. Expressions can be included in the input when placed between brackets “<>” as long as the total line length does not exceed 80 columns and fields are comma-delimited. For example, this…

```
*parameter
rterm, 0.2, istates,  80
*parameter_expression
rplot,term/(states-30)
*DATABASE_BINARY_D3PLOT
&plot
```

   is equivalent to

```
*parameter
rterm, 0.2, istates,  80
*DATABASE_BINARY_D3PLOT
<term/(states-30)>,
```

2. The integer and real properties of constants and parameters are honored when evaluating expressions. So 2/5 becomes 0, but 2.0/5 becomes 0.4.

3. The sign, atan2, min, max, and mod functions all take two arguments. The others all take only 1.

4. Functions that use an angle as their argument, e.g., sin or cos, assume the angle is in radians.

5. The MUTABLE option is used to indicate that an integer or real parameter may be redefined at some later point in the input processing (it is ignored for charac-

ter parameters). Redefinition is allowed regardless of the setting of *PARAMETER_DUPLICATION. The MUTABLE qualifier must appear on the first definition of the parameter. It is not required on any later redefinition.

6. The unary minus has higher precedence than exponentiation, i.e. the formula -3**2 will be interpreted as $(-3)^2 = 9$.

*PARAMETER_TYPE

*PARAMETER_TYPE is a variation on the *PARAMETER keyword command. In addition to its basic function of associating a parameter name (PRMR) with a numerical value (VAL), the *PARAMETER_TYPE command also includes information (PRTYP) about how the parameter is used by LS-DYNA, e.g., as a Part ID or as a segment set ID.

*PARAMETER_TYPE is useful only when (1) the parameter is used to represent an integer ID number, and (2) LS-PrePost is used to combine two or more models (keyword decks) into a larger model.

Only by knowing how the parameter is used by LS-DYNA is LS-PrePost able to increment the parameter value by the proper "offset" when LS-PrePost combines two or more input decks together into a larger deck. These offsets are necessary so that IDs of a certain type, e.g., Part IDs, are not duplicated in the assembled model. Figure 30-1 (b) shows the offset input dialog box of LS-PrePost where offset values for specific ID types are assigned.

Background:

This command is designed to support workflows involving models that are built up from discrete subassemblies created by independent workgroups. As the subassembly models evolve through the design process, Part IDs, material IDs, etc. in the models may change with each design iteration and therefore it is advantageous to parameterize those IDs. In this way, though the parameter values may change, the parameter names remain the same. When the subassembly models are combined by LS-PrePost to create a larger model of an assembly or of a complete system, for instance, an aircraft engine model, parameter values assigned using *PARAMETER_TYPE are incremented by the proper ID offset value as prescribed when LS-PrePost imports each keyword file.

Card Format:

Parameter Cards. For each parameter with type information, include an additional card. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PRMR	VAL	PRTYP					
Type	A	I	A					
Default	none	None	none					

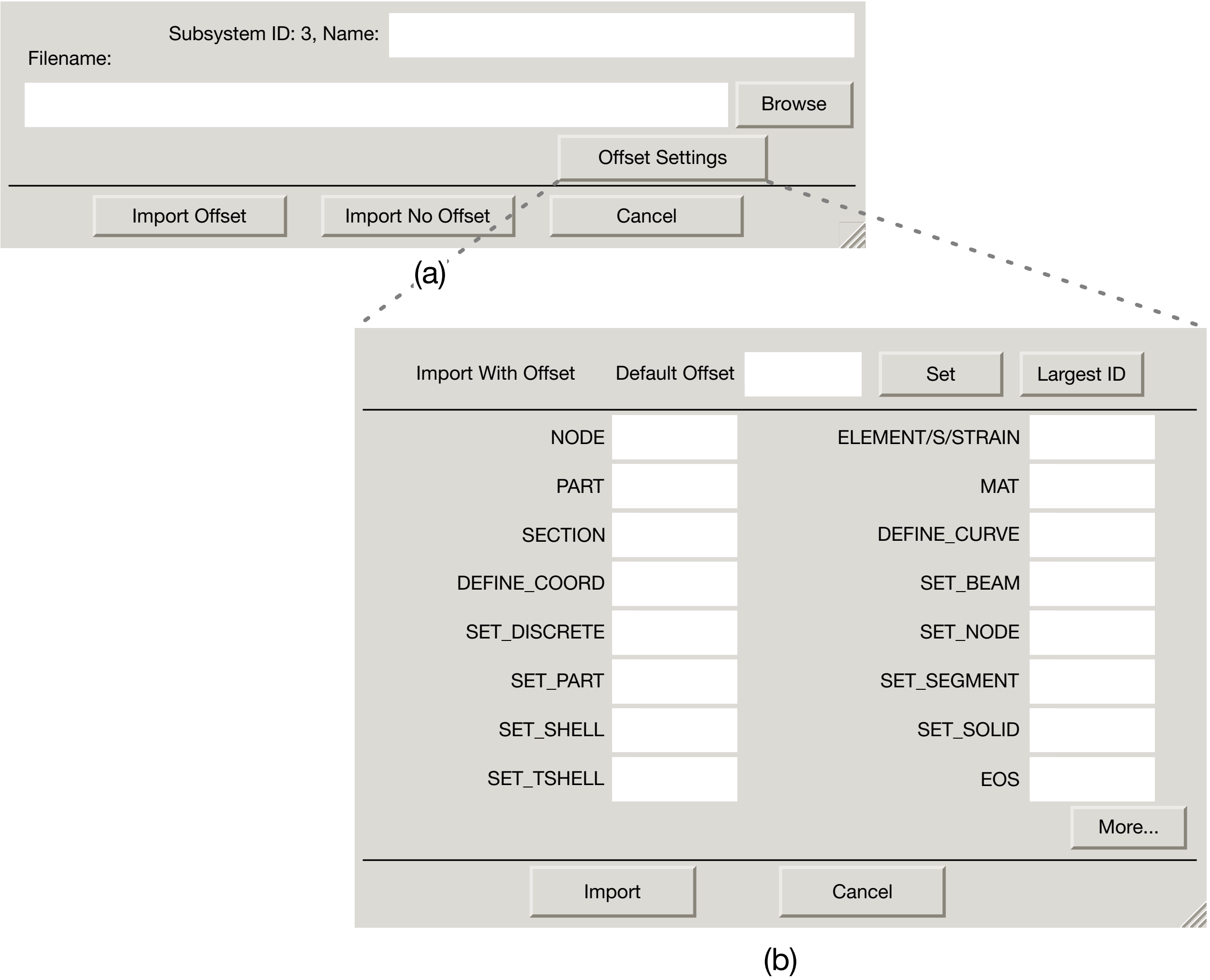


Figure 30-1. (a) the file → import → keyword dialog box; (b) the LS-PrePost dialog that takes the offset as a function of ID type.

VARIABLE **DESCRIPTION**

PRMR

PRMR must be in the following format

$$\text{PRMR} = \texttt{I}\,\underbrace{\text{xxxxxxxxx}}_{\text{9 character name}}$$

The first character is the type indicator and must be set to "I" for integer. The remaining 9 characters specifiy the name of the parameter.

For example, to define a part ID "WHLPID", the input "IWHLPID", "I␣␣␣WHLPID", or "I␣␣WHLPID␣" are all equivalent 10 character strings ("␣" is space). For instructions regard how to use the variable "WHLPID" see remark 1.

VAL

Define the value of the parameter. The VAL field must contain an integer.

VARIABLE **DESCRIPTION**

PRTYP

Describes, for the benefit of LS-PrePost only, how the parameter PRMR is used by LS-DYNA. PRTYP is ignored by LS-DYNA. For example, if VAL represents a Part ID, then PRTYP should be set to “PID”. Knowing how the parameter is used by LS-DYNA, LS-PrePost can apply the appropriate offset to VAL when input decks are combined using LS-PrePost.

EQ.NID:	Node ID,
EQ.NSID:	Node set ID,
EQ.PID:	Part ID,
EQ.PSID:	Part set ID,
EQ.MID:	Material ID,
EQ.EOSID:	Equation of state ID,
EQ.BEAMID:	Beam element ID,
EQ.BEAMSID:	Beam element set ID,
EQ.SHELLID:	Shell element ID,
EQ.SHELLSID:	Shell element set ID,
EQ.SOLIDID:	Solid element ID,
EQ.SOLIDSID:	Solid element set ID,
EQ.TSHELLID:	Tshell element ID,
EQ.TSHELLSID:	Tshell element set ID,
EQ.SSID:	Segment set ID

Remarks:

1. Parameters can be referenced anywhere in the input by placing an "&" at the first column of its field followed by the name of the parameter without blanks. For example if PRMR is set to “I␣␣WHLPID␣” then the appropriate reference is “&WHLPID”.

   Example:

```
*PARAMETER_TYPE
I WHLPID  100       PID
I WHLMID  300       MID
*PART
Wheel
&WHLPID,200,&WHLMID
```

2. *PARAMETER_TYPE is only supported by LS-PrePost 4.1 or later.

3. Combining *INCLUDE_TRANSFORM with *PARAMETER_TYPE is unsupported. This will introduce conflicting parameter offset values, and offset values specified in *INCLUDE_TRANSFORM will override offset values associated with *PARAMETER_TYPE.

*PART

The following keywords are used in this section:

PART_*{OPTION1}_{OPTION2}_{OPTION3}_{OPTION4}_{OPTION5}

*PART_ADAPTIVE_FAILURE

*PART_ANNEAL

PART_COMPOSITE_*{OPTION}

*PART_DUPLICATE

*PART_MODES

*PART_MOVE

*PART_SENSOR

*PART_STACKED_ELEMENTS

***PART**_*{OPTION1}_{OPTION2}_{OPTION3}_{OPTION4}_{OPTION5}*

For *OPTION1* the available options are

<BLANK>

INERTIA

REPOSITION

For *OPTION2* the available options are

<BLANK>

CONTACT

For *OPTION3* the available options are

<BLANK>

PRINT

For *OPTION4* the available options are

<BLANK>

ATTACHMENT_NODES

For *OPTION5* the available options are

<BLANK>

AVERAGED

Options 1, 2, 3, 4, and 5 may be specified in any order on the *PART card.

Purpose: Define parts, i.e., combine material information, section properties, hourglass type, thermal properties, and a flag for part adaptivity.

The INERTIA option allows the inertial properties and initial conditions to be defined rather than calculated from the finite element mesh. This applies to rigid bodies, see *MAT_RIGID, only. The REPOSITION option applies to deformable materials and is used to reposition deformable materials attached to rigid dummy components whose motion is controlled by either CAL3D or MADYMO. At the beginning of the calculation each component controlled by CAL3D/MADYMO is automatically repositioned to be consistent with the CAL3D/MADYMO input. However, deformable materials attached to these components will not be repositioned unless this option is used.

The CONTACT option allows part based contact parameters to be used with the automatic contact types a3, 4, a5, b5, a10, 13, a13, 15 and 26, that is

*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE,

*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_MORTAR,

*CONTACT_SINGLE_SURFACE,

*CONTACT_AUTOMATIC_NODES_TO_SURFACE,

*CONTACT_AUTOMATIC_BEAMS_TO_SURFACE,

*CONTACT_AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE,

*CONTACT_AUTOMATIC_SINGLE_SURFACE,

*CONTACT_AUTOMATIC_SINGLE_SURFACE_MORTAR,

*CONTACT_AIRBAG_SINGLE_SURFACE,

*CONTACT_ERODING_SINGLE_SURFACE,

*CONTACT_AUTOMATIC_GENERAL.

The default values to use for these contact parameters can be specified on the *CONTACT input section card.

The PRINT option allows user control over whether output data is written into the ASCII files MATSUM and RBDOUT. See *DATABASE_ASCII.

The AVERAGED option may be applied *only* to parts consisting of a single (non-branching) line of truss elements. The average strain and strain rate over the length of the truss elements in the *part* is calculated, and the resulting average axial force is applied to all of the elements in the part. Truss elements in an averaged part form one long continuous “macro-element.” The time step size for an AVERAGED part is based on the total length of the assembled trusses, rather than on the shortest truss.

Effectively, the truss elements of an AVERAGED part behave as a string under uniform tension. In an AVERAGED part there are no internal forces acting to keep the nodes separated, and other force contributions from the surrounding system *must* play that role. Therefore, the nodes connected to the truss elements should be attached to other structural members. This model is prototypically used for modeling cables in mechanical actuators. The AVERAGED option can be activated for all material types, which are available for truss elements.

Card Sets. Repeat as many sets data cards as desired (card 1 through 10). This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	HEADING							
Type	C							
Default	none							

Card 2	1	2	3	4	5	6	7	8
Variable	PID	SECID	MID	EOSID	HGID	GRAV	ADPOPT	TMID
Type	I/A	I or A10	I or A10	I or A10	I or A10	I	I	I or A10
Default	none	none	none	0	0	0	0	0

Inertia Card 1. Additional Card for the INERTIA option. See Remarks 2, 3, and 4.

Card 3	1	2	3	4	5	6	7	8
Variable	XC	YC	ZC	TM	IRCS	NODEID		
Type	F	F	F	F	I	I		

Inertia Card 2. Additional Card for the INERTIA option.

Card 4	1	2	3	4	5	6	7	8
Variable	IXX	IXY	IXZ	IYY	IYZ	IZZ		
Type	F	F	F	F	F	F		

Inertia Card 3. Additional Card for the INERTIA option.

Card 5	1	2	3	4	5	6	7	8
Variable	VTX	VTY	VTZ	VRX	VRY	VRZ		
Type	F	F	F	F	F	F		

Inertial Coordinate System Card. Optional card required for IRCS = 1 with INERTIA option. Define two local vectors or a local coordinate system ID.

Card 6	1	2	3	4	5	6	7	8
Variable	XL	YL	ZL	XLIP	YLIP	ZLIP	CID	
Type	F	F	F	F	F	F	I	
Remark	1	1	1	1	1	1	none	

Reposition Card. An additional Card is for the REPOSITION option.

Card 7	1	2	3	4	5	6	7	8
Variable	CMSN	MDEP	MOVOPT					
Type	I	I	I					

Contact Card. Additional Card is required for the CONTACT option.

Card 8	1	2	3	4	5	6	7	8
Variable	FS	FD	DC	VC	OPTT	SFT	SSF	CPARM8
Type	F	F	F	F	F	F	F	F

NOTE: If FS, FD, DC, and VC are specified they will not be used unless FS is set to a negative value (-1.0) in the *CONTACT section. These frictional coefficients apply only to contact types:

SINGLE_SURFACE,
AUTOMATIC_GENERAL,
AUTOMATIC_SINGLE_SURFACE,
AUTOMATIC_SINGLE_SURFACE_MORTAR,
AUTOMATIC_NODES_TO_...,
AUTOMATIC_SURFACE_...,
AUTOMATIC_SURFACE_..._MORTAR,
AUTOMATIC_ONE_WAY_...,
ERODING_SINGLE_SURFACE

Default values are input via *CONTROL_CONTACT input.

Print Card. An additional Card is required for the PRINT option. This option applies to rigid bodies and provides a way to turn off ASCII output in files rbdout and matsum.

Card 9	1	2	3	4	5	6	7	8
Variable	PRBF							
Type	I							

Attachment Nodes Card. Additional Card required for the ATTACHMENT_NODES option. See Remark 8.

Card 10	1	2	3	4	5	6	7	8
Variable	ANSID							
Type	I							

VARIABLE	DESCRIPTION
HEADING	Heading for the part
PID	Part identification. A unique number or label must be specified.

VARIABLE **DESCRIPTION**

SECID

Section identification defined in a *SECTION keyword. See Remark 7.

MID

Material identification defined in the *MAT section. See Remark 7.

EOSID

Equation of state identification defined in the *EOS section. Nonzero only for solid elements using an equation of state to compute pressure. See Remark 7.

HGID

Hourglass/bulk viscosity identification defined in the *HOURGLASS Section. See Remark 7.

EQ.0: default values are used.

GRAV

Flag to turn on gravity initialization according to *LOAD_DENSITY_DEPTH.

EQ.0: Part will be initialized only if included in the part set PSID in *LOAD_DENSITY_DEPTH.

EQ.1: Part will be initialized irrespective of PSID in *LOAD_DENSITY_DEPTH.

ADPOPT

Indicate if this part is adapted or not. (See also *CONTROL_ADAPTIVE):

LT.0: *r*-adaptive remeshing for 2-D solids, |ADOPT| gives the load curve ID that defines the element size as a function of time.

EQ.0: Adaptive remeshing is inactive for this part ID.

EQ.1: *h*-adaptive for 3-D shells.

EQ.2: *r*-adaptive remeshing for 2-D solids, 3-D tetrahedrons and
3-D EFG.

EQ.3: Axisymmetric *r*-adaptive remeshing for 3-D solid (see Remark 6).

EQ.9: Passive *h*-adaptive for 3-D shells. The elements in this part will not be split unless their neighboring elements in other parts need to be split more than one level.

VARIABLE	DESCRIPTION
TMID	Thermal material property identification defined in the *MAT_-THERMAL Section. Thermal properties must be specified for all solid, shell, and thick shell parts if a thermal or coupled thermal structural/analysis is being performed. Discrete elements are not considered in thermal analyses. See Remark 7.
XC	Global x-coordinate of center of mass. If nodal point, NODEID, is defined XC, YC, and ZC are ignored and the coordinates of the nodal point, NODEID, are taken as the center of mass.
YC	Global y-coordinate of center of mass
ZC	Global z-coordinate of center of mass
TM	Translational mass
IRCS	Flag for inertia tensor reference coordinate system: EQ.0: global inertia tensor, EQ.1: local inertia tensor is given in a system defined by the orientation vectors.
NODEID	Nodal point defining the CG of the rigid body. This node should be included as an extra node for the rigid body; however, this is not a requirement. If this node is free, its motion will not be updated to correspond with the rigid body after the calculation begins.
IXX	I_{xx}, xx component of inertia tensor (see Remark 4)
IXY	I_{xy},, xy component of inertia tensor (see Remark 4)
IXZ	I_{xz}, xz component of inertia tensor (see Remark 4)
IYY	I_{yy}, yy component of inertia tensor (see Remark 4)
IYZ	I_{yz}, yz component of inertia tensor (see Remark 4)
IZZ	I_{zz}, zz component of inertia tensor (see Remark 4)
VTX	initial translational velocity of rigid body in global x direction (see Remark 5)

VARIABLE	DESCRIPTION
VTY	initial translational velocity of rigid body in global y direction (see Remark 5)
VTZ	initial translational velocity of rigid body in global z direction (see Remark 5)
VRX	initial rotational velocity of rigid body about global x axis (see Remark 5)
VRY	initial rotational velocity of rigid body about global y axis (see Remark 5)
VRZ	initial rotational velocity of rigid body about global z axis (see Remark 5)
XL	x-coordinate of local x-axis. Origin lies at (0, 0, 0).
YL	y-coordinate of local x-axis
ZL	z-coordinate of local x-axis
XLIP	x-coordinate of vector in local x-y plane
YLIP	y-coordinate of vector in local x-y plane
ZLIP	z-coordinate of vector in local x-y plane
CID	Local coordinate system ID, see *DEFINE_COORDINATE_… With this option leave fields 1 - 6 blank.
CMSN	CAL3D segment number / MADYMO system number. See the numbering in the corresponding program.
MDEP	MADYMO ellipse/plane number: GT.0: ellipse number, EQ.0: default, LT.0: absolute value is plane number.

VARIABLE **DESCRIPTION**

MOVOPT

Flag to deactivate moving for merged rigid bodies, see *CONSTRAINED_RIGID_BODIES. This option allows a merged rigid body to be fixed in space while the nodes and elements of the generated CAL3D/MADYMO parts are repositioned:

EQ.0: merged rigid body is repositioned,

EQ.1: merged rigid body is not repositioned.

FS

Static coefficient of friction. The functional coefficient is assumed to be dependent on the relative velocity $v_{\mathrm{rel}}$of the surfaces in contact,

$$\mu_c = \mathrm{FD} + (\mathrm{FS} - \mathrm{FD})e^{-\mathrm{DC}\times|v_{\mathrm{rel}}|}.$$

For mortar contact $\mu_c = \mathrm{FS}$, i.e., dynamic effects are ignored.

FD

Dynamic coefficient of friction. The functional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact

$$\mu_c = \mathrm{FD} + (\mathrm{FS} - \mathrm{FD})e^{-\mathrm{DC}\times|v_{rel}|}.$$

For mortar contact $\mu_c = \mathrm{FS}$, i.e., dynamic effects are ignored.

DC

Exponential decay coefficient. The functional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact

$$\mu_c = \mathrm{FD} + (\ \mathrm{FS} - \mathrm{FD})e^{-\mathrm{DC}\times|v_{\mathrm{rel}}|}.$$

For mortar contact $\mu_c = \mathrm{FS}$ (dynamical effects are ignored).

VC

Coefficient for viscous friction. This is necessary to limit the friction force to a maximum. A limiting force is computed by,

$$F_{\mathrm{lim}} = \mathrm{VC} \times A_{\mathrm{cont}},$$

where A_{cont} is the area of the segment contacted by the node in contact. The suggested value for VC is to use the yield stress in shear $\mathrm{VC} = \frac{\sigma_o}{\sqrt{3}}$ where σ_0 is the yield stress of the contacted material.

OPTT

Optional contact thickness. This applies to solids, shells and beams.

VARIABLE **DESCRIPTION**

SFT

Optional thickness scale factor for PART ID in automatic contact (scales true thickness). This option applies only to contact with shell elements. True thickness is the element thickness of the shell elements.

SSF

Scale factor on default slave penalty stiffness for this PART ID whenever it appears in the contact definition. If zero, SSF is taken as unity.

CPARM8

Flag to exclude beam-to-beam contact from the same PID for CONTACT_AUTOMATIC_GENERAL. This applies only to MPP. Global default may be set using CPARM8 on *CONTACT_…_MPP Optional Card.

EQ.0: Flag is not set (default).

EQ.1: Flag is set.

EQ.2: Flag is set. CPARM8 = 2 has the additional effect of permitting contact treatment of spot weld (type 9) beams in AUTOMATIC_GENERAL contacts; spot weld beams are otherwise disregarded entirely by AUTOMATIC_-GENERAL contacts.

PRBF

Print flag for rbdout and matsum files.

EQ.0: default is taken from the keyword *CONTROL_OUTPUT.

EQ.1: write data into rbdout file only

EQ.2: write data into matsum file only

EQ.3: do not write data into rbdout and matsum

VARIABLE **DESCRIPTION**

ANSID

Attachment node set ID. See Remark 8. This option should be used very cautiously and applies only to rigid bodies. The attachment point nodes are updated each cycle whereas other nodes in the rigid body are updated only in the output databases. All loads seen by the rigid body must be applied through this nodal subset or directly to the center of gravity of the rigid body. If the rigid body is in contact this set must include all interacting nodes.

EQ.0: All nodal updates are skipped for this rigid body. The null option can be used if the rigid body is fixed in space or if the rigid body does not interact with other parts, e.g., the rigid body is only used for some visual purpose.

Remarks:

1. **Local Inertia Tensor Coordinate System.** The local Cartesian coordinate system is defined as described in *DEFINE_COORDINATE_VECTOR. The local *z*-axis vector is the vector cross product of the *x*-axis and the in-plane vector. The local *y*-axis vector is finally computed as the vector cross product of the *z*-axis vector and the *x*-axis vector. The local coordinate system defined by CID has the advantage that the local system can be defined by nodes in the rigid body which makes repositioning of the rigid body in a preprocessor much easier since the local system moves with the nodal points.

2. **Inertia Option and Shared Rigid/Deformable Nodes.** When specifying mass properties for a rigid body using the inertia option, the mass contributions of deformable bodies to nodes which are shared by the rigid body should be considered as part of the rigid body.

3. **Inertia Option Lacks Default Values.** If the inertia option is used, all mass and inertia properties of the body *must* be specified. *There are no default values.*

4. **Inertia Tensor Characteristics.** The inertia terms are always with respect to the center of mass of the rigid body. The reference coordinate system defines the orientation of the axes, not the origin. Note that the off-diagonal terms of the inertia tensor are opposite in sign from the products of inertia.

5. **Initial Velocity Card for Rigid Bodies.** The initial velocity of the rigid body may be overwritten by the *INITIAL_VELOCITY card.

6. **Axisymmetric Remeshing.** Axisymmetric remeshing is specially for 3-D orbital forming. The adaptive part using this option needs to meet the following requirement in both geometry and discretization:

   a) The geometry is (quasi-) symmetric with respect to the local *z*-axis, which in turn must be parallel to the global *z*-axis. See CID in *CONTROL_-REMESHING.

   b) A set of 2-D cross-sections with uniform angular interval around *z*-axis are discretized by mixed triangular and quadrilateral elements in a similar pattern.

   c) A set of circular lines around *z*-axis pass through the nodes of the cross-sections and form orbital pentahedrons and hexahedrons.

7. **Allowed ID Values.** The variables SID, MID, EOSID, HGID, and TMID in *PART, and in *SECTION, *MAT, *EOS, *HOURGLASS, and *MAT_THERMAL, respectively, may be input as an 10-character alphanumeric variable (20-characters if long format is used), e.g., “HS Steel”, or as an integer not to exceed $2^{32} - 1$, e.g.,“123456789” is allowed.

8. **Attachment Nodes Option.** All nodes are treated as attachment nodes if this option is not used. Attachment nodes apply to rigid bodies only. The motion of these nodes, which must belong to the rigid body, are updated each cycle. Other nodes in the rigid body are updated only for output purposes. Include all nodes in the attachment node set which interact with the structure through joints, contact, merged nodes, applied nodal point loads, and applied pressure. Include all nodes in the attachment node set if their displacements, accelerations, and velocities are to be written into an ASCII output file. Body force loads are applied to the c.g. of the rigid body.

*PART_ADAPTIVE_FAILURE

Purpose: This is an option for two-dimensional adaptivity to allow a part that is singly connected to split into two parts. This option is under development and will be generalized in the future to allow the splitting of parts that are multiply connected.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	T						
Type	I	F						

VARIABLE	DESCRIPTION
PID	Part ID
T	Thickness. When the thickness of the part reaches this minimum value the part is split into two parts. *The value for T should be on the order of the element thickness of a typical element.*

*PART_ANNEAL

Available options include:

<BLANK>

SET

Purpose: To initialize the stress states at integration points within a specified part to zero at a given time during the calculation. This option is valid for parts that use constitutive models where the stress is incrementally updated. This option applies to the Hughes-Liu beam elements, the integrated shell elements, thick shell elements, and solid elements. In addition to the stress tensor components, the effective plastic strain is also set to zero.

Part Cards. Include as many parts cards as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID/PSID	TIME						
Type	I	F						
Default	none	none						

VARIABLE	DESCRIPTION
PID/PSID	Part ID or part set ID if the SET option is active.
TIME	Time when the stress states are reinitialized.

PART_COMPOSITE_*{OPTION}

Available options include:

<BLANK>

CONTACT

TSHELL

LONG

Purpose: The following input provides a simplified method of defining a composite material model for shell elements and thick shell elements that eliminates the need for user defined integration rules and part ID's for each composite layer. When *PART_COMPOSITE is used, a section definition, *SECTION_SHELL or *SECTION_TSHELL, and integration rule definition, *INTEGRATION_SHELL, are unnecessary.

The material ID, thickness, material angle and thermal material ID for each through-thickness integration point of a composite shell or thick shell are provided in the input for this command. The total number of integration points is determined by the number of entries on these cards.

Unless the *ELEMENT_SHELL_THICKNESS card is set, the thickness is assumed to be constant on each shell element. The thickness, then, is given by summing the THICK*i* values from card 4 below over the integration points *i*. When the *ELEMENT_SHELL_THICKNESS card is included the THICK*i* values are scaled to fit the nodal thickness values assigned using the *ELEMENT__SHELL_THICKNESS keyword. For thick shells, the total thickness is obtained from the positions of the nodes on the top and bottom surfaces. In this case, the THICK*i*, are also scaled to conform to the geometry defined by the element's nodes.

For a more general method of defining composite shells and thick shells, see *ELEMENT_SHELL_COMPOSITE and *ELEMENT_TSHELL_COMPOSITE. These commands permit unique layer stack-ups for each element without requiring a unique part ID for each element.

With *PART_COMPOSITE, two integration points with 4 constants each are provided in each Integration Point Properties Card. On the other hand, with *PART_COMPOSITE_LONG, for each integration point there is one Integration Point Properties Card containing up to 8 constants.

To maintain a direct association of through-thickness integration point numbers with physical plies in the case where plies span over more than one part ID, see Remark 5.

The CONTACT option allows part based contact parameters to be used with the automatic contact types a3, 4, a5, a10, 13, a13, 15 and 26, which are listed under the *PART definition above.

Card Sets. Repeat as many sets data cards as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	HEADING							
Type	C							
Default	none							

Thin Shell Card. The following card is required for thin shell composites. Omit this card if the TSHELL option is used.

Card 2	1	2	3	4	5	6	7	8
Variable	PID	ELFORM	SHRF	NLOC	MAREA	HGID	ADPOPT	THSHEL
Type	I	I	F	F	F	I	I	I
Default	none	0	1.0	0.0	0.0	0	0	0

Thick Shell Card. This is an additional card for the TSHELL option.

Card 2	1	2	3	4	5	6	7	8
Variable	PID	ELFORM	SHRF			HGID		TSHEAR
Type	I	I	F			I		I
Default	none	0	1.0			0		0

Contact Card. Additional Card is required for the CONTACT option.

Card 3	1	2	3	4	5	6	7	8
Variable	FS	FD	DC	VC	OPTT	SFT	SSF	
Type	F	F	F	F	F	F	F	

NOTE: If FS, FD, DC, and VC are specified they will not be used unless FS is set to a negative value (-1.0) in the *CONTACT section. These frictional coefficients apply only to contact types:

SINGLE_SURFACE,
AUTOMATIC_GENERAL,
AUTOMATIC_SINGLE_SURFACE,
AUTOMATIC_NODES_TO_...,
AUTOMATIC_SURFACE_...,
AUTOMATIC_ONE_WAY_...,
ERODING_SINGLE_SURFACE

Default values are input via *CONTROL_CONTACT input.

Integration Point Data Cards without Long Option. The material ID, thickness, and material angle for each through-thickness integration point of a composite shell are provided below (up to two integration points per card). The integration point data should be given sequentially starting with the bottommost integration point. The total number of integration points is determined by the number of entries on these cards. Include as many cards as necessary. The next "*" card terminates this input.

Card 4	1	2	3	4	5	6	7	8
Variable	MID1	THICK1	B1	TMID1	MID2	THICK2	B2	TMID2
Type	I	F	F	I	I	F	F	I

Integration Point Data Cards for Long Option. The material ID, thickness, and material angle for each through-thickness integration point of a composite shell are provided below (one integration point per card). The integration point data should be given sequentially starting with the bottommost integration point. The total number of integration points is determined by the number of entries on these cards. Include as many cards as necessary. The next "*" card terminates this input.

Card 4	1	2	3	4	5	6	7	8
Variable	MID1	THICK1	B1	TMID1	PLYID1	SHRFAC		
Type	I	F	F	I	I	F		

VARIABLE	DESCRIPTION
HEADING	Heading for the part
PID	Part ID
ELFORM	Element formulation options for thin shells: EQ.1: Hughes-Liu, EQ.2: Belytschko-Tsay, EQ.3: BCIZ triangular shell, EQ.4: C0 triangular shell, EQ.6: S/R Hughes-Liu, EQ.7: S/R co-rotational Hughes-Liu, EQ.8: Belytschko-Leviathan shell, EQ.9: Fully integrated Belytschko-Tsay membrane, EQ.10: Belytschko-Wong-Chiang, EQ.11: Fast (co-rotational) Hughes-Liu, EQ.16: Fully integrated shell element (very fast), Element formulation options for thick shells: EQ.1: one point reduced integration, EQ.2: selective reduced 2 x 2 in plane integration, EQ.3: assumed strain 2 x 2 in plane integration,

VARIABLE **DESCRIPTION**

EQ.5: assumed strain reduced integration with brick materials

EQ.6: : assumed strain reduced integration with shell materials

EQ.7: assumed strain 2x2 in plane integration

SHRF

Shear correction factor which scales the transverse shear stress.

NLOC

Location of reference surface, available for thin shells only. If nonzero, the offset distance from the plane of the nodal points to the reference surface of the shell in the direction of the shell normal vector is a value:

$$\text{offset} = -0.50 \times \text{NLOC} \times (\text{average shell thickness}).$$

This offset is not considered in the contact subroutines unless CNTCO is set to 1 in *CONTROL_SHELL. Alternatively, the offset can be specified by using the OFFSET option in the *ELEMENT_SHELL input section.

EQ.1.0: top surface,

EQ.0.0: mid-surface (default),

EQ.-1.0: bottom surface.

MAREA

Non-structural mass per unit area. This is additional mass which comes from materials such as carpeting. This mass is not directly included in the time step calculation.

HGID

Hourglass/bulk viscosity identification defined in the *HOURGLASS Section:

EQ.0: default values are used.

ADPOPT

Indicate if this part is adapted or not. Also see, *CONTROL_ADAPTIVITY:

EQ.0: no adaptivity,

EQ.1: H-adaptive for 3-D thin shells.

THSHEL

Thermal shell formulation

EQ.0: Default is governed by THSHEL on *CONTROL_SHELL

EQ.1: Thick thermal shell

EQ.2: Thin thermal shell

VARIABLE **DESCRIPTION**

TSHEAR

Flag for transverse shear stress distribution (see Remarks 3 and 4):

EQ.0: Parabolic,

EQ.1: Constant through thickness.

FS

Static coefficient of friction. The functional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact as

$$\mu_c = \mathrm{FD} + (\mathrm{FS} - \mathrm{FD})e^{-\mathrm{DC}\times|v_{\mathrm{rel}}|}.$$

FD

Dynamic coefficient of friction. The functional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact as

$$\mu_c = \mathrm{FD} + (\mathrm{FS} - \mathrm{FD})e^{-\mathrm{DC}\times|v_{\mathrm{rel}}|}.$$

DC

Exponential decay coefficient. The functional coefficient is assumed to be dependent on the relative velocity v_{rel} of the surfaces in contact as

$$\mu_c = \mathrm{FD} + (\mathrm{FS} - \mathrm{FD})e^{-\mathrm{DC}\times|v_{\mathrm{rel}}|}.$$

VC

Coefficient for viscous friction. This is necessary to limit the friction force to a maximum. A limiting force is computed $F_{\mathrm{lim}} = \mathrm{VC} \times A_{\mathrm{cont}}$. A_{cont} being the area of the segment contacted by the node in contact. The suggested value for VC is to use the yield stress in shear $\mathrm{VC} = \frac{\sigma_o}{\sqrt{3}}$ where σ_0 is the yield stress of the contacted material.

OPTT

Optional contact thickness. This applies to shells only.

SFT

Optional thickness scale factor for PART ID in automatic contact (scales true thickness). This option applies only to contact with shell elements. True thickness is the element thickness of the shell elements.

SSF

Scale factor on default slave penalty stiffness for this PART ID whenever it appears in the contact definition. If zero, SSF is taken as unity.

MID*i*

Material ID of integration point *I*, see *MAT_… Section.

THICK*i*

Thickness of integration point *i*.

VARIABLE	DESCRIPTION
B*i*	Material angle of integration point *i*. This material angle applies only to material types 21, 22, 23, 33, 33_96, 34, 36, 40, 41-50, 54, 55, 58, 59, 103, 103_P, 104, 108, 116, 122, 133, 135, 135_PLC, 136, 157, 158, 190, 219, 226, 233, 234, 235, 242, and 243.
TMID*i*	Thermal material ID of integration point *i*
PLYID*i*	Ply ID of integration point *i* (for post-processing purposes)
SHRFAC*i*	Trnansverse shear stress scale factor

Remarks:

1. **Orthotropic Materials.** In cases where there is more than one orthotropic material model referenced by *PART_COMPOSITE, the orthotropic material orientation parameters (AOPT, BETA, and associated vectors) from the material model of the first orthotropic integration point apply to all the orthotropic integration points. AOPT, BETA, etc. input for materials of subsequent integration points is ignored. B*i*, not to be confused with BETA, is taken into account for each integration point.

2. **SHRF Field and Zero Traction Condition.** Thick shell formulations 1, 2, and 3, and all shell formulations with the exception of BCIZ and DK elements, are based on first order shear deformation theory that yields constant transverse shear strains which violates the condition of zero traction on the top and bottom surfaces of the shell. For these elements, setting SHRF=0.83333 will compensate for this error and result in the correct transverse shear deformation, so long as all layers have the same transverse stiffness. SHRF is not used by thick shell forms 3, 5, or 7 except for materials 33, 36, 133, 135, and 243.

3. **Thick Shell 5 or 6 and Shear Stress.** Thick shell formulation 5 and 6 will look to the TSHEAR parameter and use either a parabolic transverse shear stress distribution when TSHEAR=0, or a constant shear stress distribution when TSHEAR=1. The parabolic option is recommended when elements are used in a single layer to model a plate or beam. The constant option may be better when elements are stacked so there are two or more elements through the thickness.

4. **Laminated Shear Stress Theory to Minimize Discontinuities.** For composites that have a transverse shear stiffness that varies by layer, laminated shell theory, activated by LAMSHT on *CONTROL_SHELL, will correct the transverse shear stress to minimize stress discontinuities between layers and at the bottom and top surfaces by imposing a parabolic transverse shear stress. SHRF

should be set to the default value of 1.0 when the shear stress distribution is parabolic. If thick shells are stacked so that there is more than one element through the thickness of a plate or beam model, setting TSHEAR=1 will cause a constant shear stress distribution which may be more accurate than parabolic. The TSHEAR parameter is available for all thick shell forms when laminated shell theory is active. Alternatively, a scale factor can be defined for transverse shear stress in each layer of shell or thick shell composites using the SHRFAC parameter. The inputted SHRFAC values are normalized so that overall shear stiffness is unaffected by the distribution. Therefore, only the ratio of parameter values is significant.

5. **Assignment of Zero Thickness to Integration Points.** The ability to assign zero- thickness integration points in the stacking sequence allows the number of integration points to remain constant even as the number of physical plies varies from part to part and eases post-processing since a particular integration point corresponds to a physical ply. Such a capability is important when one or more of the physical plies are not continuous across a composite structure. To represent a missing ply in *PART_COMPOSITE, set THICKi to 0.0 for the corresponding integration point and additionally, either set MID=-1 or, if the LONG option is used, set PLYID to any nonzero value.

   To carry this concept a step further, in cases where the number of physical plies varies from element to element in a part, one can assign zero thickness to integration points in exactly the same manner as described above but on an element-by-element basis using *ELEMENT_SHELL_COMPOSITE(_LONG) or *ELMENT_TSHELL_COMPOSITE.

   When postprocessing the results using LS-PrePost version 4.5, read both the keyword deck and d3plot database into the code and then select *Option > N/A gray fringe*. Then, when viewing fringe plots for a particular integration point (*FriComp > IPt > intpt#*), the element will be grayed out if the selected integration point is missing (or has zero thickness) in that element.

PART_DUPLICATE_*{OPTION}

The available *OPTION* is

NULL_OVERLAY

This option is used to generate null shells for contact.

Purpose: To provide a method of duplicating parts or part sets without the need to use the *INCLUDE_TRANSFORM option.

Duplication Cards. This format is used when the keyword option is left <BLANK>. Include as many of these cards as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PTYPE	TYPEID	IDPOFF	IDEOFF	IDNOFF	TRANID		
Type	A	I	I	I	I	I		
Default	none	none	0	0	0	0		

Null Duplication Cards. This format is used when the keyword option is set to NULL_OVERLAY. Include as many of these cards as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PTYPE	TYPEID	IDPOFF	IDEOFF	DENSITY	E	PR	
Type	A	I	I	I	F	F	F	
Default	none	none	0	0	0	0	0	

VARIABLE	DESCRIPTION
PTYPE	Set to "PART" to duplicate a single part or "PSET" to duplicate a part set.
TYPEID	ID of part or part set to be duplicated.

VARIABLE	DESCRIPTION
IDPOFF	ID offset of newly created parts.
IDEOFF	ID offset of newly created elements.
IDNOFF	ID offset of newly created nodes.
TRANID	ID of *DEFINE_TRANSFORMATION to transform the existing nodes in a part or part set.
DENSITY	Density.
E	Young's modulus.
PR	Poisson's ratio.

Remarks:

1. All parts sharing common nodes have to be grouped in a *PART_SET and duplicated in a single *PART_DUPLICATE command so that the newly duplicated parts still share common nodes

2. The following elements which need a PART to complete their definition can be duplicated by using this command: *ELEMENT_SOLID, *ELEMENT_DISCRETE, *ELEMENT_SHELL, *ELEMENT_TSHELL, *ELEMET_BEAM and *ELEMENT_SEATBELT.

3. This command only duplicates definition of nodes, elements and parts, not the associated constraints. For example, TC and RC defined in *NODE will not be passed to the newly created nodes.

4. When IDNOFF = IDPOFF = IDEOFF = 0, the existing part, or part set, will be transformed as per TRANID, no new node or elements will be created.

5. The NULL_OVERLAY option may be used to generate 3 and 4-node null shell elements from the 6- and 8-node quadratic elements for use in contact. No additional nodes are generated.

*PART_MODES

Purpose: Treat a part defined with *MAT_RIGID as a linearized flexible body (LFB) whereby deformations are calculated from mode shapes. Unlike superelements (*ELEMENT_DIRECT_MATRIX_INPUT), linearized flexible bodies are accurate for systems undergoing large displacements and large rotations.

Currently, linearized flexible bodies cannot share nodes with other linearized flexible bodies or rigid bodies; however, interconnections to other linearized flexible bodies or to rigid bodies can use the penalty joint option. The linearized flexible bodies are not implemented with the Lagrange multiplier joint option (see LMF in *CONTROL_-RIGID).

The deformations are modeled using the modes shapes obtained experimentally or in a finite element analysis, e.g., NASTRAN .pch file or a LS-DYNA d3eigv or d3mode file. These files may contain a combination of normal modes, constraint modes, and attachment modes. For stress recovery in linearized flexible bodies, use of linear element formulations is recommended. A lump mass matrix is assumed in the implementation. See also *CONTROL_RIGID.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	NMFB	FORM	ANSID	FORMAT	KMFLAG	NUPDF	SIGREC
Type	I	I	I	I	I	I	I	

Card 2	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	C							
Default	none							

Kept Mode Cards. Additional card KMFLAG = 1. Use as many cards as necessary to specify the NMFB kept modes. After NMFB modes are defined no further input is expected.

Card 3.	1	2	3	4	5	6	7	8
Variable	MODE1	MODE2	MODE3	MODE4	MODE5	MODE6	MODE7	MODE8
Type	I	I	I	I	I	I	I	I
Default	none	nont	none	nont	none	nont	none	nont

Optional Modal Damping Cards. This input ends at the next keyword ("*") card.

Card 4	1	2	3	4	5	6	7	8
Variable	MSTART	MSTOP	DAMPF					
Type	I	I	F					
Default	none	none	none					

VARIABLE	DESCRIPTION
PID	Part identification. This part must be a rigid body.
NMFB	Number of kept modes in linearized flexible body. The number of modes in the file, FILENAME, must equal or exceed NMFB. If KMFLAG = 0 the first NMFB modes in the file are used.
FORM	Linearized flexible body formulation. See remark 5 below. EQ.0: exact EQ.1: fast EQ.3: general formulation (default) EQ.4: general formulation without rigid body mode orthogonalization.
ANSID	Attachment node set ID (optional).
FORMAT	Input format of modal information:

VARIABLE DESCRIPTION

EQ.0: NASTRAN.pch file.

EQ.1: (not supported)

EQ.2: NASTRAN.pch file (LS-DYNA binary version). The binary version of this file is automatically created if a NASTRAN.pch file is read. The name of the binary file is the name of the NASTRAN.pch file but with ".bin" appended. The binary file is smaller and can be read much faster.

EQ.3: LS-DYNA d3eigv binary eigenvalue database (see *CONTROL_IMPLICIT_EIGENVALUE).

EQ.4: LS-DYNA d3mode binary constraint/attachment mode database (see *CONTROL_IMPLICIT_MODE).

KMFLAG Kept mode flag. Selects method for identifying modes to keep. This flag is not supported for FORMAT = 4 (d3mode).

EQ.0: the first NMFB modes in the file, FILENAME, are used.

EQ.1: define NMFB kept modes with additional input.

NUPDF Nodal update flag. If active, an attachment node set, ANSID, must be defined.

EQ.0: all nodes of the rigid part are updated each cycle.

EQ.1: only attachment nodes are fully updated. All nodes in the body are output based on the rigid body motion without the addition of the modal displacements. For maximum benefit an attachment node set can also be defined with the PART_ATTACHMENT_NODES option. The same attachment node set ID should be used here.

VARIABLE **DESCRIPTION**

SIGREC

Stress recovery flag.

EQ.0: Do not recover stress.

EQ.1: Recover stress.

EQ.2: Recover stress and then set the recovery stress as initial stress when switching to a deformable body via *DEFORMABLE_TO_RIGID_AUTOMATIC. (shell formulations 16, 18, 20, 21 and solid formulation 2).

EQ.3: Recover stress based on shell formulation 21, and then set the recovery stress as initial stress for shell formulation 16 when switching to a deformable body via *DEFORMABLE_TO_RIGID_AUTOMATIC (shell formulation 16 only).

FILENAME

The path and name of a file which contains the modes for this rigid body.

MODE*n*

Keep normal mode, MODE*n*.

MSTART

First mode for damping, (1 ≤ MSTART ≤ NMFB).

MSTOP

Last mode for damping, MSTOP, (1 ≤ MSTOP ≤ NMFB). All modes between MSTART and MSTOP inclusive are subject to the same modal damping coefficient, DAMPF.

DAMPF

Modal damping coefficient, ζ.

Remarks:

1. The format of the file which contains the normal modes follows the file formats of NASTRAN output for modal information.

2. The mode set typically combines both normal modes and attachment modes. The eigenvalues for the attachment modes are computed from the stiffness and mass matrices.

3. The part ID specified must be either a single rigid body or a master rigid body (see *CONSTRAINED_RIGID_BODIES) which can be made up of many rigid parts.

4. The modal damping is defined by the modal damping coefficient ζ, where a value of 1.0 equals critical damping. For a one degree of freedom model system, the relationship between the damping and the damping coefficient is

$c = 2\zeta\omega_n m$, where c is the damping, m is the mass, and ω_n is the natural frequency, $\sqrt{k/m}$.

5. There are four formulations. The first is a formulation that contains all the terms of the linearized flexible body equations, and its cost grows approximately as the square of the number of modes. The second formulation ignores most of the second order terms appearing in the exact equations and its cost grows linearly with the number of modes. If the angular velocities are small and if the deflections are small with respect to the geometry of the system, the cost savings of the second formulation may make it more attractive than the first method.

   Please note that the first two formulations are only applicable when the modes are eigenmodes computed for the free-free problem, that is including the 6 rigid body modes. The third formulation, the default, is a more general formulation which allows more general mode shapes. It is strongly recommended that the default formulation be used. The fourth formulation does not orthogonalize the modes with respect to the rigid body modes, and may allow boundary conditions to be imposed more simply in some cases that the third formulation.

*PART_MOVE

Purpose: Translate a part by an incremental displacement in either a local or a global coordinate system. This option currently applies to parts defined either by shell and solid elements. All nodal points of the given part ID are moved. Care must be observed since parts that share boundary nodes with the part being moved must also be moved to avoid severe mesh distortions – the variable IFSET can be used to handle the situation.

Part/Part Set Move Cards. Include as many of following cards as desired. This input ends at the next keyword ("*") cards.

Card 1	1	2	3	4	5	6	7	8	9	10
Variable	PID/PSID	XMOV		YMOV		ZMOV		CID	IFSET	
Type	I	F		F		F		I	I	
Default	none	0.0		0.0		0.0		0	0	

VARIABLE	DESCRIPTION

PID/PSID Part or part set identification number.

XMOV Move shell/solid part ID, PID, in the x-direction by the incremental distance, XMOV.

YMOV Move shell/solid part ID, PID, in the y-direction by the incremental distance, YMOV.

ZMOV Move shell/solid part ID, PID, in the z-direction by the incremental distance, ZMOV.

CID Coordinate system ID to define incremental displacement in local coordinate system. All displacements, XMOV, YMOV, and ZMOV, are with respect to CID.

EQ.0: global

IFSET Indicate if part set ID (SID), is used in PID/SID definition.

EQ.0: part ID (PID) is used

EQ.1: part set ID (SID) is used.

Remarks:

1. A new variable IFSET is added to address the move of multiple parts that share common boundary nodes, e.g., in case of tailor-welded blanks. The new variable allows for a part set to be move simultaneously. For example, keyword *SET_PART_LIST can be used to include all tailor welded blank part IDs and the resulting Part Set ID can be used in this keyword.

2. Draw beads can be modeled as beam elements and moved in the same distance and direction as either the die or punch, depending on the draw types.

3. A partial keyword input is provided below to automatically position all tools in a toggle draw of a decklid inner, with a tailor welded blank consisting of PID 1 and PID5, as shown in Figure 31-1. With the use of the keyword *CONTROL_-FORMING_AUTOPOSITION_PARAMETER_SET, the tailor-welded blank part set ID 1 is to be positioned in the global Z-direction on top of the lower die cavity (part set ID 4); the binder (part set ID 3) is to be positioned on top of the blank; and finally the upper punch (part set ID 2) is to be positioned on top of the blank. The three positioning distances for the blank, upper binder and upper punch are calculated and stored in variables &blnkmv, &upbinmv, and &uppunmv, respectively. The keyword *PART_MOVE, with IFSET of “1”, is responsible to actually move the three part sets, using the three corresponding positioning variables. It is noted that the AUTOPOSITION keyword is only applicable to shell elements.

```
*PARAMETER
R   blnkmv       0.0
R  upbinmv       0.0
R  uppunmv       0.0
*SET_PART_LIST
1
1,5
*SET_PART_LIST
2
2
*SET_PART_LIST
3
3
*SET_PART_LIST
4
4
*CONTROL_FORMING_AUTOPOSITION_PARAMETER_SET
$  PID/SID       CID       DIR MPID/MSID  Position   PREMOVE     THICK
PARORDER
         1                   3         4         1                 1.5
blnkmv
         3                   3         1         1                 1.5
upbinmv
         2                   3         1         1                 1.5
uppunmv
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+---
-8
*PART_MOVE
$    PID            XMOV            YMOV            ZMOV     CID   IFSET
       1             0.0             0.0         &blnkmv               1
```

```
            3                  0.0                 0.0          &upbinmv                1
            2                  0.0                 0.0          &uppunmv                1
```

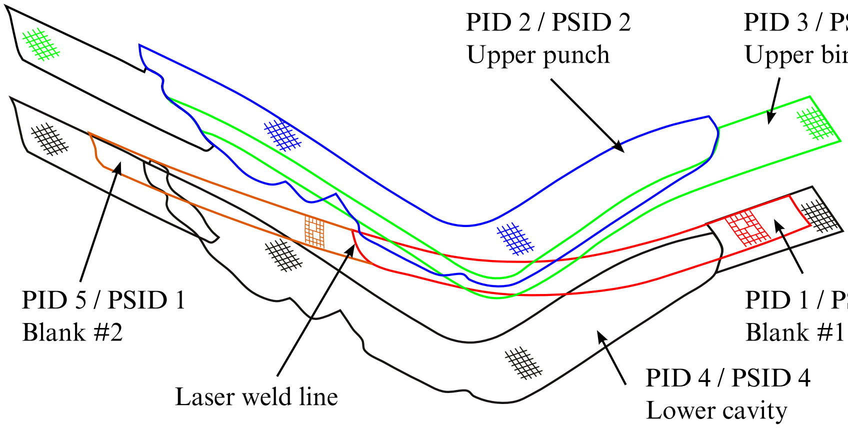


Figure 31-1. A tailor welded blank is positioned in a decklid (toggle draw).

Revision information:

This IFSET feature is available starting in LS-DYNA Revision #62935. It is also implemented in all the applicable stamping processes in LS-PrePost4.0 Metal Forming Application eZ-Setup (http://ftp.lstc.com/anonymous/outgoing/lsprepost/4.0/metalforming/).

*PART_SENSOR

Purpose: Activate and deactivate parts, based on sensor defined in ELEMENT_SEATBELT_SENSOR. This option applies to discrete beam element only.

Sensor Part Coupling Cards. Include as many of the following cards as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	SIDA	ACTIVE					
Type	I	I	I					
Default	0	0	0					

VARIABLE	DESCRIPTION
PID	Part ID, which is controlled by sensor
SIDA	Sensor ID to activate or deactivate part.
ACTIVE	Flag. If zero, the part is active from time zero until a signal is received by the part to deactivate. If one, the part is inactive from time zero and becomes active when a signal is received by the part to activate. The history variables for inactive parts are initialized at time zero.

*PART_STACKED_ELEMENTS

Purpose: This keyword provides a method of defining a stacked element model for shell-like structures. These types of plane load-bearing components possess a thickness which is small compared to their other (in-plane) dimensions. Their physical properties vary in the thickness direction according to distinct layers. Application examples include sandwich plate systems, composite laminates, plywood, and laminated glass.

With this keyword it is possible to discretize layered structures by an arbitrary sequence of shell and/or solid elements over the thickness. Whether a physical ply should be discretized by shell or solid elements depends on the individual thickness and other mechanical properties. Every single layer can consist of one shell element over the thickness or one or several solid elements over thickness.

The stacked element mesh can either be provided directly or it be automatically generated by LS-DYNA itself. For automatic generation extrusion methods are used to determine the new node locations that characterize the out-of-plane geometry. Each layer gets its own predefined properties such as individual thickness, material characteristic, and element type.

A more detailed description of this feature including a detailed description of the appropriate mesh generation procedure is given in Erhart [2015].

Card 1	1	2	3	4	5	6	7	8
Variable	HEADING							
Type	C							
Default	none							

Card 2	1	2	3	4	5	6	7	8
Variable	PIDREF	NUMLAY	ADPOPT					
Type	I	I	I					
Default	none	none	0					

Layer Data Cards. The part ID, section ID, material ID, hourglass ID, thermal material ID, thickness, and number of through thickness solid elements for each layer *i* of a stacked element model are provided below. The layer data should be given sequentially starting with the bottommost layer. This card should be included NUMLAY times (one for each layer). The definitions in Card 3 replace the usual *PART cards.

Card 3	1	2	3	4	5	6	7	8
Variable	PID*i*	SID*i*	MID*i*	HGID*i*	TMID*i*	THK*i*	NSLD*i*	
Type	I	I	I	I	I	F	I	
Default	none	none	none	0	0	none	none	

VARIABLE	DESCRIPTION
HEADING	Heading for the part composition.
PIDREF	Part ID of reference shell element mesh.
NUMLAY	Number of layers.
ADPOPT	Indicate if parts are adapted or not. (See also *CONTROL_ADAPTIVE): EQ.0: inactive EQ.1: h-adaptive refinement
PID*i*	Part identification.
SID*i*	Section identification for layer *i* defined in a *SECTION keyword.
MID*i*	Material identification for layer *i* defined in a *MAT keyword.
HGID*i*	Hourglass identification for layer *i* defined in a *HOURGLASS keyword.
TMID*i*	Thermal material identification for layer *i* defined in a *MAT_-THERMAL keyword.
THK*i*	Thickness of layer *i*.

VARIABLE	DESCRIPTION
NSLD*i*	Number of through-thickness solid elements for layer *i*.

Remarks:

1. **Provided vs. Automatically Generated Meshes.** In general, there are two different options for this keyword:

   a) The user provides a finished mesh comprising stacked shell and/or solid elements and then combines the corresponding part IDs using this keyword. This mode does not require a reference mesh in the PIDREF field nor does it require that either the layer thickness (THK*i* fields) and the the number of through-thickness solids elements (NSLD*i* fields) be specfied.

   b) The user may provide a shell reference mesh (PIDREF) together with the layup sequence. The stacked element mesh is automatically generated during the initialization phase of LS-DYNA. In that second case, layer thickness (THK*i*) and number of through-thickness solids (NSLD*i*) must be defined.

2. **Shell Solid Overlap.** In the mesh generation case, two consecutive layers (solid-solid or solid-shell) are firmly connected, meaning that they share nodes in the most obvious way possible (except they are both shell element layers, see Remark 3 for that case). This condition leads to the necessity that shell and solid elements partly overlap if they follow each other in the stacking sequence. This deficiency can be corrected afterwards by subsequent relocation of the shell mid-surfaces via NLOC on *SECTION_SHELL (only works for first or last layer in this stacked element approach) or appropriate adjustment of the material stiffness for the solid elements.

3. **Stacked Shells.** Starting with the release of LS-DYNA version R10, it is possible to define shell element layers directly on top of each other (i.e. without solid elements in between). A potential connection/interaction of such layers has to be declared separately by additional contact definitions (standard, tied, or tiebreak) otherwise they are free to penetrate each other.

4. **Chained Calculations.** This keyword (*PART_STACKED_ELEMENTS) can also be used for modeling multi-stage processes. The *INTERFACE_SPRINGBACK_LSDYNA card can be used to save a final state including deformed geometry, stresses, and strains to a dynain file. In a subsequent calculation the *INCLUDE keyword can be used with that dynain to apply the layup sequence without regenerating the mesh. LS-DYNA automatically detects if a reference shell element mesh is present or not.

5. **An Example.** An example specifying a three layer shell-solid-shell structure is given below:

```
*PART_STACKED_ELEMENTS
$ title
sandwich
$   pidref    numlay
        11         3
$#     pid       sid       mid      hgid      tmid       thk      nsld
       100       200         1         1         0      0.25         0
       101       201         2         0         0      0.60         3
       102       200         1         1         0      0.15         0
*SECTION_SHELL
$      sid    elform      shrf       nip     propt   qr/irid
       200         2     0.833       5.0       1.0       0.0
$       t1        t2        t3        t4
      0.25      0.25      0.25      0.25
*SECTION_SOLID
$      sid    elform
       201        -1
```

A sandwich structure is discretized by shell elements (SID = 200) on the outer layers with part identifiers 100 and 102. The interior of the "sandwich" consists of three solid elements (SID = 201, NSLD = 3) part identifier 101. In this case, the reference shell mesh belongs to part 11 (PIDREF). The thickness of each layer is defined by the value of the THK field, which will overwrite the thickness values from *SECTION_SHELL. Related materials (MID, TMID) and hourglass types (HGID) are treated as usual and therefore not shown here.

*PARTICLE_BLAST

Purpose: To define control parameters for particle based blast loading.

Card 1	1	2	3	4	5	6	7	8
Variable	LAGSID	LAGSTYPE	NODID	NODTYPE	HECID	HECTYPE	AIRCID	
Type	I	I	I	I	I	I	I	
Default	0	0	0	0	0	0	0	

Card 2	1	2	3	4	5	6	7	8
Variable	NPHE	NPAIR	IUNIT					
Type	I	I	I					
Default	0	0	0					

Card 3	1	2	3	4	5	6	7	8
Variable	IHETYPE	DENSITY	ENERGY	GAMMA	COVOL	DETO_V		
Type	I	F	F	F	F	F		
Default	0	0	0	0	0	0		

Card 4	1	2	3	4	5	6	7	8
Variable	DETX	DETY	DETZ	TDET	BTEND	NID		
Type	F	F	F	F	F	I		
Default	0	0	0	0	0	0		

Card 5	1	2	3	4	5	6	7	8
Variable	BCX0	BCX1	BCY0	BCY1	BCZ0	BCZ1		
Type	F	F	F	F	F	F		
Default	0	0	0	0	0	0		

Card 6	1	2	3	4	5	6	7	8
Variable	IBCX0	IBCX1	IBCY0	IBCY1	IBCZ0	IBCZ1	BC_P	
Type	I	I	I	I	I	I	I	
Default	0	0	0	0	0	0	0	

VARIABLE **DESCRIPTION**

LAGSID Structure id for particle structure interaction

LAGSTYPE Structure type

EQ.0: Part Set

EQ.1: Part

NODID Discrete element sphere (DES) or Smooth particle hydrodynamics (SPH) id for the interaction between particles and nodes.

NODTYPE Nodal type

EQ.0: Node Set

EQ.1: Node

EQ.2: Part Set

EQ.3: Part

HECID Initial container for high explosive particle

VARIABLE	DESCRIPTION
HECTYPE	Structure type EQ.0: Part Set EQ.1: Part EQ.2: Geometry, see *DEFINE_PBLAST_GEOMETRY
AIRCID	Initial geometry for air particles EQ.0: filled air particles to entire domain defined by Card 5 GT.0: Reference to *DEFINE_PBLAST_AIRGEO ID
NPHE	Number of high explosive particles
NPAIR	Number of air particles
IUNIT	Unit System EQ.0: Kg-mm-ms-K EQ.1: SI Units EQ.2: Ton-mm-s-K EQ.3: g-cm-us-K EQ.4: $lbf \bullet s^2/in$- in -s-K
IHETYPE	High Explosive type (See Remark 1) EQ.1: TNT EQ.2: C4 Others: Self Define
DENSITY	High Explosive density
ENERGY	High Explosive energy per unit volume
GAMMA	High Explosive fraction between C_p and C_v
COVOL	High Explosive co-volume
DET_V	High Explosive detonation velocity
DETX	Detonation point x
DETY	Detonation point y

VARIABLE	DESCRIPTION
DETZ	Detonation point z
TDET	Detonation time
BTEND	Blast end time
NID	An optional node ID defining the position of the detonation point. If defined, its coordinates will overwrite the DETX, DETY, and DETZ defined above.
BCX0	Global domain x-min
BCX1	Global domain x-max
BCY0	Global domain y-min
BCY1	Global domain y-max
BCZ0	Global domain z-min
BCZ1	Global domain z-max
IBCX0	Boundary conditions for global domain x-min EQ.0: Free EQ.1: Rigid reflecting boundary
IBCX1	Boundary conditions for global domain x-max EQ.0: Free EQ.1: Rigid reflecting boundary
IBCY0	Boundary conditions for global domain y-min EQ.0: Free EQ.1: Rigid reflecting boundary
IBCY1	Boundary conditions for global domain y-max EQ.0: Free EQ.1: Rigid reflecting boundary

VARIABLE	DESCRIPTION
IBCZ0	Boundary conditions for global domain z-min EQ.0: Free EQ.1: Rigid reflecting boundary
IBCZ1	Boundary conditions for global domain z-max EQ.0: Free EQ.1: Rigid reflecting boundary
BC_P	Pressure ambient boundary condition for global domain EQ.0: Off (Default) EQ.1: On (Remark 2)

Remarks:

1. **Common Material Constants for commonly used High Explosives.**

IHETYPE	ρ	e_0	γ	COV	D
TNT	$1630\,\frac{\mathrm{kg}}{\mathrm{m}^3}$	$7\,\frac{\mathrm{GJ}}{\mathrm{m}^3}$	1.35	0.6	$6930\,\frac{\mathrm{m}}{\mathrm{s}}$
C4	$1601\,\frac{\mathrm{kg}}{\mathrm{m}^3}$	$9\,\frac{\mathrm{GJ}}{\mathrm{m}^3}$	1.32	0.6	$8193\,\frac{\mathrm{m}}{\mathrm{s}}$

2. **Pressure Boundary Conditions.** If pressure boundary conditions are used, particles will not escape from the global domain when the pressure in the domain is lower than the ambient.

*PERTURBATION

The keyword *PERTURBATION provides a means of defining deviations from the designed structure such as buckling imperfections. These perturbations can be viewed in LS-PREPOST as user-defined fringe plots. Available options are:

*PERTURBATION_MAT

*PERTURBATION_NODE

*PERTURBATION_SHELL_THICKNESS

*PERTURBATION

PERTURBATION_*OPTION

Available options are:

MAT

NODE

SHELL_THICKNESS

Purpose: Define a perturbation (stochastic field) over the whole model or a portion of the model, typically to trigger an instability. The NODE option modifies the three dimensional coordinates for the whole model or a node set. For the SHELL_THICKNESS option the shell thicknesses are perturbed for the whole model or a shell set. The MAT option perturbs a material parameter value for all the elements associated with that material.

Material Perturbation Card. Card 1 for MAT keyword option. Perturb a material parameter.

Card 1	1	2	3	4	5	6	7	8
Variable	TYPE	PID	SCL	CMP	ICOORD	CID		
Type	I	I	F	I	I	I		
Default	1	0	1.0	7	0	0		

Node Perturbation Card. Card 1 for NODE keyword option. Perturb the coordinates of a node set (or all nodes).

Card 1	1	2	3	4	5	6	7	8
Variable	TYPE	NSID	SCL	CMP	ICOORD	CID		
Type	I	I	F	I	I	I		
Default	1	0	1.0	7	0	0		

Shell Thickness Card. Card 1 for SHELL_THICKNESS keyword option. Perturb the thickness of a set of shells (or all shells).

Card 1	1	2	3	4	5	6	7	8
Variable	TYPE	EID	SCL	ICOORD	CID			
Type	I	I	F	I	I			
Default	1	0	1.0	0	0			

Harmonic Perturbation Cards (TYPE = 1). Card format 2 for TYPE = 1. Include as many cards of the following card as necessary. The input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	AMPL	XWL	XOFF	YWL	YOFF	ZWL	ZOFF	
Type	F	F	F	F	F	F	F	
Default	1.0	0.0	0.0	0.0	0.0	0.0	0.0	

Fade Field Perturbation Card (TYPE = 2). Card format 2 for TYPE = 2.

Card 2	1	2	3	4	5	6	7	8
Variable	FADE							
Type	F							
Default	1.0							

Perturbation From File Card (TYPE = 3). Card format 2 for TYPE = 3.

Card 2	1	2	3	4	5	6	7	8
Variable	FNAME							
Type	A							
Default	none							

Spectral Field Perturbation Card (TYPE = 4). Card format 2 for TYPE = 4 (fade fiel)..

Card 2	1	2	3	4	5	6	7	8
Variable	CSTYPE	ELLIP1	ELLIP2	RND				
Type	I	F	F	I				
Default	none	1.0	1.0	0				

Spectral Perturbation Parameter Cards. Include One, two, or three cards of this format, depending on the value of CSTYPE.

Card 3	1	2	3	4	5	6	7	8
Variable	CFTYPE	CFC1	CFC2	CFC3				
Type	I	F	F	F				
Default	none	1.0	1.0	1.0				

VARIABLE	DESCRIPTION
TYPE	Type of perturbation EQ.1: Harmonic Field EQ.2: Fade out all perturbations at this node set EQ.3: Read perturbations from a file EQ.4: Spectral field

VARIABLE	DESCRIPTION
PID	Part ID.
NSID	Node set ID. Specify 0 to perturb all the nodes in the model.
EID	Element set ID. Specify 0 to perturb all the elements in the model.
SCL	Scale factor
CMP	Component. For the NODE option, these are given below. For the MAT option, see the description of the material. EQ.1: x coordinate EQ.2: y coordinate EQ.3: z coordinate EQ.4: x and y coordinate EQ.5: y and z coordinate EQ.6: z and x coordinate EQ.7: x, y, and z coordinate
ICOORD	Coordinate system to use; see Remarks 7, 8 and 9 EQ.0: Global Cartesian EQ.1: Cartesian EQ.2: Cylindrical (computed and applied) EQ.3: Spherical (computed and applied) EQ.-2: Computed in cartesian but applied in cylindrical EQ.-3: Computed in cartesian but applied in spherical
CID	Coordinate system ID, see *DEFINE_COORDINATE_NODES
AMPL	Amplitude of the harmonic perturbation
XWL	x wavelength of the harmonic field
XOFF	x offset of harmonic field
YWL	y wavelength of the harmonic field
YOFF	y offset of harmonic field

VARIABLE DESCRIPTION

ZWL z wavelength of the harmonic field

ZOFF z offset of harmonic field

FADE Parameter controlling the distance over which all *PERTURBATION_NODE are faded to zero

FNAME Name of file containing the perturbation definitions

CSTYPE Correlation structure:

EQ.1: 3D isotropic. The x, y and z correlations are described using one correlation function. Define CFC1.

EQ.2: 3D product. The x, y and z correlations are described using a correlation function each. Define CFC1, CFC2 and CFC3.

EQ.3: 2D isotropic. A correlation function describes the x correlation while the yz isotropic relationship is described using another correlation function. Define CFC1 and CFC2.

EQ.4: 2D isotropic. The xz isotropic relationship is described using a correlation function, while another correlation function describes the y correlation while. Define CFC1 and CFC2.

EQ.5: 2D isotropic. The xy isotropic relationship is described using a correlation function, while another correlation function describes the z correlation while. Define CFC1 and CFC2.

EQ.6: 3D elliptic. Define CSE1, CSE2 and CFC1.

EQ.7: 2D elliptic. A correlation function describes the x correlation while the yz elliptic relationship is described using another correlation function. Define CSE1 and CFC1.

EQ.8: 2D elliptic. A correlation function describes the y correlation while the zx elliptic relationship is described using another correlation function. Define CSE1 and CFC1.

EQ.9: 2D elliptic. The xy elliptic relationship is described using a correlation function, while another correlation function describes the z correlation while. Define CSE1 and

VARIABLE	DESCRIPTION
	CFC1.
ELLIP1	Elliptic constant for 2D and 3D elliptic fields
ELLIP2	Elliptic constant for 3D elliptic field
RND	Seed for random number generator. EQ.0: LS-DYNA will generate a random seed GT.0: Value to be used as seed
CFTYPE	Correlation function EQ.1: Gaussian EQ.2: Exponential EQ.3: Exponential Cosine EQ.4: Rational EQ.5: Linear
CFC*i*	Correlation function constant *i*

Remarks:

1. **Postprocessing.** The perturbation can be viewed in LS-PrePost. For the NODE option, LS-DYNA creates files named `pert_node_x/y/z/res`, which can be viewed as user-defined fringe plots. For the SHELL_THICKNESS and MAT options, the files are named `pert_shell_thickness` and `pert_mat` respectively. If a coordinate system with a radial component is used, then the file `pert_node_radial` is also written.

2. **Linear Combinations and Maximum Amplitudes.** Perturbations specified using separate *PERTURBATION cards are created separately and then added together. This is true as well for special cases such as CMP = 7 in which case the x, y and z fields are created separately and added together afterwards, which can result in an absolute amplitude greater than specified using AMPL or SCL.

3. **Harmonic Perturbations.** The harmonic perturbation is

$$p_{\mathrm{CMP}}(x,y,z) = \mathrm{SCL} \\ \times \mathrm{AMPL}\times\left[\sin\left(2\pi\frac{x+\mathrm{XOFF}}{\mathrm{XWL}}\right)+\sin\left(2\pi\frac{y+\mathrm{YOFF}}{\mathrm{YWL}}\right) \\ +\sin\left(2\pi\frac{z+\mathrm{ZOFF}}{\mathrm{ZWL}}\right)\right]$$

Note that the harmonic perturbations can sum to values greater than SCL × AMPL.

4. **The Fade Perturbation.** The fade perturbation is

$$p'(x,y,z) = SCL\times\left(1-\frac{1}{e^{\mathrm{FADE}\times x'}}\right)p(x,y,z)$$

where x' the shortest distance to a node in the node set specified and FADE the parameter controlling the sharpness of the fade perturbation.

5. **Keyword Format for FNAME Field.** The file FNAME must contain the perturbation in the LS-DYNA keyword format. This file can be created from the d3plot results using the LS-PrePost Output capability. The data must be arranged into two columns with the first column being the node ids. Lines starting with the character $ will be ignored.

6. **Correlation Functions.** The correlation functions are defined as follows:

   a) Gaussian: $B(t) = e^{-(at)^2}$

   b) Exponential: $B(t) = e^{-|at|^b}$

   c) Exponent and Cosine: $B(t) = e^{-|at|}\cos(bt)$

   d) Rational: $B(t) = (1+|at|^b)^{-c}$

   e) Piecewise Linear: $B(t) = (1-|at|)\chi(1-|at|)$

   f) With χ the Heaviside step function and a, b and c corresponding to CFC1, CFC2 and CFC3.

7. **Cylindrical Coordinates.** For the cylindrical coordinate system option (ICOORD = 2), the default is to use the global coordinate system for the location of the cylindrical part, with the base of the cylinder located at the origin, and the global z-axis aligned with the cylinder axis. For cylindrical parts not located at the global origin, define a coordinate system (numbered CID) using *DEFINE_COORDINATE_NODES by selecting any three nodes on the base of the cylinder in a clockwise direction (resulting in the local z-axis to be aligned with the cylinder).

8. **Spherical Coordinates.** For the spherical coordinate system (ICOORD = 3), the coordinates are the radius, zenith angle $[0, \pi]$, and the azimuth angle $[0,2\pi]$. The default is to use the global coordinate system with the zenith measured from the z-axis and the azimuth measured from the x-axis in the xy-plane. For spherical parts not located at the global origin, define a coordinate system using *DEFINE_COORDINATE_NODES by selecting any three nodes as follows: the first node is the center of the sphere, the second specifies the x-axis of the coordinate system, while the third point specifies the plane containing the new y-axis. The z-axis will be normal to this plane.

9. **Computed In Cartesian Applied to Cylindrical or Spherical.** It is possible to compute the perturbations in a Cartesian coordinate system, but to apply them in a cylindrical or spherical coordinate system (ICOORD = -2, -3). This is the natural method of doing say a radial perturbation of a sphere using a spectral perturbation field. We expect that computing the perturbation in the spherical coordinate system should be rare (ICOORD = 3). Computing a perturbation in a cylindrical coordinate system should be common though; for example, a circumferential harmonic perturbation.

10. **Material Perturbation Feature.** Only *MAT_238 (*MAT_PERT_PIECEWISE_LINEAR_PLASTICITY) and solid elements in an explicit analysis can be perturbed using *PERTURBATION_MAT. See the documentation of this material for allowable components. Only one part per model can be perturbed. For some perturbed quantity c, the material perturbation is applied on an element-by-element basis as

    $$c_{\text{new}} = (1 + p)c_{\text{base}}$$

    where p is a random number, which is written to the `pert_mat` file during the calculation. Values of p less than -1 are not allowed because the material behavior is not defined.

    Completely independent of *PERTURBATION_MAT, see *DEFINE_STOCHASTIC_VARIATION for a way to define a stochastic variation of yield stress and/or failure strain in material models 10, 15, 24, 81, and 98 and the shell version of material 123.

.

*RAIL

Two keywords are defined in this section.

*RAIL_TRACK

*RAIL_TRAIN

*RAIL_TRACK

Purpose: Wheel-rail contact algorithm intended for railway applications but can also be used for other purposes. The wheel nodes (defined on *RAIL_TRAIN) represent the contact patch between wheel and rail. A penalty method is used to constrain the wheel nodes to slide along the track. A track consists of two rails, each of which is defined by a set of beam elements.

Card Sets. For each track include one pair of cards 1 and 2. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	BSETID1	NORGN1	LCUR1	OSET1	SF1	GA1	IDIR
Type	I	I	I	I	F	F	F	I
Default	none	none	none	none	0.0	1.0	0.0	0

Card 2	1	2	3	4	5	6	7	8
Variable	blank	BSETID2	NORGN2	LCUR2	OSET2	SF2	GA2	
Type	-	I	I	I	F	F	F	
Default	-	none	none	none	0.0	1.0	0.0	

VARIABLE	DESCRIPTION
ID	Track ID
BSETID1,2	Beam set ID for rails 1 and 2 containing all beam elements that make up the rail, see *SET_BEAM.
NORGN1,2	Reference node at one end of each rail, used as the origin for the roughness curve. The train will move in a direction away from this node.

VARIABLE	DESCRIPTION
LCUR1,2	Load curve ID (see *DEFINE_CURVE) defining track roughness (vertical displacement from line of beam elements) of the rail as a function of distance from the reference node NORIGIN. Distance from reference node on x-axis of curve, roughness on y-axis. Default: no roughness.
OSET1,2	Origin of curve LCUR is shifted by distance OSET towards the reference node.
SF1,2	Roughness values are scaled by SF. Default: 1.0.
GA1,2	Shear stiffness of rail per unit length (used to calculate local rail shear deformation within each beam element). GA = shear modulus x cross-sectional area. Default: local shear deformation is ignored.
IDIR	Determines which way is “up” for purposes of wheel/rail contact. Vertical contact works like a normal penalty-based contact while horizontal contact follows Figure 34-2. EQ.0: (Default) global z is “up” and the global x-y plane is assumed horizontal irrespective of the geometry of the rails. EQ.1: “Up” is the normal vector to the plane containing the 2 rails, given by the vector c where c = (a x b), a is the direction along rail 1 heading away from node NORGN1 and b is the vector from rail 1 to rail 2. Both a and b are determined locally at the contact point EQ.-1: Same as IDIR = 1 except “up” is along -c.

Remarks:

*RAIL_TRACK and *RAIL_TRAIN were written by Arup to represent wheel-rail contact. They have been used to generate loading on models of bridges for vibration predictions, stress calculations and for estimating accelerations experienced by passengers. Other non-railway uses are possible: the algorithm causes the “train” nodes to follow the line defined by the “rail” beam elements and transfers forces between them. In some cases (especially vibration modeling), double precision versions of LS-DYNA may give superior results because of the small relative deflections between wheel and rail.

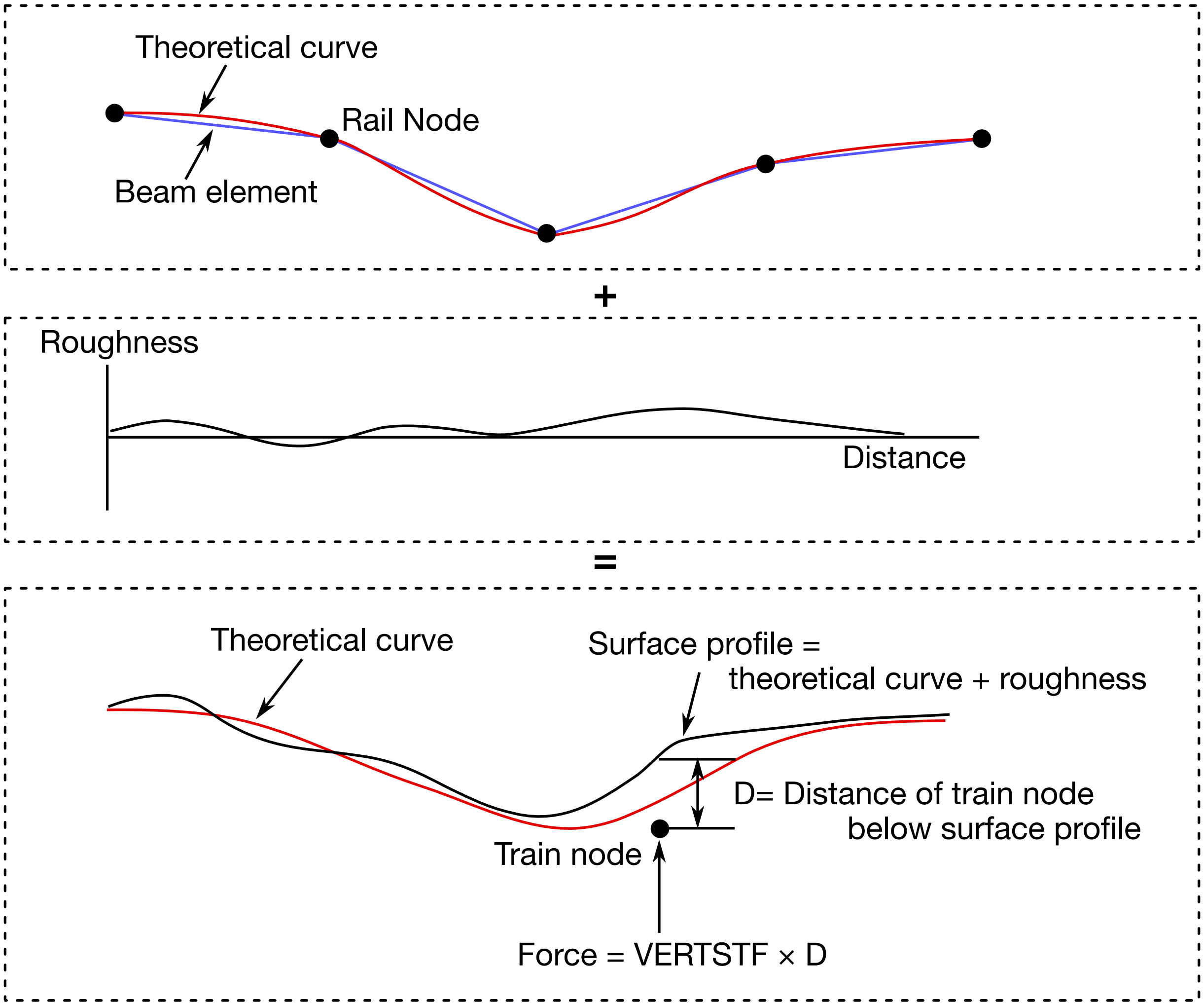


Figure 34-1. Track Model

Track modeling:

The rails of the track should be modeled by two parallel lines of beam elements. The track can be curved or straight and the rails can be modeled as deformable or rigid. If required, rail pads, sleepers and ballast may also be modeled – typically with spring, damper and beam elements. It is also possible to use this algorithm to control the motion of simple road vehicle models: beam element “rails” made of null material can be embedded in the road surface. It is recommended that the mesh size of the two rails should be similar: LS-DYNA calculates a local coordinate system for each train node based on the alignment of the currently contacted beam element and the nearest node on the other rail.

Because wheel-rail contact stiffness is generally very high, and wheel masses are large, small deviations from a straight line or smooth curve can lead to large transient forces. It is recommended that great care be taken in generating and checking the geometry for the track, especially where the track is curved. Some pre-processors write the coordinates with insufficient precision to the LS-DYNA input file, and this can cause unintended roughness in the geometry. For the same reason, if the line of the track were taken as straight between nodes, spurious forces would be generated when the wheel passes from one rail element to the next. This is avoided because the *RAIL algorithm calculates a theoretical curved centerline for the rail element to achieve continuity of slope from one element to the next. Where the length of the rail elements is similar to or shorter than the maximum section dimension, shear deformation may be significant and it is possible to include this in the theoretical centerline calculation to further reduce spurious forces at the element boundaries (inputs GA1, GA2).

Roughness (small deviations in the vertical profile from a perfect straight line) does exist in real life and is a principal source of vibration. *RAIL allows the roughness to be modeled by a load curve giving the vertical deviation (in length units) of the rail surface from the theoretical centerline of the beam elements as a function of distance along the track from the origin node of the rail. The roughness curve is optional. Ideally, roughness profiles measured from both rails of the same piece of track should be used so that the relationship between bump and roll modes is correctly captured.

Whether roughness is included or not, it is important to select as the origin nodes (NORIGIN1 and NORIGIN2) the nodes at the end of the rails away from which the train will be traveling. The train can start at any point along the rails but must travel away from the origin nodes.

Train modeling:

The vehicle models are typically modeled using spring, damper and rigid elements, or simply a point mass at each wheel position. Each node in the set referred to on *RAIL_-TRAIN represents the contact patch of one wheel (note: not the center of the wheel). These nodes should be initially on or near the line defined by either of the two rails. LS-DYNA will move the train nodes initially onto the rails to achieve the correct initial wheel-rail forces. If the results are viewed with magnified displacements, the initial movements can appear surprising.

Wheel roughness input is available. This will be applied in addition to track roughness. The input curve must continue for the total rolled distance – it is not assumed to repeat with each wheel rotation. This is to avoid problems associated with ensuring continuity between the start and end of the profile around the wheel circumference, especially since the profiles might be generated from roughness spectra rather than taken directly from measured data.

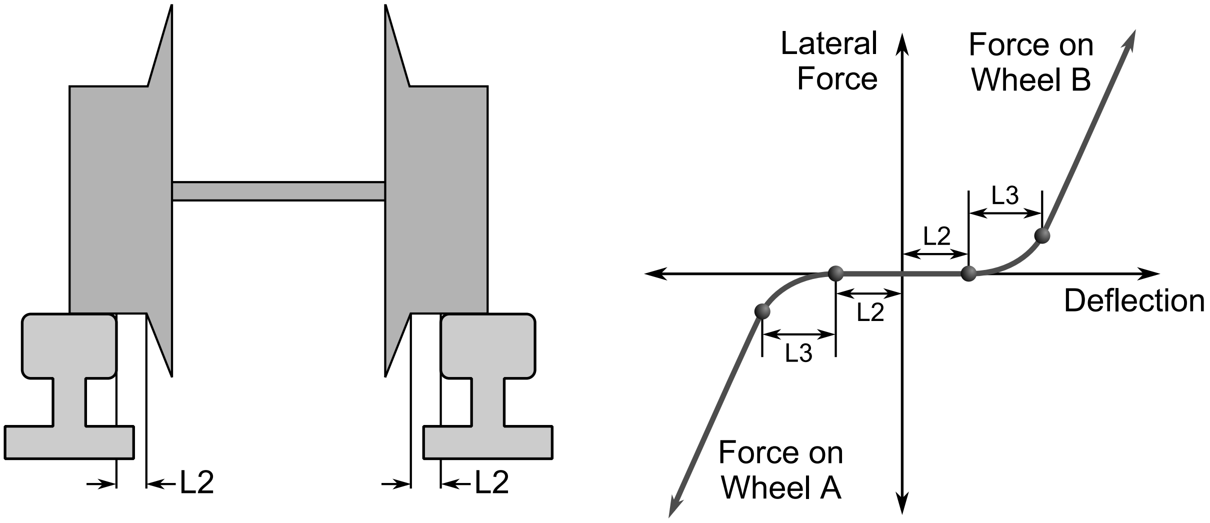


Figure 34-2. Illustration of lateral drift parameters L2 and L3 from *RAIL_-TRACK.

Wheel-rail interface:

The wheel-rail interface model is a simple penalty function designed to ensure that the train nodes follow the line of the track. It does not attempt to account for the shape of the rail profile. Vertical and lateral loads are treated independently. For this reason, the algorithm is not suitable for rail vehicle dynamics calculations.

Wheel-rail contact stiffness is input on *RAIL_TRAIN. For vertical loads, a linear force-deflection relationship is assumed in compression; no tensile force is generated (this corresponds to the train losing contact with the rail). Typical contact stiffness is 2MN/mm. Lateral deflections away from the theoretical centerline of the rail beams are also penalized by a linear force-deflection relationship. The lateral force is applied only to wheels on the side towards which the train has displaced (corresponding to wheel flanges that run inside the rails).

Optionally, a "gap" can be defined in input parameter L2 such that the wheel set can drift laterally by L2 length units before any lateral force is generated. A further option is to allow smooth transition between "gap" and "contact" by means of a transition distance input as parameter L3. Figure 34-2 illustrates the geometry of parameters L2 and L3.

Generally, with straight tracks a simple linear stiffness is sufficient. With curved tracks, a reasonable gap and transition distance should be defined to avoid unrealistic forces being generated in response to small inaccuracies in the distance between the rails. Gravity loading is expected, in order to maintain contact between rail and wheel. This is normally applied by an initial phase of dynamic relaxation. To help achieve convergence quickly, or in some cases avoid the need for dynamic relaxation altogether, the initial force expected on each train node can be input (parameter FINIT on *RAIL_-TRAIN). LS-DYNA positions the nodes initially such that the vertical contact force will be FINIT at each node. If the suspension of the rail vehicles is modeled, it is recommended that the input includes carefully calculated precompression of the spring elements; if this is not done, achieving initial equilibrium under gravity loading can be very time consuming.

The *RAIL algorithm ensures that the train follows the rails, but does not provide forward motion. This is generally applied using *INITIAL_VELOCITY, or for straight tracks, *BOUNDARY_PRESCRIBED_MOTION.

Output:

LS-DYNA generates an additional ASCII output file **train_force_*n***, where ***n*** is an integer updated to avoid overwriting any existing files. The file contains the forces on each train node, output at the same time intervals as the binary time history file (DT on *DATABASE_BINARY_D3THDT).

Checking:

It is recommended that track and train models be tested separately before adding the *RAIL cards. Check that the models respond stably to impulse forces and that they achieve equilibrium under gravity loading. The majority of problems we have encountered have been due to unstable behavior of train or track. Often, these are first detected by the *RAIL algorithm and an error message will result.

*RAIL_TRAIN

Purpose: Define train properties. A train is defined by a set of nodes in contact with a rail defined by *RAIL_TRACK.

Card Sets. For each train include one pair of cards 1 and 2. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	NSETID	(omit)	FINIT	(omit)	TRID	LCUR	OFFS
Type	I	I	F	F	F	I	I	F
Default	none	none	0.0	0.0	0.0	0	none	0.0

Card 2	1	2	3	4	5	6	7	8
Variable	VERTSTF	LATSTF	V2	V3	L2	L3		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	0.0	0.0	0.0		

VARIABLE	DESCRIPTION
ID	Train ID
NSETID	Node set ID containing all nodes that are in contact with rails.
(omit)	Unused variable – leave blank.
FINIT	Estimate of initial vertical force on each wheel (optional) – speeds up the process of initial settling down under gravity loading.
(omit)	Unused variable – leave blank.
TRID	ID of track for this train, see *RAIL_TRACK.

VARIABLE **DESCRIPTION**

LCUR

Load curve ID (see *DEFINE_CURVE) containing wheel roughness (distance of wheel surface away from perfect circle) vs. distance traveled. The curve does not repeat with each rotation of the wheel – the last point should be at a greater distance than the train is expected to travel. Default: no wheel roughness.

OFFS

Offset distance used to generate different roughness curves for each wheel from the roughness curve LCUR. The curve is offset on the x-axis by a different whole number multiple of OFFS for each wheel.

VERTSTF

Vertical stiffness of rail contact.

LATSTF

Lateral stiffness of rail contact.

V2,V3

Unused variables – leave blank.

L2

Lateral clearance from rail to wheel rim. Lateral force is applied to a wheel only when it has moved more than L2 away from the other rail, i.e. the wheel rims are assumed to be near the inner face of the rail.

L3

Further lateral distance before full lateral stiffness applies (force-deflection curve follows a parabola up to this point).

*RIGIDWALL

Two keywords are used in this section to define rigid surfaces:

RIGIDWALL_GEOMETRIC_*OPTION_{OPTION}_{OPTION}}_{OPTION}

RIGIDWALL_PLANAR_*{OPTION}_{OPTION}_{OPTION}

The RIGIDWALL option provides a simple way of treating contact between a rigid surface and nodal points of a deformable body, called slave nodes. Slave nodes which belong to rigid parts are not, in general, checked for contact with only one exception. The RIGIDWALL_PLANAR option may be used with nodal points of rigid bodies if the planar wall defined by this option is fixed in space and the RWPNAL parameter is set to a positive nonzero value on the control card, *CONTROL_CONTACT.

When the rigid wall defined in this section moves with a prescribed motion, the equations of rigid body mechanics are not involved. For a general rigid body treatment with arbitrary surfaces and motion, refer to the *CONTACT_ENTITY definition. The *CONTACT_ENTITY option is for treating contact between rigid and deformable surfaces only.

Energy dissipated due to rigidwalls (sometimes called stonewall energy or rigidwall energy) is computed only if the parameter RWEN is set to 2 in *CONTROL_ENERGY.

*RIGIDWALL_FORCE_TRANSDUCER

Purpose: Define a force transducer for a rigid wall. The output of the transducer is written to the rwforc file.

Card 1	1	2	3	4	5	6	7	8
Variable	TID	RWID						
Type	I	I						
Default	none	none		.				

VARIABLE	DESCRIPTION
TID	Transducer ID.
RWID	Rigid wall ID.

Card 2	1	2	3	4	5	6	7	8
Variable	HEADING							
Type	C							
Default	none							
Remarks	1							

Node Set Cards. For each node set add one card. This input ends at the next keyword ("*").

Card 3	1	2	3	4	5	6	7	8
Variable	NSID							
Type	I							
Default	0.	.						
Remarks								

VARIABLE	DESCRIPTION
NSID	Node set ID.

Remarks:

1. The forces acting on rigid wall RWID are reported separately for each NSID.
2. For rigid walls using the segment option, the forces acting on each segment are reported separately for each NSID.

RIGIDWALL_GEOMETRIC_*OPTION_{OPTION}_{OPTION}_{OPTION}

Available options include:

FLAT

PRISM

CYLINDER

SPHERE

If prescribed motion is desired an additional option is available:

MOTION

One of the shape types **[FLAT, PRISM, CYLINDER, SPHERE]** must be specified, followed by the optional definition of **MOTION**, both on the same line with ***RIGIDWALL_GEOMETRIC.** If an ID number is specified the additional option is available:

ID

If active, the ID card is the first card following the keyword. To view the rigid wall, the option:

DISPLAY

is available. With this option a rigid body is automatically defined which represents the shape, the physical position of the wall, and follows the walls motion if the MOTION option is active. Additional input is optional if DISPLAY is active.

For the CYLINDER and SPHERE, the option:

INTERIOR

is available. Nodes are confined to the interior of these geometric forms.

Purpose: Define a rigid wall with an analytically described form. Four forms are possible. A prescribed motion is optional. For general rigid bodies with arbitrary surfaces and motion, refer to the *CONTACT_ENTITY definition. This option is for treating contact between rigid and deformable surfaces only.

Card Sets. For each rigid wall matching the specified keyword options include one set of the following data cards. This input ends at the next keyword (“*”) card.

ID Card. Additional card for ID keyword option.

Card 1	1	2	3	4	5	6	7	8
Variable	RWID	HEADING						
Type	I	A70						

This heading is picked up by some of the peripheral LS-DYNA codes to aid in post-processing.

VARIABLE	DESCRIPTION
RWID	Rigid wall ID. This must be a unique number.
HEADING	Rigid wall descriptor. It is suggested that unique descriptions be used.

Card 2	1	2	3	4	5	6	7	8
Variable	NSID	NSIDEX	BOXID	BIRTH	DEATH			
Type	I	I	I	F	F			
Default	none	0	0	0.	1.0E+20			

VARIABLE	DESCRIPTION
NSID	Nodal set ID containing slave nodes, see *SET_NODE_*OPTION*: EQ.0: all nodes are slave to rigid wall.
NSIDEX	Nodal set ID containing nodes that exempted as slave nodes, see *SET_NODE_*OPTION*.
BOXID	If defined, only nodes in box are included as slave nodes to rigid wall.
BIRTH	Birth time of rigid wall. The time values of the load curves that control the motion of the wall are offset by the birth time.

VARIABLE	DESCRIPTION
DEATH	Death time of rigid wall. At this time the wall is deleted from the calculation. If dynamic relaxation is active at the beginning of the calculation and if BIRTH = 0.0, the death time is ignored during the dynamic relaxation.

Card 3	1	2	3	4	5	6	7	8
Variable	XT	YT	ZT	XH	YH	ZH	FRIC	
Type	F	F	F	F	F	F	F	
Default	0.	0.	0.	0.	0.	0.	0.	
Remarks								

VARIABLE	DESCRIPTION
XT	*x*-coordinate of tail of any outward drawn normal vector, **n**, originating on wall (tail) and terminating in space (head), see Figure 35-1.
YT	*y*-coordinate of tail of normal vector **n**
ZT	*z*-coordinate of tail of normal vector **n**
XH	*x*-coordinate of head of normal vector **n**
YH	*y*-coordinate of head of normal vector **n**
ZH	*z*-coordinate of head of normal vector **n**
FRIC	Coulomb friction coefficient except as noted below. EQ.0.0: frictionless sliding after contact, EQ.1.0: stick condition after contact.

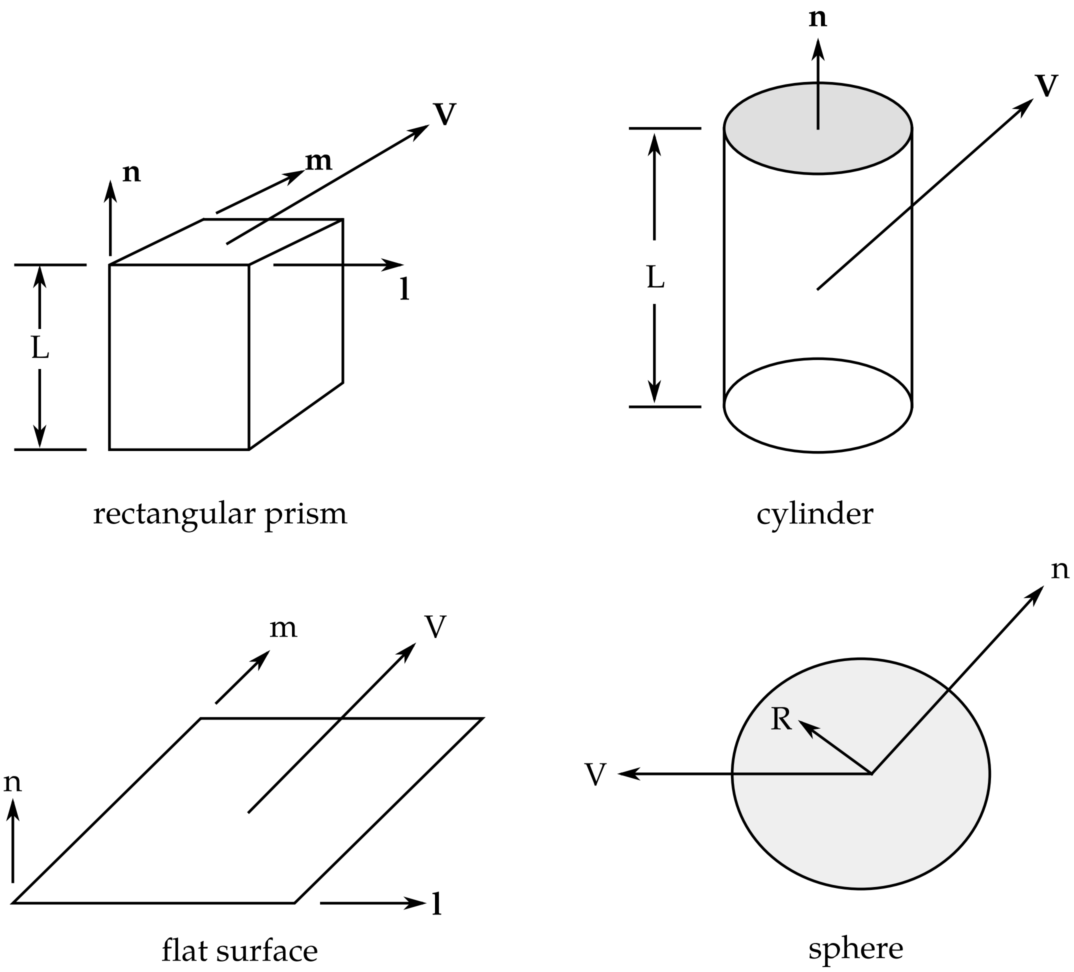


Figure 35-1. Vector **n** determines the orientation of the rigidwall. By including the MOTION option, motion of the rigidwall can be prescribed in any direction **V** as defined by variables VX, VY, VZ.

Flat Rigidwall Card. Card 4 for FLAT keyword option. A plane with a finite size or with an infinite size can be defined, see Figure 35-1. The vector **m** is computed as the vector cross product **n** × **l**. The origin, which is the tail (the start) of the normal vector, is the corner point of the finite size plane.

Card 4	1	2	3	4	5	6	7	8
Variable	XHEV	YHEV	ZHEV	LENL	LENM			
Type	F	F	F	F	F			
Default	0.	0.	0.	infinity	infinity			

VARIABLE	DESCRIPTION
XHEV	*x*-coordinate of head of edge vector **l**, see Figure 35-1.
YHEV	*y*-coordinate of head of edge vector **l**
ZHEV	*z*-coordinate of head of edge vector **l**
LENL	Length of **l** edge. A zero value defines an infinite size plane.
LENM	Length of **m** edge. A zero value defines an infinite size plane.

Prismatic Rigidwall Card. Card 4 for PRISM keyword option. The description of the definition of a plane with finite size is enhanced by an additional length in the direction negative to **n**, see Figure 35-1.

Card 4	1	2	3	4	5	6	7	8
Variable	XHEV	YHEV	ZHEV	LENL	LENM	LENP		
Type	F	F	F	F	F	F		
Default	none	0.	0.	infinity	infinity	infinity		

VARIABLE	DESCRIPTION
XHEV	*x*-coordinate of head of edge vector **l**, see Figure 35-1.
YHEV	*y*-coordinate of head of edge vector **l**
ZHEV	*z*-coordinate of head of edge vector **l**
LENL	Length of **l** edge. A zero value defines an infinite size plane.
LENM	Length of **m** edge. A zero value defines an infinite size plane.
LENP	Length of prism in the direction negative to **n**, see Figure 35-1.

Cylinderical Rigidwall Card. Card 4 for CYLINDER keyword option. The tail of **n** specifies the top plane of the cylinder. The length is defined in the direction negative to **n**. See Figure 35-1.

Card 4	1	2	3	4	5	6	7	8
Variable	RADCYL	LENCYL	NSEGS					
Type	F	F	I					
Default	none	infinity	none					

VARIABLE	DESCRIPTION
RADCYL	Radius of cylinder
LENCYL	Length of cylinder, see Figure 35-1. Only if a value larger than zero is specified is a finite length assumed.
NSEGS	Number of subsections

NSEGS Card. Additional card for NSEGS option.

Card 5	1	2	3	4	5	6	7	8
Variable	VL	HEIGHT						
Type	F	F						
Default	none	none						

VARIABLE	DESCRIPTION
VL	Distance from the Cylinder base
HEIGHT	Section height

Spherical Rigidwall Card. Card 4 for SPHERE keyword option. The center of the sphere is identical to the tail (start) of **n**, see Figure 35-1.

Card 4	1	2	3	4	5	6	7	8
Variable	RADSPH							
Type	F							
Default	0.							

VARIABLE	DESCRIPTION
RADSPH	Radius of sphere

Motion Card. Additional card for motion keyword option.

Card 5	1	2	3	4	5	6	7	8
Variable	LCID	OPT	VX	VY	VZ			
Type	I	I	F	F	F			
Default	none	none	none	none	none			

VARIABLE	DESCRIPTION
LCID	Rigidwall motion curve number, see *DEFINE_CURVE.
OPT	Type of motion: EQ.0: velocity specified, EQ.1: displacement specified.
VX	x-direction cosine of velocity/displacement vector
VY	y-direction cosine of velocity/displacement vector
VZ	z-direction cosine of velocity/displacement vector

Display Card. Optional card for DISPLAY keyword option. If this card is omitted default values are set. The values set here have no effect on the solution other than the PID appearing in the postprocessing.

Card 6	1	2	3	4	5	6	7	8
Variable	PID	RO	E	PR				
Type	I	I	I	F				
Default	none	1.0E-09	1.0E-04	0.3				

VARIABLE	DESCRIPTION
PID	Unique part ID for moving geometric rigid wall. If zero, a part ID will be set that is larger than the maximum of all user defined part ID's.
RO	Density of rigid wall. The default is set to 1.0E-09.
E	Young's modulus. The default is set to 1.0E-04.
PR	Poisson's ratio. The default is set to 0.30.

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$   *RIGIDWALL_GEOMETRIC_SPHERE_MOTION
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Define a rigid sphere:
$     - with a radius of 8
$     - centered at (x,y,z) = (20,20,9)
$     - that moves in the negative z-direction with a specified displacement
$          given by a load curve (load curve: lcid = 5)
$     - which prevents all nodes within a specified box from penetrating the
$          sphere (box number: boxid = 3), these nodes can slide on the sphere
$          without friction
$
*RIGIDWALL_GEOMETRIC_SPHERE_MOTION
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$     nsid    nsidex     boxid
                             3
$
$       xt        yt        zt        xh        yh        zh      fric
      20.0      20.0       9.0      20.0      20.0       0.0       0.0
$
$   radsph
       8.0
$
$     lcid       opt        vx        vy        vz
         5         1       0.0       0.0      -1.0
```

```
$
$
*DEFINE_BOX
$    boxid       xmn       xmx       ymn       ymx       zmn       zmx
         3       0.0      40.0       0.0      40.0      -1.0       1.0
$
$
*DEFINE_CURVE
$     lcid      sidr      scla      sclo      offa      offo
         5
$           abscissa            ordinate
                 0.0                 0.0
              0.0005                15.0
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

*RIGIDWALL_PLANAR_*{OPTION}*_*{OPTION}*_*{OPTION}*

Available options include:

<BLANK>

ORTHO

FINITE

MOVING

FORCES

The ordering of the options in the input below must be observed but the ordering of the options on the command line is unimportant, i.e.; the **ORTHO** card is first, the **FINITE** definition card below must precede the **MOVING** definition card, and the **FORCES** definition card should be last. The **ORTHO** option does not apply if the **MOVING** option is used.

An ID number may be assigned to the rigid wall using the following option:

ID

If this option is active, the ID card is the first card following the keyword.

Display of a non-moving, planar rigid wall is on by default (see SKIPRWG in *CONTROL_CONTACT). The option

DISPLAY

is available for display of moving rigid walls. With this option active, a rigid body is automatically created which represents the shape of the rigid wall and tracks its position without need for additional input. The part ID of the rigid body defaults to RWID if the ID option is active, and if RWID is a unique ID within the set of all part IDs.

Purpose: Define planar rigid walls with either finite (**FINITE**) or infinite size. Orthotropic friction can be defined (**ORTHO**). Also, the plane can possess a mass and an initial velocity (**MOVING**); otherwise, the wall is assumed to be stationary. The **FORCES** option allows the specification of segments on the rigid walls on which the contact forces are computed. In order to achieve a more physical reaction related to the force versus time curve, the SOFT value on the **FORCES** card can be specified.

Card Sets. For each rigid wall matching the specified keyword options include one set of the following data cards. This input ends at the next keyword ("*") card.

ID. Card. Additional card for ID keyword option.

Card 1	1	2	3	4	5	6	7	8
Variable	RWID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
RWID	Rigid wall ID. Up to 8 characters can be used.

Card 2	1	2	3	4	5	6	7	8
Variable	NSID	NSIDEX	BOXID	OFFSET	BIRTH	DEATH	RWKSF	
Type	I	I	I	F	F	F	F	
Default	none	0	0	0.	0.	1.0E+20	1.0	

VARIABLE **DESCRIPTION**

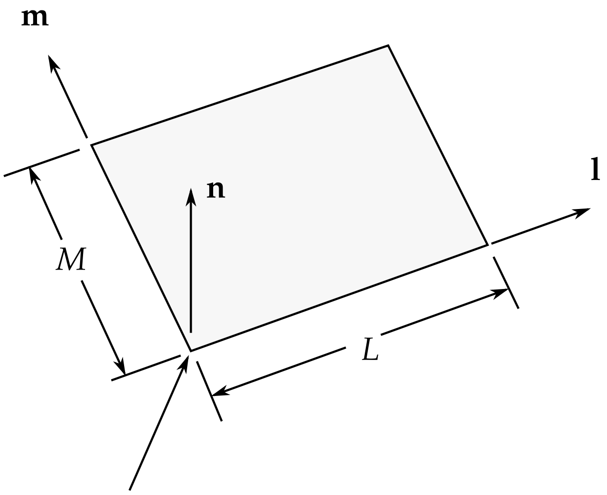
NSID
Nodal set ID containing slave nodes, see *SET_NODE_*OPTION*:

EQ.0: all nodes are slave to rigid wall.

NSIDEX
Nodal set ID containing nodes that exempted as slave nodes, see *SET_NODE_*OPTION*.

BOXID
All nodes in box are included as slave nodes to rigid wall, see *DEFINE_BOX. If options NSID or NSIDEX are active then only the subset of nodes activated by these options are checked to see if they are within the box.

OFFSET
All nodes within a normal offset distance, OFFSET, to the rigid wall are included as slave nodes for the rigid wall. If options NSID, NSIDEX, or BOXID are active then only the subset of nodes activated by these options are checked to see if they are within the offset distance. This option applies to the PLANAR wall only.



Tail of normal vector is the origin and
corner point if extent of stonewall is finite.

Figure 35-2. Vector **n** is normal to the rigidwall. An optional vector **l** can be defined such that **m** = **n** × **l**. The extent of the rigidwall is limited by defining L (LENL) and M (LENM). A zero value for either of these lengths indicates that the rigidwall is infinite in that direction.

VARIABLE	DESCRIPTION
BIRTH	Birth time of rigid wall. The time values of the load curves that control the motion of the wall are offset by the birth time.
DEATH	Death time of rigid wall. At this time the wall is deleted from the calculation. If dynamic relaxation is active at the beginning of the calculation and if BIRTH = 0.0, the death time is ignored during the dynamic relaxation.
RWKSF	Stiffness scaling factor. If RWKSF is also specified in *CONTROL_CONTACT, the stiffness is scaled by the product of the two values.

Card 3	1	2	3	4	5	6	7	8
Variable	XT	YT	ZT	XH	YH	ZH	FRIC	WVEL
Type	F	F	F	F	F	F	F	F
Default	0.	0.	0.	0.	0.	0.	0.	0.

VARIABLE **DESCRIPTION**

XT *x*-coordinate of tail of any outward drawn normal vector, **n**, originating on wall (tail) and terminating in space (head), see Figure 35-2.

YT *y*-coordinate of tail of normal vector **n**

ZT *z*-coordinate of tail of normal vector **n**

XH *x*-coordinate of head of normal vector **n**

YH *y*-coordinate of head of normal vector **n**

ZH *z*-coordinate of head of normal vector **n**

FRIC Coulomb friction coefficient except as noted below.

EQ.0.0: frictionless sliding after contact,

EQ.1.0: no sliding after contact,

EQ.2.0: node is welded after contact with frictionless sliding. Welding occurs if and only if the normal value of the impact velocity exceeds the critical value specified by WVEL.

EQ.3.0: node is welded after contact with no sliding. Welding occurs if and only if the normal value of the impact velocity exceeds the critical value specified by WVEL.

In summary, FRIC could be any positive value. Three special values of FRIC trigger special treatments as follows:

FRIC	1.0	2.0	3.0
Bouncing back from wall	allowed	not allowed	not allowed
Sliding on wall	not allowed	allowed	not allowed

WVEL Critical normal velocity at which nodes weld to wall (FRIC = 2 or 3).

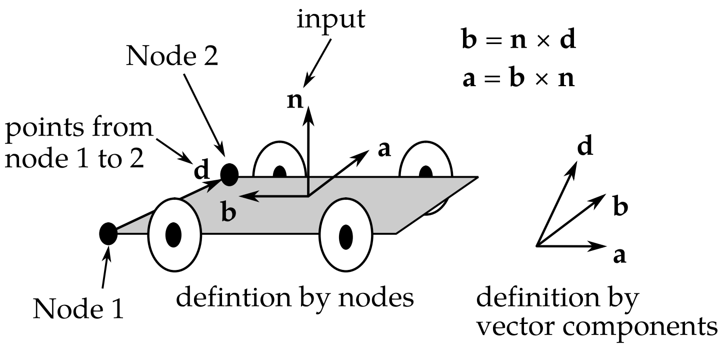


Figure 35-3. Definition of orthotropic friction vectors. The two methods of defining the vector, **d**, are shown. If vector **d** is defined by nodes 1 and 2, the local coordinate system may rotate with the body which contains the nodes; otherwise, **d** is fixed in space, thus on the rigid wall, and the local system is stationary.

Orthotropic Friction Card 1. Additional card for ORTHO keyword option. See Figure 35-3 for the definition of orthotropic friction.

Card 4	1	2	3	4	5	6	7	8
Variable	SFRICA	SFRICB	DFRICA	DFRICB	DECAYA	DECAYB		
Type	F	F	F	F	F	F		
Default	0.	0.	0	0	0.	0.		

Orthotropic Friction Card 2. Additional card for ORTHO keyword option. See Figure 35-3 for the definition of orthotropic friction.

Card 5	1	2	3	4	5	6	7	8
Variable	NODE1	NODE2	D1	D2	D3			
Type	I	I	F	F	F			
Default	0.	0.	0	0	0.			

VARIABLE	DESCRIPTION
SFRICA	Static friction coefficient in local a-direction, μ_{sa}, see Figure 35-3
SFRICB	Static friction coefficient in local b-direction, μ_{sb}

VARIABLE	DESCRIPTION
DFRICA	Dynamic friction coefficient in local a-direction, μ_{ka}
DFRICB	Dynamic friction coefficient in local b-direction, μ_{kb}
DECAYA	Decay constant in local a-direction, d_{ya}
DECAYB	Decay constant in local b-direction, d_{yb}
NODE1	Node 1, alternative to definition with vector **d** below. See Figure 35-3. With the node definition the direction changes if the nodal pair rotates.
NODE2	Node 2
D1	d_1, x-component of vector, alternative to definition with nodes above. See Figure 35-3. This vector is fixed as a function of time.
D2	d_2, y-component of vector
D3	d_3, z-component of vector

Remarks:

1. The coefficients of friction are defined in terms of the static, dynamic and decay coefficients and the relative velocities in the local a and b directions as

$$\mu_a = \mu_{ka} + (\mu_{sa}\mu_{ka})e^{d_{va}V_{\text{relative},a}}$$

$$\mu_b = \mu_{kb} + (\mu_{sb}\mu_{kb})e^{d_{vb}V_{\text{relative},b}}$$

2. Orthotropic rigid walls can be used to model rolling objects on rigid walls where the frictional forces are substantially higher in a direction transverse to the rolling direction. To use this option define a vector $\boldsymbol{d}$ to determine the local frictional directions via:

$$\mathbf{b} = \mathbf{n} \times \mathbf{d}, \qquad \mathbf{a} = \mathbf{b} \times \mathbf{n}$$

where **n** is the normal vector to the rigid wall. If **d** is in the plane of the rigid wall, then **a** is identical to **d**.

Finite Wall Size Card. Additional card for FINITE keyword option. See Figure 35-3 for the definition of orthotropic friction. See Figure 35-2. The $\boldsymbol{m}$ vector is computed as the vector cross product $\boldsymbol{m} = \boldsymbol{n} \times \boldsymbol{l}$. The origin, the tail of the normal vector, is taken as the corner point of the finite size plane.

Card 6	1	2	3	4	5	6	7	8
Variable	XHEV	YHEV	ZHEV	LENL	LENM			
Type	F	F	F	F	F			
Default	0.	0.	0.	infinity	infinity			

VARIABLE	DESCRIPTION
XHEV	x-coordinate of head of edge vector $\boldsymbol{l}$, see Figure 35-2.
YHEV	y-coordinate of head of edge vector $\boldsymbol{l}$
ZHEV	z-coordinate of head of edge vector $\boldsymbol{l}$
LENL	Length of $\boldsymbol{l}$ edge
LENM	Length of $\boldsymbol{m}$ edge

Moving Wall Card. Additional card for MOVING keyword option. Note: The MOVING option is not compatible with the ORTHO option.

Card 7	1	2	3	4	5	6	7	8
Variable	MASS	V0						
Type	F	F						
Default	none	0.						

VARIABLE	DESCRIPTION
MASS	Total mass of rigidwall
V0	Initial velocity of rigidwall in direction of defining vector, **n**

Forces Card. Additional card for FORCES keyword option. This option allows the force distribution to be monitored on the plane. Also four points can be defined for visualization of the rigid wall. A shell or membrane element must be defined with these four points as the connectivity for viewing in LS-PREPOST.

Card 7	1	2	3	4	5	6	7	8
Variable	SOFT	SSID	N1	N2	N3	N4		
Type	I	I	I	I	I	I		
Default	0	0	0	0	0	0		
Remarks		1	2					

VARIABLE	DESCRIPTION
SOFT	Number of cycles to zero relative velocity to reduce force spike
SSID	Segment set identification number for defining areas for force output, see *SET_SEGMENT and remark 1 below.
N1-N4	Optional node for visualization

Remarks:

1. The segment set defines areas for computing resultant forces. These segments translate with the moving rigidwall and allow the forced distribution to be determined. The resultant forces are written in file "RWFORC."

2. These four nodes are for visualizing the movement of the wall, i.e., they move with the wall. To view the wall in LS-PREPOST it is necessary to define a single shell element with these four nodes as its connectivity. The single element must be deformable (non rigid) or else the segment will be treated as a rigid body and the nodes will have their motion modified independently of the rigidwall.

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$   *RIGIDWALL_PLANAR_MOVING_FORCES
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Define a moving planar rigid wall:
$     - that is parallel to the y-z plane starting at x = 250 mm
$     - with an initial velocity of 8.94 mm/ms in the negative z-direction
$     - that has a mass of 800 kg
$     - which prevents all nodes in the model from penetrating the wall
$     - with a friction coefficient for nodes sliding along the wall of 0.1
$     - track the motion of the wall by creating a node (numbered 99999)
$         at the tail of the wall and assigning the node to move with the wall
$
*RIGIDWALL_PLANAR_MOVING_FORCES
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$     nsid    nsidex     boxid
         0         0         0
$
$       xt        yt        zt        xh        yh        zh      fric
     250.0       0.0       0.0       0.0       0.0       0.0       0.1
$
$  SW mass    SW vel
    800.00      8.94
$
$     soft      ssid     node1     node2     node3     node4
         0         0     99999
$
$
*NODE
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$    nid               x               y               z      tc      rc
   99999           250.0             0.0             0.0       0       0
$
$
*DATABASE_HISTORY_NODE
$   Define nodes that output into nodout
$      id1       id2       id3
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
     99999
$
*DATABASE_NODOUT
$       dt
       0.1
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

*SECTION

In this section, the element formulation, integration rule, nodal thicknesses, and cross sectional properties are defined. All section identifiers (SECID's) defined in this section must be unique, i.e., if a number is used as a section ID for a beam element then this number cannot be used again as a section ID for a solid element. The keyword cards in this section are defined in alphabetical order:

*SECTION_ALE1D

*SECTION_ALE2D

SECTION_BEAM_*{OPTION}

*SECTION_BEAM_AISC

*SECTION_DISCRETE

*SECTION_POINT_SOURCE

*SECTION_POINT_SOURCE_MIXTURE

*SECTION_SEATBELT

SECTION_SHELL_*{OPTION}

SECTION_SOLID_*{OPTION}

SECTION_SPH_*{OPTION}

*SECTION_TSHELL

The location and order of these cards in the input file are arbitrary.

An additional option **TITLE** may be appended to all the ***SECTION** keywords. If this option is used then an addition line is read for each section in 80a format which can be used to describe the section. At present LS-DYNA does make use of the title. Inclusion of titles gives greater clarity to input decks.

*SECTION_ALE1D

Purpose: Define section properties for 1D ALE elements

Card Sets. For each ALE1D section add one pair of cards 1 and 2. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	SECID	ALEFORM	AET	ELFORM				
Type	I/A	I	I	I				
Default	none	none	0	none				

Card 2	1	2	3	4	5	6	7	8
Variable	THICK	THICK						
Type	F	F						
Default	none	none						

VARIABLE	DESCRIPTION
SECID	Section ID. SECID is referenced on the *PART card. A unique number or label must be specified.
ALEFORM	ALE formulation: EQ.11: Multi-Material ALE formulation.
AET	Ambient Element Type EQ.4: Pressure inflow
ELFORM	Element formulation: EQ.7: Plane strain EQ.8: Axisymmetric (per radian) EQ.-8: spherical (per unit of solid angle)

VARIABLE	DESCRIPTION
THICK	Nodal thickness. See Remark 1

Remarks:

1. *SECTION_ALE1D is using the common *SECTION_BEAM reader which expects two thickness values. However, the ALE 1D will simply take the average of these two values as the beam thickness.

   The thickness is not used for ELFORM = -8 but the reader routine expects values on the 2nd line.

*SECTION_ALE2D

Purpose: Define section properties for 2D ALE elements. This supersedes the old way of defining section properties for 2D ALE elements via *SECTION_SHELL.

For coupling between 2D Lagrangian elements and 2D ALE elements, use *CONSTRAINED_LAGRANGE_IN_SOLID rather than *CONTACT_2D_AUTOMATIC_SURFACE_IN_CONTINUUM.

In the case of an axisymmetric analysis, ELFORM for *SECTION_ALE2D can only be set to 14 (area-weighted). In the same analysis, axisymmetric Lagrangian elements are not restricted to an area-weighted formulation. In other words, shell formulation 14 or 15 are permitted for Lagrangian shells and beam formulation 8 is permitted for Lagrangian beams. Coupling forces between the axisymmetric ALE elements and axisymmetric Lagrangian elements are automatically adjusted as needed.

Section Cards. For each ALE2D section include a card. This input terminates at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	SECID	ALEFORM	AET	ELFORM				
Type	I/A	I	I	I				
Default	none	none	0	none				

VARIABLE	DESCRIPTION
SECID	Section ID. SECID is referenced on the *PART card. A unique number or label must be specified.
ALEFORM	ALE formulation: EQ.11: Multi-Material ALE formulation.

VARIABLE	DESCRIPTION
AET	Part type flag EQ.0: This is a regular or non-ambient part (default) EQ.4: Reservoir or ambient type part EQ.5: Reservoir or ambient type part, but only used together with *LOAD_BLAST_ENHANCED (LBE). It defines this part as an “ambient receptor part” for the transient blast load supplied by a corresponding LBE KW (see *LOAD_BLAST_ENHANCED, available only for ALEFORM = 11).
ELFORM	Element formulation: EQ.13: Plane strain (x-y plane) EQ.14: Axisymmetric solid (x-y plane, y-axis of symmetry) – area weighted

SECTION_BEAM_*{OPTION}

Available options include:

<BLANK>

AISC

such that the keyword cards appear:

*SECTION_BEAM

*SECTION_BEAM_AISC

Purpose: Define cross sectional properties for beam, truss, discrete beam, and cable elements.

The AISC option may be used to specify standard steel sections as specified by the American Institute of Steel Construction, and is described separately after *SECTION_-BEAM

Card Sets. For each BEAM section in the model add one set of the following 2 (maybe 3 for ELFORM = 12) cards. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	SECID	ELFORM	SHRF	QR/IRID	CST	SCOOR	NSM	
Type	I/A	I	F	F	F	F	F	
Default	none	1	1.0	2.0	0.0	0.0	0.0	

Integrated Beam Card (types 1 and 11). Card 2 for ELFORM set to either type 1 or 11.

Card 2	1	2	3	4	5	6	7	8
Variable	TS1	TS2	TT1	TT2	NSLOC	NTLOC		
Type	F	F	F	F	F	F		

Resultant Beam With Shape Card (types 2, 3, and 12). Card 2 for ELFORM equal to 2, 3, or 12 and when first 7 characters of the card spell out "SECTION".

Card 2	1	2	3	4	5	6	7	8
Variable	STYPE	D1	D2	D3	D4	D5	D6	
Type	A10	F	F	F	F	F		

Resultant Beam Card 1 (types 2, 12, and 13). Card 2 for ELFORM equal to 2, 12, or 13 and when first 7 characters of card *do not* spell "SECTION".

Card 2	1	2	3	4	5	6	7	8
Variable	A	ISS	ITT	J	SA	IST		
Type	F	F	F	F	F	F		

Resultant Beam Card 2 (type 12 only). *Card 3* for ELFORM equal to 12 and when first 7 characters of *card 2 do not* spell "SECTION".

Card 3	1	2	3	4	5	6	7	8
Variable	YS	ZS	IYR	IZR	IRR	IW	IWR	
Type	F	F	F	F	F	F	F	

Resultant Beam Card (type 3). Card 2 for ELFORM equal to 3.

Card 2	1	2	3	4	5	6	7	8
Variable	A	RAMPT	STRESS					
Type	F	F	F					

Integrated Beam Card (types 4 and 5). Card 2 for ELFORM equal to 4 or 5.

Card 2	1	2	3	4	5	6	7	8
Variable	TS1	TS2	TT1	TT2				
Type	F	F	F	F				

Discrete Beam Card (type 6). Card 2 for ELFORM equal to 6 for any material *other* than material type 146.

Card 2	1	2	3	4	5	6	7	8
Variable	VOL	INER	CID	CA	OFFSET	RRCON	SRCON	TRCON
Type	F	F	F	F	F	F	F	F

Discrete Beam Card (type 6, mat 146). Card 2 for ELFORM equal to 6 for material type 146.

Card 2	1	2	3	4	5	6	7	8
Variable	VOL	INER	CID	DOFN1	DOFN2			
Type	F	F	F	F	F			

2D Shell Card (types 7 and 8). Card 2 for ELFORM equal to 7 or 8.

Card 2	1	2	3	4	5	6	7	8
Variable	TS1	TS2	TT1	TT2				
Type	F	F	F	F				

Spot Weld Card (type 9). Card 2 for ELFORM equal to 9.

Card 2	1	2	3	4	5	6	7	8
Variable	TS1	TS2	TT1	TT2	PRINT			
Type	F	F	F	F	F			

Integrated Beam Card (types 14). Card 2 for ELFORM equal to 14.

Card 2	1	2	3	4	5	6	7	8
Variable	PR	IOVPR	IPRSTR					
Type	F	F	F	F				

VARIABLE **DESCRIPTION**

SECID Section ID. SECID is referenced on the *PART card. A unique number or label must be specified.

ELFORM Element formulation options:

EQ.1: Hughes-Liu with cross section integration (default),

EQ.2: Belytschko-Schwer resultant beam (resultant),

EQ.3: truss (resultant). See Remark 2.

EQ.4: Belytschko-Schwer full cross-section integration,

EQ.5: Belytschko-Schwer tubular beam with cross-section integration,

EQ.6: discrete beam/cable,

EQ.7: 2D plane strain shell element (*xy* plane),

EQ.8: 2D axisymmetric volume weighted shell element (*xy* plane, *y*-axis of symmetry),

EQ.9: spotweld beam, see *MAT_SPOTWELD.

EQ.11: integrated warped beam. See Remark 4)

EQ.12: resultant warped beam

EQ.13: small displacement, linear Timoshenko beam with exact stiffness. See Remark 6

EQ.14: Elbow integrated tubular beam element. An user

VARIABLE **DESCRIPTION**

defined integration rule with tubular cross section (9) must be used.

Note that the 2D and 3D element types must not be mixed, and different types of 2D elements must not be used together. For example, the plane strain element type must not be used with the axisymmetric element type. In 3D the different beam elements types, i.e., 1-6 and 9 can be freely mixed together.

SHRF

Shear factor. This factor is not needed for truss, resultant beam, discrete beam, and cable elements. The recommended value for rectangular sections is 5/6, the default is 1.0.

QR/IRID

Quadrature rule or rule number for user defined rule for integrated beams. See Remark 10 regarding beam formulations 7 and 8.

EQ.1.0: one integration point,

EQ.2.0: 2 × 2 Gauss quadrature (default beam),

EQ.3.0: 3 × 3 Gauss quadrature,

EQ.4.0: 3 × 3 Lobatto quadrature,

EQ.5.0: 4 × 4 Gauss quadrature

EQ.-n: where |n| is the number of the user defined rule. IRID integration rule n is defined using *INTEGRATION_BEAM card.

CST

Cross section type, not needed for truss, resultant beam, discrete beam, and cable elements:

EQ.0.0: rectangular,

EQ.1.0: tubular (circular only),

EQ.2.0: arbitrary (user defined integration rule).

SCOOR

Affects the discrete beam formulation (see Remark 7) and also the update of the local coordinate system of the discrete beam element. This parameter does not apply to cable elements. The force and moment resultants in the output databases are output in the local coordinate system. See Remark 9 for more on the local coordinate system update.

EQ.-13.0: Like -3.0, but with correction for beam rotation

VARIABLE **DESCRIPTION**

EQ.-12.0: Like -2.0, but with correction for beam rotation

EQ.-3.0: beam node 1, the angular velocity of node 1 rotates triad,

EQ.-2.0: beam node 1, the angular velocity of node 1 rotates triad but the r-axis is adjusted to lie along the line between the two beam nodal points. This option is not recommended for zero length discrete beams.,

EQ.-1.0: beam node 1, the angular velocity of node 1 rotates triad,

EQ.0.0: centered between beam nodes 1 and 2, the average angular velocity of nodes 1 and 2 is used to rotate the triad,

EQ.+1.0: beam node 2, the angular velocity of node 2 rotates triad.

EQ.+2.0: beam node 2, the angular velocity of node 2 rotates triad. but the r-axis is adjusted to lie along the line between the two beam nodal points. This option is not recommended for zero length discrete beams.

EQ.+3.0: beam node 2, the angular velocity of node 2 rotates triad.

EQ.+12.0: Like +2.0, but with correction for beam rotation

EQ.+13.0: Like +3.0, but with correction for beam rotation

NSM Nonstructural mass per unit length. This option applies to beam types 1-5 and does not apply to discrete, 2D, and spotweld beams, respectively.

TS1 Beam thickness (CST = 0.0, 2.0) or outer diameter (CST = 1.0) in s direction at node $n_!$. Note that the thickness defined on the *ELEMENT_BEAM_THICKNESS card overrides the definition give here. Thickness at node n_1 for beam formulations 7 and 8.

TS2 Beam thickness (CST = 0.0, 2.0) or outer diameter (CST = 1.0) in s direction at node n_2. For truss elements only, it is the ramp up time for the stress initialization by dynamic relaxation. Thickness at node n_2 for beam formulations 7 and 8.

VARIABLE **DESCRIPTION**

TT1

Beam thickness (CST = 0.0, 2.0) or inner diameter (CST = 1.0) in t direction at node n_1. For truss elements only, it is the stress for the initialization of the stress by dynamic relaxation. Not used by beam formulations 7 and 8.

TT2

Beam thickness (CST = 0.0, 2.0) or inner diameter (CST = 1.0) in t direction at node n_2. Not used by beam formulations 7 and 8.

NSLOC

Location of reference surface normal to s axis for Hughes-Liu beam elements only. See Remark 5.

EQ.1.0: side at $s = 1.0$,

EQ.0.0: center,

EQ.-1.0: side at $s = -1.0$.

NTLOC

Location of reference surface normal to t axis for Hughes-Liu beam elements only. See Remark 5.

EQ.1.0: side at $t = 1.0$,

EQ.0.0: center,

EQ.-1.0: side at $t = -1.0$.

A

Cross-sectional area. The definition on *ELEMENT_BEAM_THICKNESS overrides the value defined here.

ISS

I_{ss}, area moment of inertia about local s-axis. The definition on *ELEMENT_BEAM_THICKNESS overrides the value defined here.

ITT

I_{tt}, area moment of inertia about local t-axis. The definition on *ELEMENT_BEAM_THICKNESS overrides the value defined here.

J

J, torsional constant. The definition on *ELEMENT_BEAM_THICKNESS overrides the value defined here If J is zero, then J is reset to the sum of ISS + ITT as an approximation for warped beam.

SA

Shear area. The definition on *ELEMENT_BEAM_THICKNESS overrides the value defined here.

VARIABLE **DESCRIPTION**

IST

I_{st}, product area moment of inertia w.r.t. local s- and t-axis. This is only non-zero for asymmetric cross sections and it can take positive and negative values, e.g. it is negative for SECTION_03.

YS

s coordinate of shear center of cross-section. (The coordinate system is located at the centroid.)

ZS

t coordinate of shear center of cross-section. (The coordinate system is located at the centroid.)

IYR

$\int_A sr^2 dA$, where $r^2 = s^2 + t^2$

IZR

$\int_A tr^2 dA$, where $r^2 = s^2 + t^2$

IRR

$\int_A r^4 dA$, where $r^2 = s^2 + t^2$

IW

Warping constant. $\int_A \omega^2 dA$, where ω is the sectorial area.

IWR

$\int_A \omega r^2 dA$

PR

Pressure inside ELBOW elements that belong to the section. The pressure acts as a stiffener and will reduce the ovalization of the pipe. Pressure acting on the inside wall is taken as positive.

IOVPR

Print flag for the ELBOW ovalization degrees of freedom.

EQ.1.0: an ascii file named elbwov is created and filled with the ovalization. Default no file is created.

IPRSTR

Flag for adding stress due to pressure PR into the material routine.

EQ.0: No stress is added to the material. In this case the pressure only acts as a stiffener for the tube.

EQ.1: The pressure PR is used to calculate additional axial and circumferential stresses due to the applied pressure PR. The stress added is given by:

$$\sigma_{\text{axial}} = \text{PR} \times \frac{D}{4T}, \qquad \sigma_{\text{circ}} = \text{pr} \times \frac{D}{2T}$$

for a straight pipe, and

$$\sigma_{\text{axial}} = \text{PR} \times \frac{D}{4T}, \qquad \sigma_{\text{circ}} = \text{PR} \times \frac{D}{4T} \frac{[2R + r\cos(\theta)]}{[R + r\cos(\theta)]}$$

for a curved pipe. D is the pipe diameter, T is the thickness, R is

VARIABLE **DESCRIPTION**

the curvature of the bend, r is the pipe radius (mean) and θ is an angle pointing out a point on the pipe. EQ.1 also includes the stiffening effect.

RAMPT

Optional ramp-up time for dynamic relaxation. At the end of the ramp-up time, a uniform stress, STRESS, will exist in the truss in the truss element. This option will not work for hyperelastic materials.

STRESS

Optional initial stress for dynamic relaxation. At the end of dynamic relaxation a uniform stress equal to this value should exist in the truss element.

STYPE

Section type (A format) of resultant beam, see Figure 36-1:

EQ.SECTION_01: I-Shape
EQ.SECTION_02: Channel
EQ.SECTION_03: L-shape
EQ.SECTION_04: T-shape
EQ.SECTION_05: Tubular box
EQ.SECTION_06: Z-Shape
EQ.SECTION_07: Trapezoidal
EQ.SECTION_08: Circular
EQ.SECTION_09: Tubular
EQ.SECTION_10: I-Shape 2
EQ.SECTION_11: Solid Box
EQ.SECTION_12: Cross
EQ.SECTION_13: H-Shape
EQ.SECTION_14: T-Shape 2
EQ.SECTION_15: I-Shape 3
EQ.SECTION_16: Channel 2
EQ.SECTION_17: Channel 3
EQ.SECTION_18: T-Shape 3
EQ.SECTION_19: Box-Shape 2
EQ.SECTION_20: Hexagon
EQ.SECTION_21: Hat-Shape
EQ.SECTION_22: Hat-Shape 2

D1-D6

Input parameters for section option using STYPE above.

VOL

Volume of discrete beam and scalar (MAT_146) beam. Used in calculating mass. If VOL = 0 for cable elements, volume is calculated as the product of cable length and cable area. If the mass density of the material model for the discrete beam is set to unity, the magnitude of the lumped mass can be defined here instead. This lumped mass is partitioned to the two nodes of the beam element. The translational time step size for the type 6 beam is dependent on the volume, mass density, and the translational stiffness values, so it is important to define this parameter. Defining the volume is also essential for mass scaling if the type 6 beam controls the time step size.

VARIABLE **DESCRIPTION**

INER

Mass moment of inertia for the six degree of freedom discrete beam and scalar (MAT_146) beam. This parameter does not apply to cable elements. This lumped inertia is partitioned to the two nodes of the beam element. The rotational time step size for the type 6 beam is dependent on the lumped inertia and the rotational stiffness values, so it is important to define this parameter if the rotational springs are active. Defining the rotational inertia is also essential for mass scaling if the type 6 beam rotational stiffness controls the time step size. It is recommended to always set this parameter to a reasonable nonzero value to avoid instabilities and/or having model dependent rotational inertia properties, if the value set is smaller than that of an equivalent solid sphere LS-DYNA will issue a warning.

CID

Coordinate system ID for orientation (material types 66-69, 93, 95, 97), see *DEFINE_COORDINATE_option. If CID = 0, a default coordinate system is defined in the global system or on the third node of the beam, which is used for orientation. This option is not defined for material types than act between two nodal points, such as cable elements. The coordinate system rotates with the discrete beam, see SCOOR above.

CA

Cable area. See material type 71, *MAT_CABLE_DISCRETE_BEAM.

OFFSET

Optional offset for cable. See material type 71, *MAT_CABLE_DISCRETE_BEAM.

RRCON

r-rotational constraint for local coordinate system (see Remark 8)

EQ.0.0: Coordinate ID rotates about r axis with nodes.

EQ.1.0: Rotation is constrained about the r-axis

SRCON

s-rotational constraint for local coordinate system (see Remark 8)

EQ.0.0: Coordinate ID rotates about s axis with nodes.

EQ.1.0: Rotation is constrained about the s-axis

TRCON

t-rotational constraint for local coordinate system (see Remark 8)

EQ.0.0: Coordinate ID rotates about t axis with nodes.

EQ.1.0: Rotation is constrained about the t-axis

VARIABLE **DESCRIPTION**

CID

Coordinate system ID for orientation, material type 146, see *DEFINE_COORDINATE_SYSTEM. If CID = 0, a default coordinate system is defined in the global system.

DOFN1

Active degree-of-freedom at node 1, a number between 1 and 6 where 1 in x-translation and 4 is x-rotation.

DOFN2

Active degree-of-freedom at node 2, a number between 1 and 6.

PRINT

Output spot force resultant from spotwelds.

EQ.0.0: Data is output to swforc file.

EQ.1.0: Output is suppressed.

Remarks:

1. **Implicit Time Integrator.** For implicit calculations all of the beam element choices are implemented:

2. **Truss Elements.** For the truss element, define the cross-sectional area, A, only.

3. **Local Coordinate System Rotation.** The local coordinate system rotates as the nodal points that define the beam rotate. In some cases this may lead to unexpected results if the nodes undergo significant rotational motions. In the definition of the local coordinate system using *DEFINE_COORDINATE_NODES, if the option to update the system each cycle is active then this updated system is used. This latter technique seems to be more stable in some applications.

4. **Integrated Warped Beam.** The integrated warped beam (type 11) is a 7 degree of freedom beam that must be used with an integration rule of the open standard cross sections, see *INTEGRATION_BEAM. To incorporate the additional degrees of freedom corresponding to the twist rates, the user should declare one scalar node (*NODE_SCALAR) for each node attached to a warped beam. This degree of freedom is associated to the beam element using the warpage option on the *ELEMENT_BEAM card.

5. **Beam Offsets.** Beam offsets are sometimes necessary for correctly modeling beams that act compositely with other elements such as shells or other beams. A beam offset extends from the beam's n_1-to-n_2 axis to the reference axis of the beam. The beam reference axis lies at the origin of the local s and t axes. This origin is located at the center of the cross-section footprint for beam formulations 1 and 11 but it is located at the cross-section centroid for beam formula-

tion 2. Note that for cross-sections that are not doubly symmetric, e.g, a T-section, the center of the cross-section footprint and the centroid of the cross-section do not coincide. The offset in the positive *s*-direction is

$$\begin{aligned}\text{s-offset} &= -0.5 \times \text{ NSLOC} \\ &\qquad \times \text{(beam cross-section dimension in } s\text{-direction)}.\end{aligned}$$

Similarly, the offset in the positive t-direction is

$$\begin{aligned}\text{t-offset} &= -0.5 \times \text{ NTLOC} \\ &\qquad \times \text{(beam cross-section dimension in } t\text{-direction)}.\end{aligned}$$

If IRID is used to point to an integration rule with ICST > 0, then offsets must be defined using SREF and TREF on the *INTEGRATION_BEAM card as they will override NSLOC and NTLOC even if SREF = 0 or TREF = 0. See also *ELEMENT_BEAM_OFFSET for an alternate approach to defining beam offsets.

6. **3-D Timoshenko Beam.** Element type 13 is a 3-D Timoshenko resultant-based beam element with two nodes for small displacement, linear isotropic elasticity. The stiffness matrix is identical to the residual stiffness formulation used in the Belytschko-Schwer element (type 2). This element only works with *MAT_-ELASTIC. It uses the reference geometry to calculate the element stiffness and calculates the element forces by multiplying the element stiffness by the displacements. Offsets work but they are fixed for all time like the reference geometry.

7. **SCOOR.** If the magnitude of SCOOR is less than or equal to unity then zero length discrete beams are assumed with infinitesimal separation between the nodes in the deformed state. For large separations or nonzero length beams set |SCOOR| to 2, 3, 12, or 13, in which case true beam-like behavior is invoked to provide equilibrating torques to offset any force couples that arise due to translational stiffness or translational damping. Also, rigid body rotation is measured and the spring strain modified so that rotation does not create strain. A flaw in this strain modification was found in the implementation of |SCOOR| = 2 and 3 the improved formulation is activated by setting |SCOOR| = 12 and 13. The original options were left in place to allow legacy data to run without change.

8. **Disabling Nodal Rotations.** RRCON, SRCON, and TRCON are optional and apply only to non-cable discrete beams. If set to 1, RRCON, SRCON, and TRCON will prevent nodal *rotations* about the local *r*, *s*, *t* axes, respectively, from affecting the update of the local coordinate system. These three parameters have no influence on how nodal translations may affect the local coordinate system update.

9. **Note about Local Coordinate Updates and FLAG.** If CID is nonzero for a discrete beam and the coordinate system identified by CID uses *DEFINE_COORDINATE_NODES with FLAG=1, the beam local system is updated based on the current orientation of the three nodes identified in *DEFINE_COORDINATE_NODES. In this case, local coordinate system updates per SCOOR types 0, ±1, ±3, and ±13 are inactive while for SCOOR types ±2 and ±12, a final adjustment is made to the local coordinate system so that the local *r*-axis lies along the n_1-to-n_2 axis of the beam. An optional output database (*DATABASE_DISBOUT) will report relative displacements, rotations, and force resultants of discrete beams, all in the local coordinate system.

10. **Beams 7 and 8.** Beam formulations 7 and 8 are 2D shell elements. For these two formulations, variable QR/IRID is the number of through thickness integration points for the shell. Output for these integration points is controlled by the variable BEAMIP in *DATABASE_EXTENT_BINARY.

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$   *SECTION_BEAM
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Define a Belytschko-Schwer resultant beam (elform = 2) with the following
$  properties.  This beam models the connection/stiffening beams of a medium
$  size roadside sign.
$
$    cross sectional area:                       a =     515.6 mm2
$    2nd moment of area about s-axis:          iss =  99,660.0 mm4
$    2nd moment of area about t-axis:          iss =  70,500.0 mm4
$    2nd polar moment of area about beam axis:   j = 170,000.0 mm4
$
*SECTION_BEAM
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      sid    elform      shrf   qr/irid       cst
       111         2
$
$        a       iss       itt         j        sa
     515.6   99660.0   70500.0  170000.0
$
*SECTION_BEAM_TITLE
    Main beam member
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      sid    elform      shrf   qr/irid       cst
       111         2
$
$        a       iss       itt         j        sa
     515.6   99660.0   70500.0  170000.0
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

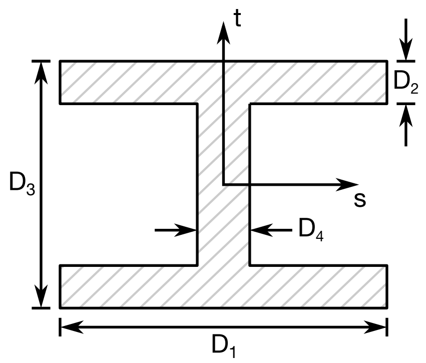


Figure 36-1. SECTION_01 ⇒ I-Shape

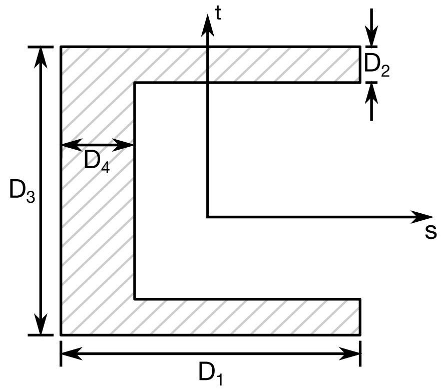


Figure 36-2. SECTION_02 ⇒ Channel

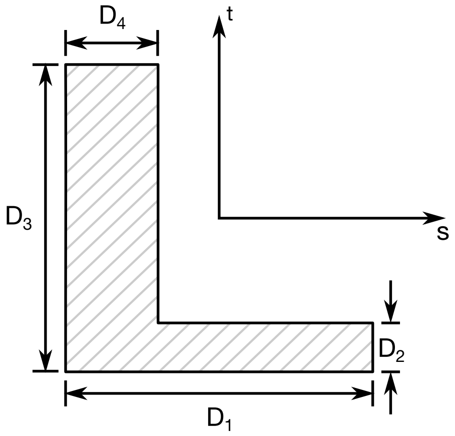


Figure 36-3. SECTION_03 ⇒ L-Shape

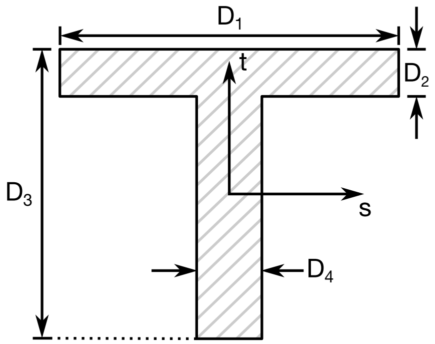


Figure 36-4. SECTION_04 ⇒ T-Shape

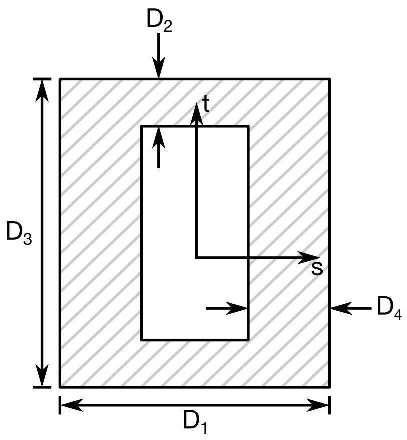


Figure 36-5. SECTION_05 ⇒ Box-Shape

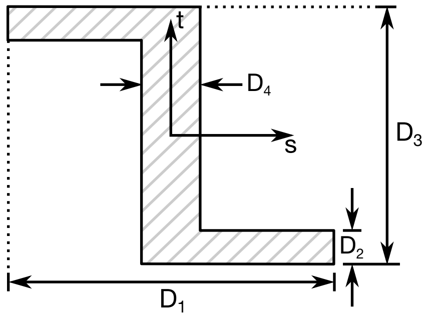


Figure 36-6. SECTION_06 ⇒ Z-Shape

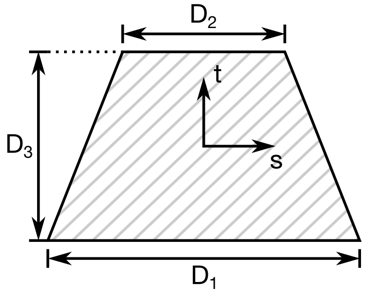


Figure 36-7. SECTION_07 ⇒ Trapezoidal-Shape

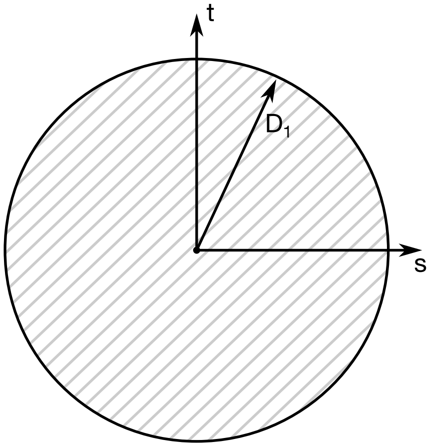


Figure 36-8. SECTION_08 ⇒ Circular

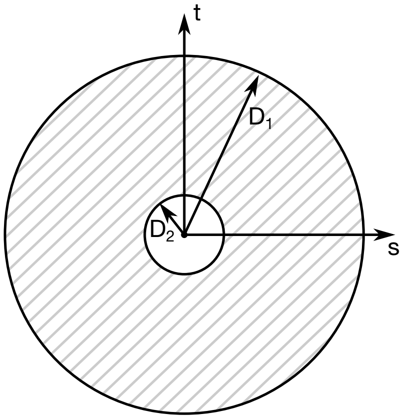


Figure 36-9. SECTION_09 ⇒ Tubular

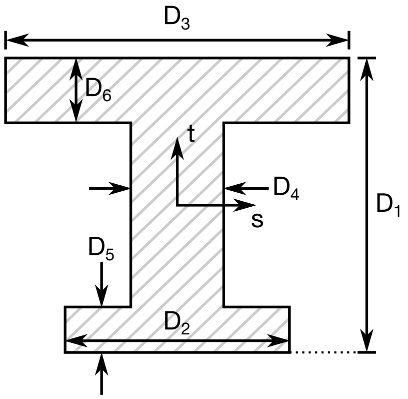
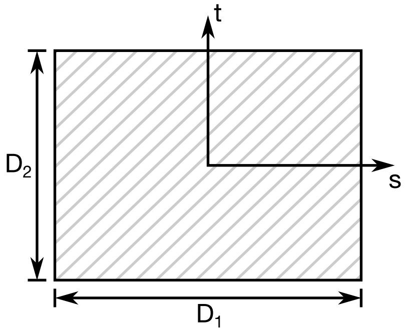


Figure 36-10. SECTION_10 ⇒ I-Shape 2



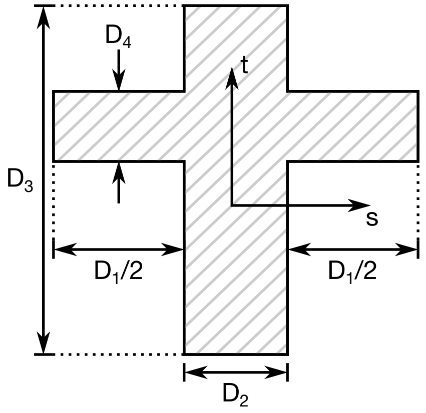


Figure 36-11. SECTION_11 ⇒ Solid Box

Figure 36-12. SECTION_12 ⇒ Cross

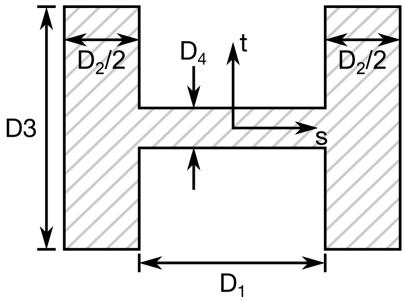


Figure 36-13. SECTION_13 ⇒ H-Shape

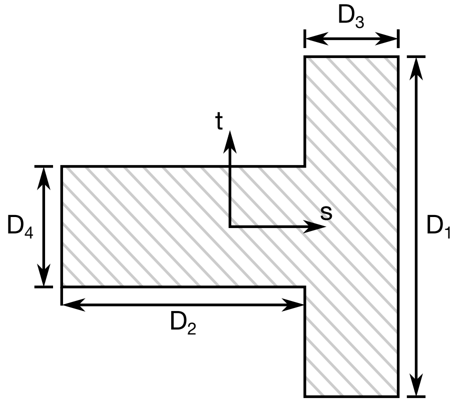


Figure 36-14. SECTION_14 ⇒ T-Shape 2

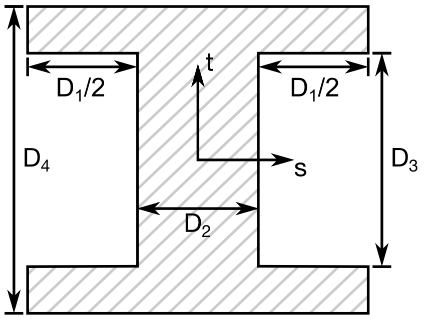


Figure 36-15. SECTION_15 ⇒ I-Shape 3

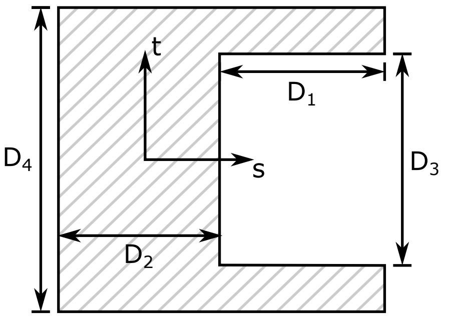
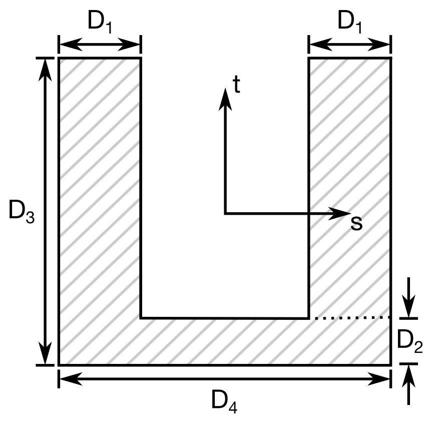


Figure 36-16. SECTION_16 ⇒ Channel 2



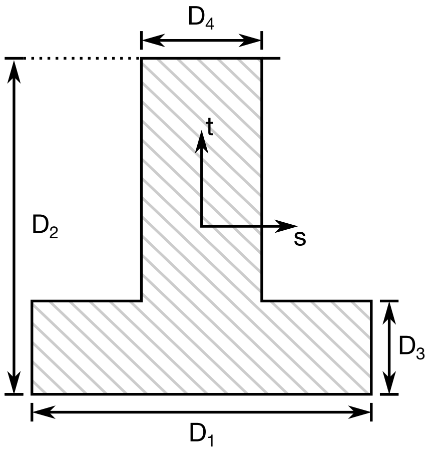


Figure 36-17. SECTION_17 ⇒ Channel 3 **Figure 36-18.** SECTION_18 ⇒ T-Shape 3

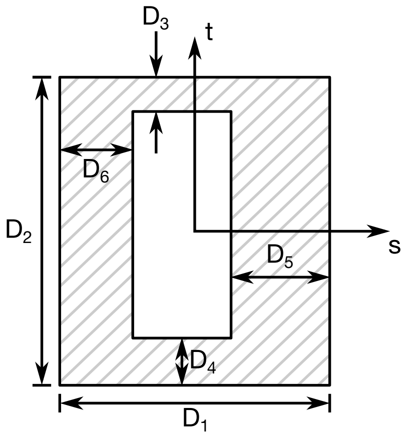


Figure 36-19. SECTION_19 ⇒ Box-Shape

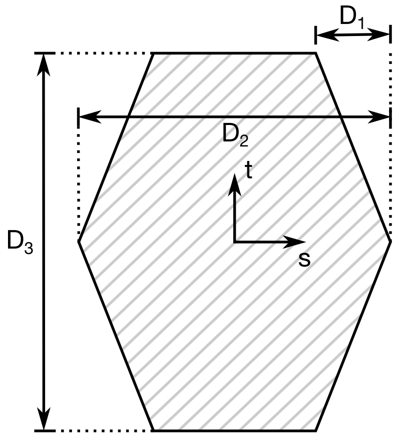


Figure 36-20. SECTION_20 ⇒ Hexagon

2

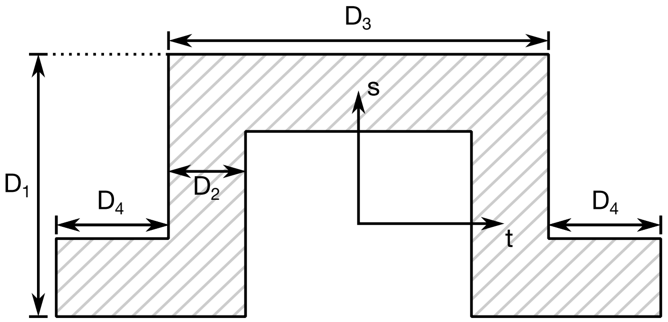


Figure 36-21. SECTION_21 ⇒ Hat Shape

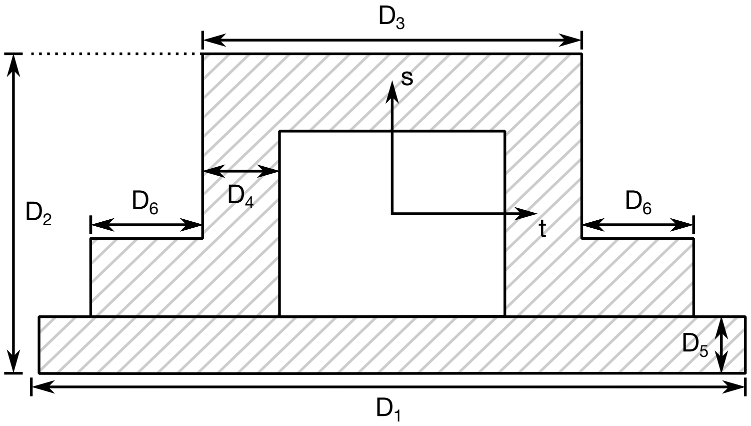


Figure 36-22. SECTION_22 ⇒ Hat Shape 2

*SECTION_BEAM_AISC

Purpose: Define cross-sectional properties for beams and trusses using section labels from the AISC Steel Construction Manual, 2005, 13th Edition, as published in the AISC Shapes Database V13.1.1

Card Sets. For each BEAM_AISC section include one pair of cards 1 and 2. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	SECID	LABEL						
Type	I	A70						

Integrated Beam Card (types 1 and 11). Card 2 for ELFORM equal to 1 or 11.

Card 2	1	2	3	4	5	6	7	8
Variable	ELFORM	SHRF	NSM	LFAC	NSLOC	NTLOC	K	
Type	I	F	F	F	F	F	I	

Resultant Beam Card (types 2 and 12). Card 2 for ELFORM equal to 2 or 12.

Card 2	1	2	3	4	5	6	7	8
Variable	ELFORM	SHRF	NSM	LFAC				
Type	I	F	F	F				

Truss Beam Card (type 3). Card 2 for ELFORM equal to 3.

Card 2	1	2	3	4	5	6	7	8
Variable	ELFORM	LFAC	RAMPT	STRESS				
Type	I	F	F	F				

Integrated Beam Card (types 4 and 5). Card 2 for ELFORM equal to 4 or 5.

Card 2	1	2	3	4	5	6	7	8
Variable	ELFORM	SHRF	NSM	LFAC	K			
Type	I	F	F	K				

VARIABLE	DESCRIPTION
SECID	Section ID. SECID is referenced on the *PART card. A unique number or label must be specified.
LABEL	AISC section label
ELFORM	Element formulation (see *SECTION_BEAM). Only types 1–5,11,12 are allowed
SHRF	Shear factor (see *SECTION_BEAM)
NSM	Non-structural mass per unit length
LFAC	GT.0.0: Length scale factor to convert dimensions from standard units LT.0.0: Use predefined length factor for specific model units EQ.-1.0: ft EQ.-2.0: m EQ.-3.0: in EQ.-4.0: mm EQ.-5.0: cm
NSLOC	Location of reference surface (see *SECTION_BEAM)
NTLOC	Location of reference surface (see *SECTION_BEAM)
K	Integration refinement parameter (see *INTEGRATION_BEAM)
RAMPT	Optional ramp-up time (see *SECTION_BEAM)
STRESS	Optional initial stress (see *SECTION_BEAM)

Remarks:

This keyword uses the dimensions of the standard AISC beams sections — as defined by the section label — to define *SECTION_BEAM and *INTEGRATION_BEAM cards with the appropriate parameters.

The AISC section label may be specified either as the shape designation as seen in the AISC Steel Construction Manual, 2005, or the designation according to the AISC Naming Convention for Structural Steel Products for Use in Electronic Data Interchange (EDI), 2001. As per the EDI convention, the section labels are to be case-sensitive and space sensitive, i.e. "W36X150" is acceptable but "W36 x 150" is not. Labels can be specified in terms of either the U.S. Customary units (in) or metric units (mm), which will determine the length units for the section dimensions. The parameter LFAC may be used as a multiplier to convert the dimensions to other lengths units. For user convenience, predefined conversion factors are provided for specific choices of the model length unit.

AISC requires the following legal notice specifying that LS-DYNA users may not extract AISC shapes data in any manner from LS-DYNA to create commercial software. Users are certainly allowed to create LS-DYNA input models for commercial purposes using this card.

AISC FLOW-DOWN LICENSE TERMS (TERMS OF USE)

This application contains software from the American Institute of Steel Construction, Inc. of Chicago, Illinois d/b/a AISC ("AISC"). The software from AISC (the "AISC Shapes Database") enables this application to provide dimensions and properties of structural steel shapes and to perform other functions. You may use AISC Data only by means of the intended End User functions of this application software. You agree that you will use AISC Data for non-commercial use only, meaning that it will not be used to develop or create revenue-producing software. You agree not to assign, copy, transfer or transmit the AISC Shapes Database or any AISC Data to any third party.
YOU AGREE NOT TO USE OR EXPLOIT AISC DATA OR THE AISC SHAPES DATABASE EXCEPT AS EXPRESSLY PERMITTED HEREIN.

*SECTION_DISCRETE

Purpose: Defined spring and damper elements for translation and rotation. These definitions must correspond with the material type selection for the elements, i.e., *MAT_SPRING_... and *MAT_DAMPER_...

Card Sets. For each DISCRETE section include a pair of cards 1 and 2. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	SECID	DRO	KD	V0	CL	FD		
Type	I/A	I	F	F	F	F		

Card 2	1	2	3	4	5	6	7	8
Variable	CDL	TDL						
Type	F	F						

VARIABLE	DESCRIPTION
SECID	Section ID. SECID is referenced on the *PART card. A unique number or label must be specified.
DRO	Displacement/Rotation Option: EQ.0: the material describes a translational spring/damper, EQ.1: the material describes a torsional spring/damper.
KD	Dynamic magnification factor. See Remarks 1 and 2 below.
V0	Test velocity
CL	Clearance. See Remark 3 below.
FD	Failure deflection (twist for DRO = 1). Negative for compression, positive for tension.
CDL	Deflection (twist for DRO = 1) limit in compression. See Remark 4 below.

VARIABLE	DESCRIPTION
TDL	Deflection (twist for DRO = 1) limit in tension. See Remark 4 below.

Remarks:

1. The constants from KD to TDL are optional and do not need to be defined.

2. If k_d is nonzero, the forces computed from the spring elements are assumed to be the static values and are scaled by an amplification factor to obtain the dynamic value:

$$F_{\text{dynamic}} = \left(1. + k_d \frac{V}{V_0}\right) F_{\text{static}}$$

   where

   V = absolute value of the relative velocity between the nodes.

   V_0 = dynamic test velocity.

   For example, if it is known that a component shows a dynamic crush force at 15m/s equal to 2.5 times the static crush force, use k_d = 1.5 and V_0 = 15.

3. Here, “clearance” defines a compressive displacement which the spring sustains before beginning the force-displacement relation given by the load curve defined in the material selection. If a non-zero clearance is defined, the spring is compressive only.

4. The deflection limit in compression and tension is restricted in its application to no more than one spring per node subject to this limit, and to deformable bodies only. For example in the former case, if three springs are in series, either the center spring or the two end springs may be subject to a limit, but not all three. When the limiting deflection is reached, momentum conservation calculations are performed and a common acceleration is computed in the appropriate direction. An error termination will occur if a rigid body node is used in a spring definition where deflection is limited.

   Constrained boundary conditions on the *NODE cards and the BOUNDARY_-SPC cards must not be used for nodes of springs with deflection limits.

5. Discrete elements can be included in implicit applications.

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$  *SECTION_DISCRETE
$
```

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Note: These examples are in kg, mm, ms, kN units.
$
$  A translational spring (dro = 0) is defined to have a failure deflection
$  of 25.4 mm (fd = 25.4).  The spring has no dynamic effects or
$  deflection limits, thus, those parameters are not set.
$
*SECTION_DISCRETE
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      sid       dro        kd        v0        cl        fd
       104         0                                    25.4
$
$      cdl       tdl

$
$
$  Define a translational spring that is known to have a dynamic crush force
$  equal to 2.5 times the static force at a 15 mm/ms deflection rate.
$  Additionally, the spring is known to be physically constrained to deflect
$  a maximum of 12.5 mm in both tension and compression.
$
*SECTION_DISCRETE
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      sid       dro        kd        v0        cl        fd
       107         0       1.5      15.0
$
$      cdl       tdl
      12.5      12.5
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

*SECTION_POINT_SOURCE

Purpose: This command provides the inlet boundary condition for single gas in flow (inflation potential) via a set of point source(s). It also provides the inflator orifice geometry information. It requires 3 curves defining the inlet condition for the inflator gas coming into the tank or an airbag as input ($\bar{T}_{\text{gas corrected}}(t)$, $v_r(t)$, and $\text{vel}(t)$). Please see also the *ALE_TANK_TEST card for additional information.

Card 1	1	2	3	4	5	6	7	8
Variable	SECID	LCIDT	LCIDVR	LCIDVEL	NIDLC1	NIDLC2	NIDLC3	
Type	I/A	I	I	I	I	I	I	
Default	0	0	0	0	0	0	0	

Source Node Cards. Include one card for each source node. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	NODEID	VECID	ORIFA					
Type	I	I	F					
Default	0	0	0.0					

VARIABLE	DESCRIPTION
SECID	Section ID. A unique number or label must be specified.
LCIDT	Temperature load curve ID
LCIDVR	Relative volume load curve ID
LCIDVEL	Inlet flow velocity load curve ID
NIDLC1	The 1st node ID defining a local coordinate (See Remark 2).
NIDLC2	The 2nd node ID defining a local coordinate (See Remark 2).

VARIABLE	DESCRIPTION
NIDLC3	The 3rd node ID defining a local coordinate (See Remark 2).
NODEID	The node ID(s) defining the point source(s).
VECID	The vector ID defining the direction of flow at each point source.
ORIFA	The orifice area at each point source.

Remarks:

1. In an airbag inflator tank test, the tank pressure data is measured. This pressure is used to derive $\dot{m}(t)$ and the estimated $\bar{T}_{gas}(t)$, usually via a lumped-parameter method, a system of conservation equations and EOS. Subsequently $\dot{m}(t)$ and $\bar{T}_{gas}(t)$ (stagnation temperature) are used as input to obtain $\bar{T}_{gas\_corrected}(t)$ (static temperature), $v_r(t)$, and $vel(t)$. These 3 curves are then used to describe inflator gas inlet condition (see *ALE_TANK_TEST for more information).

2. In a car crash model, the inflator housing may get displaced during the impact. The 3 node IDs defines the local reference coordinate system to which the point sources are attached. These 3 reference nodes may be located on a rigid body which can translate and rotate as the inflator moves during the impact. This allows for the point sources to move in time. These reference nodes may be used as the point sources themselves.

3. If the *ALE_TANK_TEST card is present, please see the Remarks under that card.

Example:

Consider a tank test model which consists of the inflator gas (PID 1) and the air inside the tank (PID 2). The 3 load curves define the thermodynamic and kinetic condition of the incoming gas. The nodes define the center of the orifice, and the vector the direction of flow at each orifice.

```
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|...8
*PART
inflator gas
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
         1         1         1         0         0         0         0         0
*SECTION_POINT_SOURCE
$    SECID     LCIDT  LCIDVOLR   LCIDVEL NIDLCOOR1 NIDLCOOR2 NIDLCOOR3
         1         3         4         5         0         0         0
$   NODEID    VECTID      AREA
     24485         3    15.066
       ...
     24557         3    15.066
*PART
air inside the tank
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
         2         2         2         0         0         0         0         0
*SECTION_SOLID
$    SECID    ELFORM       AET
         2        11         0
*ALE_MULTI-MATERIAL_GROUP
$      SID   SIDTYPE
         1         1
         2         1
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|...8
```

*SECTION_POINT_SOURCE_MIXTURE

Purpose: This command provides (a) an element formulation for a solid ALE part of the type similar to ELFORM = 11 of *SECTION_SOLID, and (b) the inlet gas injection boundary condition for multiple-gas mixture in-flow via a set of point source(s). It also provides the inflator orifice geometry information. This must be used in combination with the *MAT_GAS_MIXTURE and/or *INITIAL_GAS_MIXTURE card (see Remark 1).

Card 1	1	2	3	4	5	6	7	8
Variable	SECID	LCIDT	Not Used	LCIDVEL	NIDLC1	NIDLC2	NIDLC3	IDIR
Type	I/A	I		I	I	I	I	I
Default	none	none		none	none	none	none	0

Card 2	1	2	3	4	5	6	7	8
Variable	LCMD1	LCMD2	LCMD3	LCMD4	LCMD5	LCMD6	LCMD7	LCMD8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

Source Node Cards. Include one card for each source node. This input ends at the next keyword (“*”) card.

Card 3	1	2	3	4	5	6	7	8
Variable	NODEID	VECID	ORIFA					
Type	I	I	F					
Default	none	none	0.0					

VARIABLE	DESCRIPTION
SECID	Section ID. A unique number or label must be specified.

VARIABLE	DESCRIPTION
LCIDT	Inflator gas mixture average stagnation temperature load curve ID (all gases of the mixture are assumed to have the same average temperature).
LCIDVEL	User-defined inflator gas mixture average velocity load curve ID. If LCIDVEL = 0 or blank, LSDYNA will estimate the inlet gas velocity.
NIDLC001	The 1^{st} node ID defining a local coordinate (see Remark 2).
NIDLC002	The 2^{nd} node ID defining a local coordinate (see Remark 2).
NIDLC003	The 3^{rd} node ID defining a local coordinate (see Remark 2).
IDIR	A flag for constraining the nodal velocity of the nodes of the ALE element containing a point source. If IDIR = 0 (default), then the ALE nodes behind the point source (relative position of nodes based on the vector direction of flow of point source) will have zero velocity. If IDIR = 1, then all ALE nodes will have velocity distributed based on energy conservation. The latter option seems to be more robust in airbag modeling (see Remark 6).
LCMD1	The mass flow rate load curve ID of the1^{st} gas in the mixture.
LCMDn	The mass flow rate load curve ID of the n^{th} gas in the mixture.
LCMD8	The mass flow rate load curve ID of the 8^{th} gas in the mixture.
NODEID	The node ID(s) defining the point sources (see Remark 6).
VECID	The vector ID defining the direction of flow at each point source.
ORIFA	The orifice area at each point source.

Remarks:

1. This command is used to define a part that acts as the ideal gas mixture injection source. The associated ALE material (gas mixture) may not be present at time zero, but can be introduced (injected) into an existing ALE domain. For airbag application, the input from control volume analysis, inlet mass flow rate, $\dot{m}(t)$, and, inlet stagnation gas temperature, $\overline{T}_{gas}(t)$ may be used as direct input for ALE analysis. If available, the user may input a load curve for the gas mixture average inlet velocity. If not, LS-DYNA will estimate the inlet gas velocity.

2. The gas mixture is assumed to have a uniform temperature ($\bar{T} \approx T_i$) and inlet velocity. However, the species in the mixture may each have a different inlet mass flow rate.

3. A brief review of the concept used is presented. The total energy (e_T) is the sum of internal (e_i) and kinetic $\left(\frac{V^2}{2}\right)$ energies, (per unit mass).

$$e_T = e_i + \frac{V^2}{2}$$

$$C_V T_{stag} = C_V T + \frac{V^2}{2}$$

$$T_{stag} = T + \frac{V^2}{2C_V}$$

The distinction between stagnation and static temperatures is shown above. C_V is the constant-volume heat capacity. The gas mixture average internal energy per unit mass in terms of mixture species contribution is

$$e_i = \bar{C}_V \bar{T} = \sum_i \left(\frac{\rho_i}{\rho_{mixture}}\right) C_{V_i} T_i = \left[\sum_i \left(\frac{\rho_i}{\rho_{mixture}}\right) C_{V_i}\right] \bar{T}$$

$$\bar{C}_V = \left[\sum_i \left(\frac{\rho_i}{\rho_{mixture}}\right) C_{V_i}\right]$$

Since we approximate $\bar{T} \approx T_i$, then gas mixture average static temperature is related to the mixture average internal energy per unit mass as following

$$\bar{T} = \frac{e_i}{\left[\sum_i \left(\frac{\rho_i}{\rho_{mixture}}\right) C_{V_i}\right]}$$

Note that the "i" subscript under "e" denotes "internal" energy, while the other "i" subscripts denote the "ith" species in the gas mixture. The total mixture pressure is the sum of the partial pressures of the individual species.

$$\bar{p} = \sum_i p_i$$

The ideal gas EOS applies to each individual species (by default)

$$P_i = \rho_i (C_{P_i} - C_{V_i}) T_i$$

4. Generally, it is not possible to conserve both momentum and kinetic (KE) at the same time. Typically, internal energy (IE) is conserved and KE may not be. This may result in some KE loss (hence, total energy loss). For many analyses this is tolerable, but for airbag application, this may lead to the reduction of the inflating potential of the inflator gas.

   In *MAT_GAS_MIXTURE computation, any kinetic energy not accounted-for during advection is stored in the internal energy. Therefore, there is no kinetic energy loss, and the total energy of the element is conserved over the advection step. This is a simple, ad hoc approach that is not rigorously derived for the whole system based on first principles. Therefore it is not guaranteed to apply universally to all scenarios. It is the user's responsibility to validate the model with data.

5. Since ideal gas is assumed, there is no need to define the EOS for the gases in the mixture.

6. In general, it is best to locate a point source near the center of an ALE element. Associated with each point source is an area and a vector indicating flow direction. Each point source should occupy 1 ALE element by itself, and there should be at least 2 empty ALE elements between any 2 point sources. A point source should be located at least 3 elements away from the free surface of an ALE mesh for stability.

Example 1:

Consider a tank test model without coupling which consists of:

-a background mesh with air (PID 1 = gas 1) initially inside that mesh (tank space), and
-the inflator gas mixture (PID 2 consisting of inflator gases 2, 3, and 4).

The mixture is represented by one AMMGID and the air by another AMMGID.

The tank internal space is simply modeled with an Eulerian mesh of the same volume. The Tank itself is not modeled thus no coupling is required. The inflator gases fill up this space mixing with the air initially inside the tank.

The background air (gas 1) is included in the gas mixture definition in this case because that air will participate in the mixing process. Only include in the mixture those gases that actually undergo mixing (gases 1, 2, 3 and 4). Note that for an airbag model, the "outside" air should not be included in the mixture (it should be defined independently) since it does not participate in the mixing inside the airbag. This is shown in the next example.

The nodes define the center of the orifices, and the vectors define the directions of flow at these orifices.

```
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
*PART
Tank background mesh, initially filled with air, allows gas mixture to flow in.
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
         1         1         1         0         0         0         0         0
*SECTION_SOLID
$    SECID    ELFORM       AET
         1        11         0
$ The next card defines the properties of the gas species in the mixture.
*MAT_GAS_MIXTURE
$      MID
         1
$      Cv1       Cv2       Cv3       Cv4       Cv5       Cv6       Cv7       Cv8
    654.47    482.00   2038.30    774.64       0.0       0.0       0.0       0.0
$      Cp1       Cp2       Cp3       Cp4       Cp5       Cp6       Cp7       Cp8
    941.32    666.67   2500.00   1071.40       0.0       0.0       0.0       0.0

$ The next card specifies that gas 1 (background air) occupies PID 1 at time 0.
*INTIAL_GAS_MIXTURE
$      SID     STYPE    AMMGID     TEMP0
         1         1         1    293.00
$     RHO1      RHO2      RHO3      RHO4      RHO5      RHO6      RHO7      RHO8
   1.20E-9       0.0       0.0       0.0       0.0       0.0       0.0       0.0
*PART
The gas mixture (inlet) definition (no initial mesh required for this PID)
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
         2         2         1         0         0         0         0         0
*SECTION_POINT_SOURCE_MIXTURE
$    SECID     LCIDT   NOTUSED   LCIDVEL NIDLCOOR1 NIDLCOOR2 NIDLCOOR3      IDIR
         2         1         0         5         0         0         0         0
$  LCMDOT1   LCMDOT2   LCMDOT3   LCMDOT4   LCMDOT5   LCMDOT6   LCMDOT7   LCMDOT8
         0         2         3         4         0         0         0         0
$   NODEID    VECTID      AREA
     24485         1      25.0
       ...
     24557         1      25.0
*ALE_MULTI-MATERIAL_GROUP
$      SID   SIDTYPE
         1         1
         2         1
*DEFINE_VECTOR
$   VECTID     XTAIL     YTAIL     ZTAIL     XHEAD     YHEAD     ZHEAD
         1       0.0       0.0       0.0       0.0       1.0       0.0
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
```

Example 2:

Consider an airbag inflation model which consists of:

-a background Eulerian mesh for air initially outside the airbag (PID 1)
-the inflator gas mixture (PID 2 consisting of inflator gases 1, 2, and 3).

The mixture is represented by one AMMGID and the air by another AMMGID.

The background air (PID 1) is NOT included in the gas mixture definition in this case because that air will NOT participate in the mixing process. Only include in the mixture those gases that actually undergo mixing (gases 1, 2, and 3). Gases 1, 2, and 3 in this example correspond to gases 2, 3, and 4 in example 1. Compare the air properties in PID 1 here to that of example 1. Note that the *INITIAL_GAS_MIXTURE card is not required to initialize the background mesh in this case.

```
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
*PART
Tank background mesh, initially filled with air, allows gas mixture to flow in.
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
         1         1         1         0         0         0         0         0
*SECTION_SOLID
$    SECID    ELFORM       AET
         1        11         0
*MAT_NULL
$      MID       RHO      PCUT        MU     TEROD     CEROD        YM        PR
         1   1.20E-9   -1.0E-6       0.0       0.0       0.0       0.0       0.0
*EOS_IDEAL_GAS
$    EOSID       CV0       CP0     COEF1     COEF2        T0   RELVOL0
         1    654.47    941.32       0.0       0.0    293.00       1.0
$ The next card defines the properties of the gas species in the mixture.
*PART
The gas mixture (inlet) definition (no initial mesh required for this PID)
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
         2         2         2         0         0         0         0         0
*SECTION_POINT_SOURCE_MIXTURE
$    SECID     LCIDT   NOTUSED   LCIDVEL NIDLCOOR1 NIDLCOOR2 NIDLCOOR3      IDIR
         2         1         0         5         0         0         0         0
$  LCMDOT1   LCMDOT2   LCMDOT3   LCMDOT4   LCMDOT5   LCMDOT6   LCMDOT7   LCMDOT8
         2         3         4         0         0         0         0         0
$   NODEID    VECTID      AREA
     24485         1      25.0
       ...
     24557         1      25.0
*MAT_GAS_MIXTURE
$      MID
         2
$      Cv1       Cv2       Cv3       Cv4       Cv5       Cv6       Cv7       Cv8
    482.00   2038.30    774.64       0.0       0.0       0.0       0.0
$      Cp1       Cp2       Cp3       Cp4       Cp5       Cp6       Cp7       Cp8
    666.67   2500.00   1071.40       0.0       0.0       0.0       0.0
$ The next card specifies that gas 1 (background air) occupies PID 1 at time 0.
*ALE_MULTI-MATERIAL_GROUP
$      SID   SIDTYPE
         1         1
         2         1
*DEFINE_VECTOR
$   VECTID     XTAIL     YTAIL     ZTAIL     XHEAD     YHEAD     ZHEAD
         1       0.0       0.0       0.0       0.0       1.0       0.0
$...|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8
```

*SECTION_SEATBELT

Purpose: Define section properties for the seat belt elements. This card is required for the *PART Section. Currently, only the ID is required.

Seatbelt Section Cards. Include one card for each SEATBELT section. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	SECID	AREA	THICK					
Type	I/A	F	F					
Default	none	0.01	none					

VARIABLE	DESCRIPTION
SECID	Section ID. A unique number or label must be specified.
AREA	Optional area of cross-section used in the calculation of contact stiffness, which is proportional to the cross-section area.
THICK	Optional contact thickness which can be overwritten by a nonzero SST defined in *CONTACT. If not defined, a value proportional to element length is used as the contact thickness.

Remarks:

Seatbelt elements are implemented for both explicit and implicit calculations.

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$   *SECTION_SEATBELT
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Define a seat belt section that is referenced by part 10.  Nothing
$  more than the sid is required.
$
*SECTION_SEATBELT
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      sid
       111
$
$
*PART
Seatbelt material
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      pid       sid       mid     eosid      hgid    adpopt
        10       111       220
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

SECTION_SHELL_*{OPTION}

Available options include:

<BLANK>

EFG

THERMAL

XFEM

Purpose: Define section properties for shell elements.

Card Sets. For each shell section, of a type matching the keyword's options, include one set data cards. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	SECID	ELFORM	SHRF	NIP	PROPT	QR / IRID	ICOMP	SETYP
Type	I/A	I	F	F	F	F	I	I
Default	none		1.0	2	0.0	0.0	0	1
Remarks		1						

Card 2	1	2	3	4	5	6	7	8
Variable	T1	T2	T3	T4	NLOC	MAREA	IDOF	EDGSET
Type	F	F	F	F	F	F	F	I
Default	0.0	T1	T1	T1	0.0	0.0	0.0	
Remarks							6	7

Angle Cards. Additional cards for ICOMP = 1. Include the minimum number of cards necessary to input NIP values: 8 values per card ⇒ number of cards = ceil(NIP/8) where ceil(x) = the smallest integer greater than x.

Card 3	1	2	3	4	5	6	7	8
Variable	B1	B2	B3	B4	B5	B6	B7	B8
Type	F	F	F	F	F	F	F	F

EFG Card. Additional card for EFG keyword option. See *CONTROL_EFG.

Card 4	1	2	3	4	5	6	7	8
Variable	DX	DY	ISPLINE	IDILA	IEBT	IDIM		
Type	F	F	I	I	I	I		
Default	1.1	1.1	0	0	-1 or 1	2 or 1		

Thermal Card. Additional Card for THERMAL keyword option.

Card 4	1	2	3	4	5	6	7	8
Variable	ITHELFM							
Type	I							
Default	0							

XFEM Card. Additional card for XFEM keyword option. See Remark 8.

Card 4	1	2	3	4	5	6	7	8
Variable	CMID	BASELM	DOMINT	FAILCR	PROPCR	FS	LS/FS1	NC/CL
Type	I	I	I	I	I	F		
Default			0	1		0.0		

User Defined Element Card. Additional card for ELFORM = 101,102,103,104 or 105. See Appendix C

Card 5	1	2	3	4	5	6	7	8
Variable	NIPP	NXDOF	IUNF	IHGF	ITAJ	LMC	NHSV	ILOC
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

User Defined Element Integration Point Cards. Additional cards for ELFORM = 101, 102, 103, 104 or 105. Define NIPP cards according to the following format. See Appendix C.

Card 6	1	2	3	4	5	6	7	8
Variable	XI	ETA	WGT					
Type	F	F	F					
Default	none	none	none					

User Defined Element Property Cards. Include the minimum number of cards necessary to input LMC values: 8 values per card ⇒ number of cards $=$ ceil$(\mathrm{LMC}/8)$ where ceil(x) $=$ the smallest integer greater than x. See Appendix C.

Card 7	1	2	3	4	5	6	7	8
Variable	P1	P2	P3	P4	P5	P6	P7	P8
Type	F	F	F	F	F	F	F	F
Default	0	0	0	0	0	0	0	0

VARIABLE	DESCRIPTION
SECID	Section ID. SECID is referenced on the *PART card. A unique

VARIABLE **DESCRIPTION**

number or label must be specified.

ELFORM Element formulation options, see Remarks 1 and 2 below:

EQ.1: Hughes-Liu,

EQ.2: Belytschko-Tsay,

EQ.3: BCIZ triangular shell,

EQ.4: C0 triangular shell,

EQ.5: Belytschko-Tsay membrane,

EQ.6: S/R Hughes-Liu,

EQ.7: S/R co-rotational Hughes-Liu,

EQ.8: Belytschko-Leviathan shell,

EQ.9: Fully integrated Belytschko-Tsay membrane,

EQ.10: Belytschko-Wong-Chiang,

EQ.11: Fast (co-rotational) Hughes-Liu,

EQ.12: Plane stress (x-y plane),

EQ.13: Plane strain (x-y plane),

EQ.14: Axisymmetric solid (x-y plane, y-axis of symmetry) - area weighted (see Remark 10),

EQ.15: Axisymmetric solid (x-y plane, y-axis of symmetry) - volume weighted,

EQ.16: Fully integrated shell element (very fast),

EQ.-16: Fully integrated shell element modified for higher accuracy, see Remark 0.

EQ.17: Fully integrated DKT, triangular shell element,

EQ.18: Fully integrated linear DK quadrilateral/triangular shell

EQ.20: Fully integrated linear assumed strain C0 shell (See Remark 2).

EQ.21: Fully integrated linear assumed strain C0 shell (5 DOF).

EQ.22: Linear shear panel element. 3 DOF per node. (See Remark 3).

EQ.23: 8-node quadratic quadrilateral shell

VARIABLE **DESCRIPTION**

EQ.24: 6-node quadratic triangular shell

EQ.25: Belytschko-Tsay shell with thickness stretch.

EQ.26: Fully integrated shell with thickness stretch.

EQ.27: C0 triangular shell with thickness stretch.

EQ.29: Cohesive shell element for edge-to-edge connection of shells. See Remark 0.

EQ.-29: Cohesive shell element for edge-to-edge connection of shells (more suitable for pure shear). See Remark 0.

EQ.41: Mesh-free (EFG) shell local approach. (more suitable for crashworthiness analysis)

EQ.42: Mesh-free (EFG) shell global approach. (more suitable for metal forming analysis)

EQ.43: Mesh-free (EFG) plane strain formulation (x-y plane).

EQ.44: Mesh-free (EFG) axisymmetric solid formulation (x-y plane, y-axis of symmetry).

EQ.46: Cohesive element for two-dimensional plane strain, plane stress, and area-weighted axisymmetric problems (type 14 shells).

EQ.47: Cohesive element for two-dimensional volume-weighted axisymmetric problems (use with type 15 shells).

EQ.52: Plane strain (x-y plane) XFEM, base element type 13.

EQ.54: Shell XFEM, base element type defined by BASELM (default 16).

EQ.55: 8-node singular plane strain (x-y plane) finite element, see Remark 11.

EQ.98: Interpolation shell

EQ.99: Simplified linear element for time-domain vibration studies. See Remark 4 below.

EQ.101: User defined shell

EQ.102: User defined shell

EQ.103: User defined shell

EQ.104: User defined shell

VARIABLE	DESCRIPTION
	EQ.105: User defined shell
	EQ.201: Isogeometric shells with NURBS. See *ELEMENT_SHELL_NURBS_PATCH
	GE.1000: Generalized shell element formulation (user defined). See *DEFINE_ELEMENT_GENERALIZED_SHELL
	The type 18 element is only for linear static and normal modes. It can also be used for linear springback in sheet metal stamping. For implicit modal computations element type 14 must be switched to type 15. Note that the 2D and 3D element types must not be mixed, and different types of 2D elements must not be used together. For example, 2D axisymmetric calculations can use either element types 14 or 15 but these element types must not be mixed together. Likewise, the plane strain element type must not be used with either the plane stress element or the axisymmetric element types. In 3D, the different shell elements types, i.e., 1-11 and 16, can be freely mixed together.
SHRF	Shear correction factor which scales the transverse shear stress. The shell formulations in LS-DYNA, with the exception of the BCIZ and DK elements, are based on a first order shear deformation theory that yields constant transverse shear strains which violates the condition of zero traction on the top and bottom surfaces of the shell. The shear correction factor is attempt to compensate for this error. A suggested value is 5/6 for isotropic materials. This value is incorrect for sandwich or laminated shells; consequently, laminated/sandwich shell theory is now an option in some of the constitutive models, e.g., material types 22, 54, and 55.
NIP	Number of through thickness integration points. Either Gauss (default) or Lobatto integration can be used. The flag for Lobatto integration can be set on the control card, *CONTROL_SHELL. The location of the Gauss and Lobatto integration points are tabulated below. EQ.0.0: set to 2 integration points for shell elements. EQ.1.0: 1 point (no bending) EQ.2.0: 2 point EQ.3.0: 3 point

VARIABLE **DESCRIPTION**

EQ.4.0: 4 point

EQ.5.0: 5 point

EQ.6.0: 6 point

EQ.7.0: 7 point

EQ.8.0: 8 point

EQ.9.0: 9 point

EQ.10.: 10 point

GT.10.: trapezoidal or user defined rule

Through thickness integration for the two-dimensional elements (options 12-15 above) is not meaningful; consequently, the default is equal to 1 integration point. Fully integrated two-dimensional elements are available for options 13 and 15 (but not 12 and 14) by setting NIP equal to a value of 4 corresponding to a 2 by 2 Gaussian quadrature. If NIP is 0 or 1 and the *MAT_SIMPLIFIED_JOHNSON_COOK model is used, then a resultant plasticity formulation is activated. NIP is always set to 1 if a constitutive model based on resultants is used.

PROPT

Printout option (***NOT ACTIVE***):

EQ.1.0: average resultants and fiber lengths,

EQ.2.0: resultants at plan points and fiber lengths,

EQ.3.0: resultants, stresses at all points, fiber lengths.

QR/IRID

Quadrature rule or Integration rule ID, see *INTEGRATION_-SHELL:

LT.0.0: absolute value is specified rule number,

EQ.0.0: Gauss/Lobatto (up to 10 points are permitted),

EQ.1.0: trapezoidal, not recommend for accuracy reasons.

ICOMP

Flag for orthotropic/anisotropic layered composite material model. This option applies to material types 21, 22, 23, 33, 33_96, 34, 36, 40, 41-50, 54, 55, 58, 59, 103, 103_P, 104, 108, 116, 122, 133, 135, 135_PLC, 136, 157, 158, 190, 219, 226, 233, 234, 235, 242, and 243. For these material types, see *PART_COMPOSITE as an alternative to *SECTION_SHELL. **Note:** Please refer to ***Remark 5*** under *MAT_034 for additional information specific to fiber directions for fabrics.

VARIABLE	DESCRIPTION
	EQ.1: a material angle in degrees is defined for each through thickness integration point. Thus, each layer has one integration point.
SETYP	Not used (obsolete).
T1	Shell thickness at node n_1, unless the thickness is defined on the *ELEMENT_SHELL_*OPTION* card.
T2	Shell thickness at node n_2, see comment for T1 above.
T3	Shell thickness at node n_3, see comment for T1 above.
T4	Shell thickness at node n_4, see comment for T1 above.
NLOC	Location of reference surface (shell mid-thickess) for three dimensional shell elements. If nonzero, the offset distance from the plane of the nodal points to the reference surface of the shell in the direction of the shell normal vector is a value, offset = $-0.50 \times \text{NLOC} \times$ (average shell thickness). Except for Mortar contacts, this offset is not considered in the contact subroutines unless CNTCO is set to 1 in *CONTROL_-SHELL. Alternatively, the offset can be specified by using the OFFSET option in the *ELEMENT_SHELL input section. For Mortar contacts, NLOC or OFFSET determines the location of the contact surface regardless the value of CNTCO. EQ.1.0: nodes are located at top surface of shell, EQ.0.0: nodes are located at mid-thickness of shell (default), EQ.-1.0: nodes are located at bottom surface of shell.
MAREA	Non-structural mass per unit area. This is additional mass which comes from materials such as carpeting. This mass is not directly included in the time step calculation. Another and often more convenient alternative for defining distributed mass is by the option: *ELEMENT_MASS_PART, which allows additional non-structural mass to be distributed by an area weighted distribution to all nodes of a given part ID).
IDOF	Treatment of through thickess strain. LT.0: Same as IDOF.EQ.3 but the contact pressure is averaged over a time –IDOF in order to reduce noise and thus

VARIABLE **DESCRIPTION**

improve stability.

EQ.1: The thickness field is continuous across the element edges for metalforming applications. This option applies to element types 25 and 26.

EQ.2: The thickness field is discontinuous across the element edges. This is necessary for crashworthiness simulations due to shell intersections, sharp included angles, and non-smooth deformations. This option applies to element types 25, 26 and 27 and is mandatory for element 27. This is the default for these element types.

EQ.3: The thickness strain is governed by the contact stress, meaning that the strain is adjusted for the through thickness stress to equilibrate the contact pressure. This option applies to element types 2, 4, and ±16 .

EDGSET

Edge node set required for shell type seatbelts. Input an ordered set of nodes along one of the transverse edges of a seatbelt. If there is no retractor associated with a belt, the node set can be on either edge. If the retractor exists, the edge must be on the retractor side and input in the same sequence of retractor node set. Therefore, another restriction on the seatbelt usage is that each belt has its own section definition and, therefore, a unique part ID. See Figure 17-16 in the section *ELEMENT_SEATBELT for additional clarification.

B1

β_1, material angle at first integration point

B2

β_2, material angle at second integration point

B3

β_3, material angle at third integration point

⋮

⋮

BNIP

β_{nip}, material angle at NIPthintegration point.

DX, DY

Normalized dilation parameters of the kernel function in X and Y directions. The normalized dilation parameters of the kernel function are introduced to provide the smoothness and compact support properties on the construction of the mesh-free shape functions. Values between 1.0 and 2.0 are recommended. Values smaller than 1.0 are not allowed. Larger values will increase the computation time and will sometimes result in a divergence problem.

VARIABLE **DESCRIPTION**

ISPLINE

Replace the choice for the EFG kernel functions definition in *CONTROL_EFG. This allows users to define different ISPLINE in different sections.

IDILA

Replace the choice for the normalized dilation parameter definition in *CONTROL_EFG. This allows users to define different IDILA in different sections.

IEBT

Essential boundary condition treatment

EQ.1: Full transformation (default for ELFORM = 42)

EQ.-1: Without full transformation (default for ELFORM = 41)

EQ.3: Coupled FEM/EFG

EQ.7: Maximum entropy approximation

IDIM

For mesh-free shell local approach (ELFORM = 41)

EQ.1: First-kind local boundary condition method

EQ.2: Gauss integration (default)

For mesh-free shell global approach (ELFORM = 42)

EQ.1: First-kind local boundary condition method (default)

EQ.2: Second-kind local boundary condition method

ITHELFM

Thermal shell formulation

EQ.0: Default is governed by THSHEL on *CONTROL_SHELL

EQ.1: Thick thermal shell

EQ.2: Thin thermal shell

CMID

Cohesive material ID (only *MAT_COHESIVE_TH is available)

BASELM

Base element type for XFEM (type 13 for 2D, types 2 and 16 for shell)

DOMINT

Option for domain integration in XFEM:

EQ.0: Phantom element integration

EQ.1: Subdomain integration with triangular local boundary integration (available in 2D only)

FAILCR

Option for different failure criteria:

VARIABLE **DESCRIPTION**

EQ.1: Maximum tensile stress

EQ.2: Maximum shear stress

EQ.-1: Effective plastic strain (see remark 14)

EQ.-2: Crack length dependent EPS (see remark 14)

EQ.-n: n > 10, (n-10) points to HSVS for mat282/283 (see remark 14)

PROPCR Not used

FS Failure strain/Failure critical value

LS Length scale for strain regularization. > 0 activates regularization, available for FAILCR = -1 and –n for mat282/283

NC Number of cracks allowed in the part

When FAILCR = -2, following three parameters represent (see Remark 14):

FS Initial failure plastic strain

FS1 Final failure plastic strain

CL Crack length failure strain reaches FS1

NIPP Number of in-plane integration points for user-defined shell (0 if resultant/discrete element)

NXDOF Number of extra degrees of freedom per node for user-defined shell

IUNF Flag for using nodal fiber vectors in user-defined shell:

EQ.0: Nodal fiber vectors are not used.

EQ.1: Nodal fiber vectors are used.

IHGF Flag for using hourglass stabilization (NIPP.GT.0)

EQ.0: Hourglass stabilization is not used

EQ.1: LS-DYNA hourglass stabilization is used

VARIABLE	DESCRIPTION
	EQ.2: User-defined hourglass stabilization is used EQ.3: Same as 2, but the resultant material tangent moduli are passed
ITAJ	Flag for setting up finite element matrices (NIPP.GT.0) EQ.0: Set up matrices wrt isoparametric domain EQ.1: Set up matrices wrt physical domain
LMC	Number of property parameters
NHSV	Number of history variables
ILOC	Coordinate system option: EQ.0: Pass all variables in LS-DYNA local coordinate system EQ.1: Pass all variables in global coordinate system
XI	First isoparametric coordinate
ETA	Second isoparametric coordinate
WGT	Isoparametric weight
P1	First user defined element property.
P2	Second user defined element property.
⋮	⋮
PLCM	LCMth user defined element property.

Gaussian Quadrature Points

Point	1 Point	2 Points	3 Points	4 Points	5 Points
#1	.0	-.5773503	.0	-.8611363	.0
#2		+.5773503	-.7745967	-.3399810	-.9061798
#3			+.7745967	+.3399810	-.5384693
#4				+.8622363	+.5384693
#5					+.9061798

Point	6 Points	7 Points	8 Points	9 Points	10 Points
#1	-.9324695	-.9491080	-.9702896	-.9681602	-.9739066
#2	-.6612094	-.7415312	-.7966665	-.8360311	-.8650634
#3	-.2386192	-.4058452	-.5255324	-.6133714	-.6794096
#4	+.2386192	.0	-.1834346	-.3242534	-.4333954
#5	+.6612094	+.4058452	+.1834346	.0	-.1488743
#6	+.9324695	+.7415312	+.5255324	+.3242534	+.1488743
#7		+.9491080	+.7966665	+.6133714	+.4333954
#8			+.9702896	+.8360311	+.6794096
#9				+.9681602	+.8650634
#10					+.9739066

Location of through thickness Gauss integration points. The coordinate is referenced to the shell midsurface at location 0. The inner surface of the shell is at -1 and the outer surface is at +1.

Lobatto Quadrature Points

Point	1 Point	2 Points	3 Points	4 Points	5 Points
#1			0.0	-1.0	0.0
#2			-1.0	-0.4472136	-1.0
#3			+1.0	+0.4472136	-0.6546537
#4				+1.0	+0.6546537
#5					+1.0
Point	**6 Points**	**7 Points**	**8 Points**	**9 Points**	**10 Points**
#1	-1.0	-1.0	-1.0	-1.0	-1.0
#2	-0.7650553	-0.8302239	-0.8717401	-0.8997580	-0.9195339
#3	-0.2852315	-0.4688488	-0.5917002	-0.6771863	-0.7387739
#4	+0.2852315	0.0	-0.2092992	-0.3631175	-0.4779249
#5	+0.7650553	+0.4688488	+0.2092992	0.0	-0.1652790
#6	+1.0	+0.8302239	+0.5917002	+0.3631175	+0.1652790
#7		+1.0	+0.8717401	+0.6771863	+0.4779249
#8			+1.0	+0.8997580	+0.7387739
#9				+1.0	+0.9195339
#10					+1.0

Location of through thickness Lobatto integration points. The coordinate is referenced to the shell midsurface at location 0. The inner surface of the shell is at -1 and the outer surface is at +1.

Remarks:

1. **Formulation.** The default shell formulation is 2 unless overridden by THEORY in *CONTROL_SHELL. ELFORM in *SECTION_SHELL overrides THEORY.

   For implicit calculations the following element formulations are implemented: 2, 5, 6, 10, 12,12, 13, 14, 15, 16, -16, 17, 18, 20, 21, 22, 23, 24, 25, 26, 27, 29, 41, 42, 55.

   If another element formulation is requested for an implicit analysis, LS-DYNA will substitute one of the above in place of the one chosen.

2. **Linear Elements Type 18 and 20**. The linear elements consist of an assembly of membrane and plate elements. They have six degrees of freedom per node and can, therefore, be connected to beams, or used in complex shell surface

intersections. These elements possess the required zero energy rigid body modes and have exact constant strain and curvature representation, i.e. they pass all the first order patch tests. In addition, the elements have behavior approaching linear bending (cubic displacement) in the plate-bending configuration.

a) The membrane component is based on an 8-node/6-node isoparametric mother element which incorporates nodal in-plane rotations through cubic displacement constraints of the sides [Taylor 1987; Wilson 2000].

b) The plate component of element 18 is based on the Discrete Kirchhoff Quadrilateral (DKQ) [Batoz 1982]. Because the Kirchhoff assumption is enforced, the DKQ is transverse-shear rigid and can only be used for thin shells. No transverse shear stress information is available. The triangle is based on a degeneration of the DKQ. This element sometimes gives slightly lower eigenvalues when compared with element type 20.

c) The plate component of element 20 is based on the 8-node serendipity element. At the mid-side, the parallel rotations and transverse displacements are constrained and the normal rotations are condensed to yield a 4-node element. The element is based on thick plate theory and is recommended for thick and thin plates.

d) The quadrilateral elements contain a warpage correction using rigid links.

e) The membrane component of element 18 has a zero energy mode associated with in-plane rotations. This is automatically suppressed in a non-flat shell by the plate stiffness of the adjacent elements. In contrast, element 20 has no spurious zero energy modes.

3. **Linear Shear Element (22).** The linear shear panel element resist tangential in plane shearing along the four edges and can only be used with the elastic material constants of *MAT_ELASTIC. Membrane forces and out-of-plane loads are not resisted.

4. **Simplified Element for Time Domain Vibrations (99).** Element type 99 is intended for vibration studies carried out in the time domain. These models may have very large numbers of elements and may be run for relatively long durations. The purpose of this element is to achieve substantial CPU savings. This is achieved by imposing strict limitations on the range of applicability, thereby simplifying the calculations:

a) Elements must be rectangular; all edges must parallel to the global *x*-, *y*- or *z*-axis;

b) Small displacement, small strain, negligible rigid body rotation;

c) Elastic material only

If these conditions are satisfied, the performance of the element is similar to the fully integrated shell (ELFORM = 16) but at less CPU cost than the default Belytschko-Tsay shell element (ELFORM = 2). Single element torsion and in-plane bending modes are included; meshing guidelines are the same as for fully integrated shell elements.

No damping is included in the element formulation (e.g. volumetric damping). It is strongly recommended that damping be applied, e.g. *DAMPING_PART_-MASS or *DAMPING_FREQUENCY_RANGE.

5. **2D Formulations.** For 2D formulations (12-15, 46, 47), nodes must lie in the global x-y plane, i.e., the z-coordinate must be zero. Furthermore, the element normal should be in positive z direction. For axisymmetric element formulations, the global y-axis is taken as the axis of symmetry and all nodes must have x-coordinate values greater than or equal to 0.

   Shell thickness values on Card 2 are ignored by formulations 13, 14, and 15. For formulation 14 Input values of loads, lumped masses, discrete element stiffnesses, etc. in axisymmetric models are interpreted as values per unit circumference (i.e., per unit length in the circumferential direction) whereas for formulation 15 they are interpreted per radian. Output of forces for shell formulation 15 are in units of force per radian, e.g, as in bndout, nodfor, secforc, spcforc, rcforc. The units of forces output for shell formulation 14 are, at present, inconsistent. For defining contact in 2D simulations, see the entry for the *CONTACT_2D keyword.

6. **Shells with Thickness Stretch.** Shell element formulation 25 and 26 are the fully integrated shell element based on the Belytschko-Tsay element but with two additional degrees of freedom allowing for a linear variation of strain through the thickness. By default, the thickness field is continuous across the element edges implying that there can be no complex intersections since this would lock up the structure. It assumes a relatively flat surface and is intended primarily for sheets in metal forming. By specifying IDOF = 2, the thickness field is decoupled between elements which makes the element suited for crash. If there are any thickness stretch triangles (formulation 27), IDOF must be set to 2.

7. **Seatbelts.** Users must input a set of nodes along one of the transverse edges of a seatbelt. If there is no retractor associated with a belt, the node set can be on either edge. If the retractor exists, the edge should be on the retractor side and input in the same sequence of retractor node set. Therefore, another restriction on the seatbelt usage is each belt has its own section definition and a different part.

8. **Fracture.** XFEM 2D and shell formulations are recommended for brittle or semi-brittle fracture with pre-cracks, see *BOUNDARY_PRECRACK, or geometry imperfection such as a notch or a hole. XFEM shell formulation can be used for ductile fracture analysis with regularized effective plastic strain criterion.

9. **Discrete Kirchoff Theory Shell (17).** Shell element formulation 17 (DKT) is based on discrete Kirchhoff theory. It neglects out-of-plane shear strain energy and is thus valid only for thin plates where shear strain energy is negligible compared to bending energy.

10. **Limitations of Area-Weight Shell (14).** The exact stiffness matrix for the area-weighted shell formulation type 14 is nonsymmetric. The nonsymmetric terms are dropped for computational efficiency, making this formulation unsuitable for implicit linear analysis and eigenvalue analysis. It may, however, be used effectively for implicit nonlinear analysis.

    For explicit dynamics, viscous hourglass limits high frequency noise that may otherwise lead to nonphysical element distortion when nodes are on or near the axis of symmetry. Viscosity can be added to type 6 or 7 hourglass control by using VDC > 0 under the *HOURGLASS keyword. Alternatively, type 1 hourglass control is viscous.

11. **Eight Node Singular Shell (55).** The eight-noded singular element for fracture analysis is based on the eight-noded quadratic quadrilateral element. There are two ways to include the singularity around the crack tip:

    a) Move the two mid-nodes on the edges connected to the crack tip to the quarter location and obtain a strain singularity of $1/\sqrt{r}$

    b) Collapse the three nodes on one side of a quadrilateral element to the crack tip (but the three nodes remain independent) and move the two neighboring mid-nodes to the quarter location to obtain a strain singularity of $1/r$.

    This element uses 3×3 quadrature and is available to both implicit and explicit analyses.

    Cohesive Shell (29). Element type +/-29 is a cohesive element that models cohesive interfaces between shell element edges. The element takes bending forces into account and uses drilling force stabilization.

    Consider two shell elements in the same plane with nodes m_1, m_2, m_3, m_4, n_1, n_2, n_3, and n_4 such that the (m_3,m_4) edge and the (n_1,n_2) edge are connected through a cohesive shell having nodes m_4, m_3, n_2, and n_1, see Figure 36-23. The initial area of the cohesive element may be zero, in which case density is defined in terms of the length of the single connecting edge.

Element type +/-29 works similarly to solid element type 20. For example, extruding the two non-cohesive shells in their respective normal directions defines two 8 node solids. The cohesive mid-surface is located between the opposing faces of the solids, and tractions are calculated in four mid-surface points using differences of displacements between the opposing faces, giving rise to nodal forces and moments in the cohesive shell nodes m_4, m_3, n_2, and n_1. Additional details can be found in the Theory Manual.

Difference between type 29 and -29: In both formulations, the cohesive coordinate system direction q_2 is defined by the midpoints of the (n_1, m_4) and (n_2, m_3) edges. Type 29 defines the cohesive midsurface normal q_3 using the midpoints on the far side edges (m_1, m_2) and (n_3, n_4), while type -29 defines q_1 to be the mean of the neighboring element normals. Thus, in pure out-of-plane shear, type 29 will initially have pure tangential traction that turns into a normal traction as the separation increases, while type -29 will only have tangential traction.

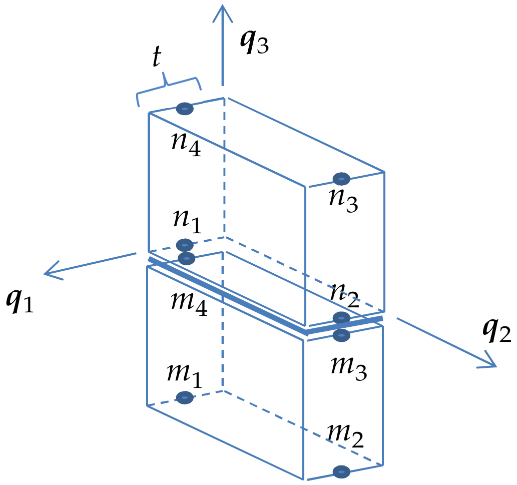


Figure [36-23]. Cohesive interface coordinate system for element type +/-29

12. **Accurate fully-integrated shell (-16).** Accuracy issues have been observed in shell formulation 16 under large deformations/rotations over a single time step. Formulation -16 is a correspondently enhanced version of formulation 16. Formulation 16 is unchanged to maintain back compatibility and, although, element -16 is supported in explicit mode it is primarily intended for implicit time integration. A strongly objective version can be activated by combining ELFORM = -16 and IACC = 1 on *CONTROL_ACCURACY, thereby ensuring that arbitrarily large rigid body rotation in a single step will transform stresses correctly and not generate any spurious strains.

13. **XFEM ductile fracture.** This feature is supported by a joint research among Honda, JSOL and LSTC. For FAILCR = -2, the failure strain is defined by: EPS = FS+(FS1-FS)*min(L/CL, 1.0) where L is the current crack length.

Example:

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$  *SECTION_SHELL
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  Define a shell section that specifies the following:
$     elform = 10  Belytschko-Wong-Chiang shell element formulation.
$        nip = 3   Three through the shell thickness integration points.
$    t1 - t4 = 2.0 A shell thickness of 2 mm at all nodes.
$
*SECTION_SHELL
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      sid    elform      shrf       nip     propt   qr/irid     icomp
         1        10               3.0000
$
$       t1        t2        t3        t4      nloc
       2.0       2.0       2.0       2.0
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

SECTION_SOLID_*{OPTION}

Available options include:

<BLANK>

EFG

SPG

Purpose: Define section properties for solid continuum and fluid elements.

Card Sets. For each unique solid section, include one set of data cards. The EFG option and the SPG option cannot both appear in the same model. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	SECID	ELFORM	AET					
Type	I/A	I	I					

EFG Card. Additional card for the EFG keyword option. See *CONTROL_EFG.

Card 2	1	2	3	4	5	6	7	8
Variable	DX	DY	DZ	ISPLINE	IDILA	IEBT	IDIM	TOLDEF
Type	F	F	F	I	I	I	I	F
Default	1.01	1.01	1.01	0	0	1	2	0.01

Optional EFG Card. Additional optional card for the EFG keyword option. See *CONTROL_EFG.

Card 3	1	2	3	4	5	6	7	8
Variable	IPS	STIME	IKEN	SF	CMID	IBR	DS	ECUT
Type	I	F	I	I	I	I	F	F
Default	0	10^{20}	0	0.0		1	1.01	0.1

SPG Card. Additional card for the SPG keyword option.

Card 2	1	2	3	4	5	6	7	8
Variable	DX	DY	DZ	ISPLINE	KERNEL	LSCALE	SMSTEP	SWTIME
Type	F	F	F	I	I	F	I	F
Default	1.50	1.50	1.50	0	0		15	

Optional SPG Card. Additional optional card for the SPG keyword option.

Card 3	1	2	3	4	5	6	7	8
Variable	IDAM	FS	STRETCH	ITB				
Type	I	F	F	I				
Default	0			0				

User Defined Element Card. Additional card for ELFORM = 101, 102, 103, 104 or 105. See Appendix C.

Card	1	2	3	4	5	6	7	8
Variable	NIP	NXDOF	IHGF	ITAJ	LMC	NHSV		
Type	I	I	I	I	I	I		
Default	0	0	0	0	0	0		

Integration Point Card. Additional card for ELFORM = 101, 102, 103, 104 or 105. Add NIP cards adhering to the format below. Because the default value for NIP is 0, these cards are read only for user-defined elements. See Appendix C.

Card	1	2	3	4	5	6	7	8
Variable	XI	ETA	ZETA	WGT				
Type	F	F	F	F				
Default	none	none	none	none				

Property Parameter Cards. Additional card for ELFORM = 101, 102, 103, 104 or 105. Add LMC property parameters by packing 8 parameters per card. See Appendix C.

Card	1	2	3	4	5	6	7	8
Variable	P1	P2	P3	P4	P5	P6	P7	P8
Type	F	F	F	F	F	F	F	F
Default	0	0	0	0	0	0	0	0

VARIABLE	DESCRIPTION
SECID	Section ID. SECID is referenced on the *PART card. A unique number or label must be specified.
ELFORM	Element formulation options. Remark 2 enumerates the element

VARIABLE	DESCRIPTION

formulations available for implicit calculations:

EQ.-2: Fully integrated S/R solid intended for elements with poor aspect ratio, accurate formulation (see Remark 15)

EQ.-1: Fully integrated S/R solid intended for elements with poor aspect ratio, efficient formulation (see Remark 15)

EQ.0: 1 point corotational for *MAT_MODIFIED_HONEYCOMB (see Remark 3)

EQ.1: Constant stress solid element: default element type. By specifying hourglass type 10 with this element, a Cosserat Point Element is invoked, see *CONTROL_HOURGLASS

EQ.2: Fully integrated S/R solid (see Remark 4)

EQ.3: Fully integrated quadratic 8 node element with nodal rotations

EQ.4: S/R quadratic tetrahedron element with nodal rotations

EQ.5: 1 point ALE

EQ.6: 1 point Eulerian

EQ.7: 1 point Eulerian ambient

EQ.8: Acoustic

EQ.9: 1 point corotational for *MAT_MODIFIED_HONEYCOMB (see Remark 3)

EQ.10: 1 point tetrahedron (see Remark 1)

EQ.11: 1 point ALE multi-material element

EQ.12: 1 point integration with single material and void

EQ.13: 1 point nodal pressure tetrahedron (see Remark 14)

EQ.14: 8 point acoustic

EQ.15: 2 point pentahedron element (see Remark 1)

EQ.16: 4 or 5 point 10-noded tetrahedron (see Remark 13). By specifying hourglass type 10 with this element, a Cosserat Point Element is invoked, see *CONTROL_HOURGLASS

VARIABLE **DESCRIPTION**

EQ.17: 10-noded composite tetrahedron (see Remark 13)

EQ.18: 8 point enhanced strain solid element for linear statics only (see Remark 4)

EQ.19: 8-noded, 4 point cohesive element (see Remarks 1 and 6)

EQ.20: 8-noded, 4 point cohesive element with offsets for use with shells (see Remarks 1, 6, and 8)

EQ.21: 6-noded, 1 point pentahedron cohesive element (see Remarks 1 and 7)

EQ.22: 6-noded, 1 point pentahedron cohesive element with offsets for use with shells (see Remarks 1, 7, and 8)

EQ.23: 20-node solid formulation

EQ.24: 27-noded, fully integrated S/R quadratic solid element

EQ.41: Mesh-free (EFG) solid formulation (see Remark 16)

EQ.42: Adaptive 4-noded mesh-free (EFG) solid formulation (see Remark 16)

EQ.43: Mesh-free enriched finite element

EQ.45: Tied Mesh-free enriched finite element

EQ.47: Smoothed particle Galerkin method (see Remark 17)

EQ.98: Interpolation solid

EQ.99: Simplified linear element for time-domain vibration studies (See Remark 5)

EQ.101: User defined solid

EQ.102: User defined solid

EQ.103: User defined solid

EQ.104: User defined solid

EQ.105: User defined solid

EQ.115: 1 point pentahedron element with hourglass control

GE.201: Isogeometric solids with NURBS. (See *ELEMENT_SOLID_NURBS_PATCH)

GE.1000: Generalized user-defined solid element formulation (See *DEFINE_ELEMENT_GENERALIZED_SOLID)

VARIABLE **DESCRIPTION**

AET

Ambient Element type: Can be defined for ELFORM 7, 11 and 12.

EQ.0: Non-ambient

EQ.1: Temperature (not currently available)

EQ.2: Pressure and temperature (not currently available)

EQ.3: Pressure outflow (obsolete)

EQ.4: Pressure inflow/outflow (Default for ELFORM 7)

EQ.5: Receptor for blast load (See *LOAD_BLAST_ENHANCED. Available only for ELFORM = 11)

DX, DY, DZ

Normalized dilation parameters of the kernel function in x, y and z directions. The normalized dilation parameters of the kernel function are introduced to provide the smoothness and compact support properties on the construction of the mesh-free shape functions. Values between 1.0 and 1.5 are recommended. Values smaller than 1.0 are not allowed. Larger values will increase the computation time and will sometimes result in a divergence problem.

ISPLINE

Replace the choice for the EFG kernel functions definition in *CONTROL_EFG. This allows users to define different ISPLINE in different sections.

EQ.0: Cubic spline function (default).

EQ.1: Quadratic spline function.

EQ.2: Cubic spline function with circular shape.

IDILA

Replace the choice for the normalized dilation parameter definition in *CONTROL_EFG. This allows users to define different IDILA in different sections.

EQ.0: Maximum distance based on the background elements.

EQ.1: Maximum distance based on surrounding nodes.

IEBT

Essential boundary condition treatment: See Remark 9 and 10.

EQ.1: Full transformation method (default)

EQ.-1: (w/o transformation)

EQ.2: Mixed transformation method

EQ.3: Coupled FEM/EFG method

VARIABLE	DESCRIPTION
	EQ.4: Fast transformation method EQ.-4: (w/o transformation) EQ.5: Fluid particle method for E.O.S and *MAT_ELASTIC_FLUID materials, currently supports only 4-noded background elements. EQ.7: Maximum entropy approximation
IDIM	Domain integration method: See Remark 11. EQ.1: Local boundary integration EQ.2: Two-point Gauss integration (default) EQ.3: Improved Gauss integration for IEBT = 4 or -4 EQ.-1: Stabilized EFG integration method (apply to 6-noded cell, 8-noded cell or combination of these two) EQ.-2: EFG fracture method (apply to 4-noded cell and SMP only)
TOLDEF	Deformation tolerance for the activation of adaptive EFG Semi-Lagrangian and Eulerian kernel. See Remark 12. EQ.0.0: Lagrangian kernel GT.0.0: Semi_Lagrangian kernel LT.0.0: Eulerian kernel
IPS	EQ.0: No pressure smoothing (default) EQ.1: Moving-least squared pressure recovery
STIME	Time to switch from stabilized EFG to standard EFG formulation
IKEN	EQ.0: Moving-least-square approximation (default, recommended) EQ.1: Maximum Entropy approximation
SF	Failure strain, recommended as an extra condition for the crack initiation under slow loading besides the stress-based cohesive law
CMID	Cohesive material ID for EFG fracture analysis (only Mode I crack is considered and only *MAT_COHESIVE_TH is available)

VARIABLE **DESCRIPTION**

IBR

EQ.1: No branching allowed

EQ.2: Branching is allowed

DS

Normalized support defined for computing the displacement jump in fracture analysis

ECUT

Define the minimum distance to the node that a crack surface can cut to the edge

KERNEL

Type of kernel approximation

EQ.0: updated Lagrangian kernel, no failure, less shear deformation

EQ.1: Eulerian kernel, failure analysis, global extreme deformation

EQ.2: Semi-pseudo Lagrangian kernel, failure analysis, local extreme deformation

LSCALE

Length scale for displacement regularization (not used yet)

SMSTEP

Interval of time steps to conduct displacement regularization

SWTIME

Time to switch from updated Lagrangian kernel to Eulerian kernel

IDAM

Option of damage mechanism

EQ.0: Continuum damage mechanics (default), the failed nodes are not eroded but converted to free nodes still carrying mass and momentum

EQ.1: Phenomenological strain damage

FS

Failure effective plastic strain if IDAM = 1

STRETCH

Stretching parameter if IDAM = 1

ITB

Flag for using stabilization

EQ.0: standard meshfree approximation + T-bond failure algorithm

EQ.1: fluid particle approximation (accurate but slow)

EQ.2: simplified fluid particle approximation (efficient and robust)

VARIABLE	DESCRIPTION
NIP	Number of integration points for user-defined solid (0 if resultant element)
NXDOF	Number of extra degrees of freedom per node for user-defined solid
IHGF	Flag for using hourglass stabilization (NIP.GT.0) EQ.0: Hourglass stabilization is not used EQ.1: LS-DYNA hourglass stabilization is used EQ.2: User-defined hourglass stabilization is used EQ.3: Same as 2, but the resultant material tangent moduli are passed
ITAJ	Flag for setting up finite element matrices (NIP.GT.0) EQ.0: Set up matrices wrt isoparametric domain EQ.1: Set up matrices wrt physical domain
LMC	Number of property parameters
NHSV	Number of history variables
XI	First isoparametric coordinate
ETA	Second isoparametric coordinate
ZETA	Third isoparametric coordinate
WGT	Isoparametric weight
PI	Ith property parameter

Remarks:

1. **ESORT to Stabilize Degenerate Solids.** The ESORT variable of the *CONTROL_SOLID keyword can be set to automatically convert degenerate tetrahedrons and degenerate pentahedrons into more suitable solid element formulations. The sorting is performed internally and is transparent to the user. See *CONTROL_SOLID for details.

2. **Implicit Analysis.** For implicit calculations the following element choices are implemented:

   EQ.-2: Fully integrated S/R solid element for poor aspect ratios, accurate formulation.

   EQ.-1: Fully integrated S/R solid element for poor aspect ratios, efficient formulation.

   EQ.1: Constant stress solid element.

   EQ.2: Fully integrated S/R solid.

   EQ.3: Fully integrated 8 node solid with rotational DOFs.

   EQ.4: Fully integrated S/R 4 node tetrahedron with rotational DOFs.

   EQ.10: 1 point tetrahedron.

   EQ.13: 1 point nodal pressure tetrahedron.

   EQ.15: 2 point pentahedron element.

   EQ.16: 5 point 10-noded tetrahedron.

   EQ.17: 10-noded composite tetrahedron.

   EQ.18: 8 point enhanced strain solid element for linear statics only.

   EQ.19: 8-noded, 4 point cohesive element (see Remarks 1 and 6)

   EQ.20: 8-noded, 4 point cohesive element with offsets for use with shells (see Remarks 1, 6, and 8)

   EQ.21: 6-noded, 1 point pentahedron cohesive element (see Remarks 1 and 7)

   EQ.22: 6-noded, 1 point pentahedron cohesive element with offsets for use with shells (see Remarks 1, 7, and 8)

   EQ.23: 20-node solid formulation

   EQ.24: 27-node solid formulation

   EQ.41: Mesh-free (EFG) solid formulation.

   EQ.42: 4-noded mesh-free (EFG) solid formulation.

   EQ.43: Mesh-free enriched finite element.

   If another element formulation is requested, LS-DYNA will substitute, when possible, one of the above in place of the one chosen. The type 1 element, constant stress, is generally much more accurate than the type 2 element, the selective reduced integrated element for implicit problems.

3. **Element for Modified Honeycomb Material.** Element formulations 0 and 9, applicable only to *MAT_MODIFIED_HONEYCOMB, behave essentially as

nonlinear springs so as to permit severe distortions sometimes seen in honeycomb materials. In formulation 0, the local coordinate system follows the element rotation whereas in formulation 9, the local coordinate system is based on axes passing through the centroids of the element faces. Formulation 0 is preferred for severe shear deformation where the barrier is fixed in space. If the barrier is attached to a moving body, which can rotate, then formulation 9 is usually preferred.

4. **Elements for Shear and Pressure Locking: Types 2 and 18.** The selective reduced integrated solid element, element type 2, assumes that pressure is constant throughout the element to avoid pressure locking during nearly incompressible flow. However, if the element aspect ratios are poor, shear locking will lead to an excessively stiff response. A better choice, given poor aspect ratios, is the one point solid element which work well for implicit and explicit calculations. For linear statics, the type 18 enhanced strain element works well with poor aspect ratios. Please note that highly distorted elements should always be avoided since excessive stiffness will still be observed even in the enhanced strain formulations.

5. **Element Type 99 for Vibration.** Element type 99 is intended for vibration studies carried out in the time domain. These models may have very large numbers of elements and may be run for relatively long durations. The purpose of this element is to achieve substantial CPU savings. This is achieved by imposing strict limitations on the range of applicability, thereby simplifying the calculations:

   a) Elements must be cubed; all edges must parallel to the global x-, y- or z-axis;

   b) Small displacement, small strain, negligible rigid body rotation;

   c) Elastic material only

   If these conditions are satisfied, the performance of the element is similar to the fully integrated S/R solid (ELFORM = 2) but at less CPU cost than the default solid element (ELFORM = 1). Single element bending and torsion modes are included, so meshing guidelines are the same as for fully integrated solids – e.g. relatively thin structures can be modeled with a single solid element through the thickness if required. Typically, the CPU requirement per element-cycle is roughly two thirds that of the default solid element.

   No damping is included in the element formulation (e.g. volumetric damping). It is strongly recommended that damping be applied, e.g. *DAMPING_PART_MASS or *DAMPING_FREQUENCY_RANGE.

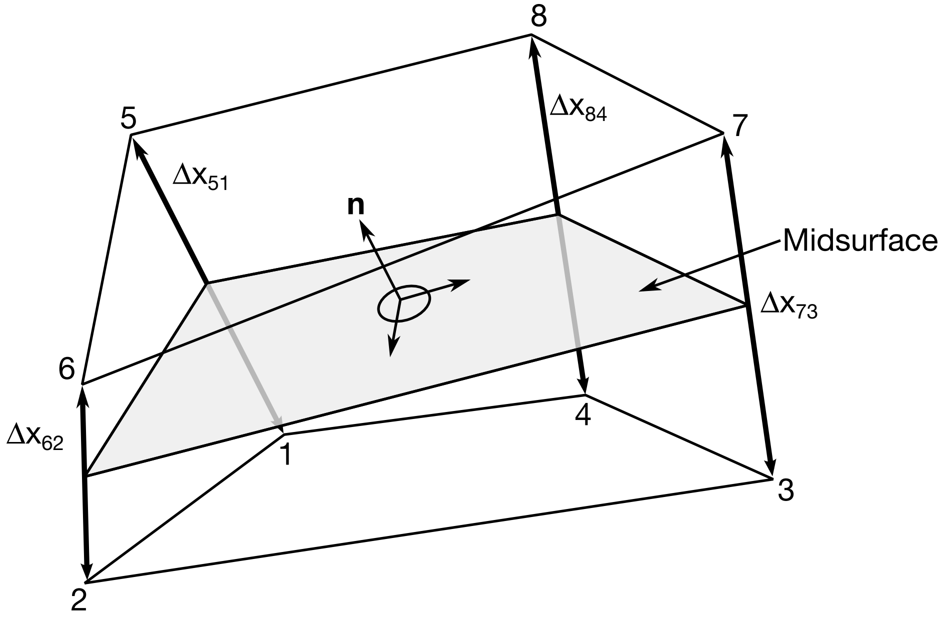


Figure 36-24. Illustration of solid local coordinates.

6. **8-Node Cohesive Element: Type 19.** Element type 19 is a cohesive element. The tractions on the mid-surface defined as the mid-points between the nodal pairs 1-5, 2-6, 3-7, and 4-8 are functions of the differences of the displacements between nodal pairs interpolated to the four integration points. The initial volume of the cohesive element may be zero, in which case, the density may be defined in terms of the area of nodes 1-2-3-4. See Appendix A and the user material description for additional details. See also *MAT_ADD_COHESIVE.

   The tractions are calculated in the local coordinate system defined at the centroid of the element, see the Figure 36-24. Defining the rotation matrix from the local to the global coordinate system at time t as $\mathbf{R}(t)$, the initial coordinates as $\mathbf{X}$, and the current coordinates as $\mathbf{x}$, the displacements at an integration point are

$$\Delta\mathbf{u} = \mathbf{R}^{\mathrm{T}}(t)\Delta\mathbf{x} - \mathbf{R}^{\mathrm{T}}(0)\Delta\mathbf{X}$$

$$\Delta\mathbf{x} = \sum_{i=1}^{4} N_i(s,t)\Delta\mathbf{x}_{i+4,i}$$

$$\Delta\mathbf{X} = \sum_{i=1}^{4} N_i(s,t)\Delta\mathbf{X}_{i+4,i}$$

   The forces are obtained by integrating the tractions over the mid-surface, and rotating them into the global coordinate system. It is the sum over integration points g = 1,2,3,4.

$$\mathbf{F}_i = \mathbf{R}(t)\sum_{g=1}^{4}\mathbf{T}_g N_i(s_g, t_g)\det(\mathrm{J}_g)\,, \quad \text{for } 1 \le i \le 4, \quad \text{and } \mathbf{F}_{i+4} = -\mathbf{F}_i$$

Where,

$\mathbf{T}_g$ = is the traction stress in the local coordinate system

N_i = The shape function of the cohesive element at node i

s_g and t_g = The parameteric coordinates of the 4 integration points

J_g = The integration point's portion of the determinate of the cohesive element which is equivalent to the element volume

7. **6-Node Cohesive Element: Type 21.** Element type 21 is the pentahedral counterpart to element type 19 with three nodes on the bottom and top surface. The tractions on the mid-surface are defined as the mid-points between the nodal pairs 1-5, 2-6, and 3-7 are functions of the differences of the displacements between nodal pairs interpolated to one integration point. The ordering of the nodal points in *ELEMENT_SOLID is given by:

   6-noded (cohesive) pentahedron   N1, N2, N3, N3, N5, N6, N7, N7, 0, 0

   Setting ESORT.gt.0 in *CONTROL_SOLID will automatically sort degenerated cohesive elements type 19 to cohesive pentahedron elements type 21.

8. **Cohesive Element with Offsets: Types 20 and 22.** Element type 20 is identical to element 19 but with offsets for use with shells. The element is assumed to be centered between two layers of shells on the cohesive element's lower (1-2-3-4) and upper (5-6-7-8) surfaces. The offset distances for both shells are one half the initial thicknesses of the nodal pairs (1-5, 2-6, 3-7, and 4-8) separating the two shells. These offsets are used with the nodal forces to calculate moments that are applied to the shells. Element type 20 in tied contacts will work correctly with the option, TIED_SHELL_EDGE_TO_SURFACE, which transmits moments. Other tied options will leave the rotational degrees-of-freedom unconstrained with the possibility that the rotational kinetic energy will cause a large growth in the energy ratio.

   Element type 22 is the pentahedron counterpart to element type 20 with three nodes on the bottom and top surface. The ordering of the nodal points in *ELEMENT_SOLID are identical to element type 21 (see Remark 7). Setting ESORT.GT.0 in *CONTROL_SOLID will automatically sort degenerated cohesive elements type 20 to cohesive pentahedron elements type 22.

9. **Automatic Sorting for EFG Background Mesh.** The current EFG formulation performs automatic sorting for finite element tetrahedral, pentahedron, and

hexahedral elements as the background mesh to identify the mesh-free geometry and provide the contact surface definition in the computation.

10. **Essential Boundary Conditions.** The mixed transformation method, the coupled FEM/EFG method and the fast transformation method were implemented in EFG 3D solid formulation. These three features were added to improve the efficiency on the imposition of essential boundary conditions and the transfer of real nodal values and generalized nodal values. The mixed transformation method is equivalent to the full transformation method with improved efficiency. The behavior of the coupled FEM/EFG method is between FEM and EFG. The fast transformation method provides the most efficient and robust results.

11. **IDIM.** For compressible material like foam and soil, IDIM=1 is recommended. For nearly incompressible material like metal and rubber, IDIM=2 (default) is recommended.

12. **TOLDEF.** This parameter is introduced to improve the negative volume problem usually seen during large deformation analysis. For the same analysis, the larger value of Toldef, the earlier Semi-Lagrangian or Eulerian kernel is introduced into the EFG computation and more cpu time is expected. Value between 0.0 and 0.1 is suggested in the crashworthiness analysis. Semi-Lagrangian kernel is suggested for the solid materials and Eulerian kernel is suggested for the fluid and E.O.S. materials.

13. **10-Node Tetrahedra: Types 16 and 17.** Formulations 16 and 17 are 10-noded, tetrahedral formulations. The parameter NIPTETS in *CONTROL_SOLID controls the number of integration points for these formulations. Formulation 17 is generally preferred over formulation 16 because, unlike 16, the nodal weighting factors are equal and thus nodal forces from contact and applied pressures are distributed correctly.

    When applying loads to 10-noded tetrahedrons via segments, no load will be applied to the midside nodes if the segments contain only corner nodes. When defining contact, it is recommended that *CONTACT_AUTOMATIC_… be used and the contact surface of the 10-noded tetrahedral part be specified by its part ID. In this manner, midside nodes receive contact forces.

    If the 10-noded element connectivity is not defined in accordance with the figure shown in *ELEMENT_SOLID, the order of the nodes can be quickly changed via a permutation vector specified with *CONTROL_SOLID. If *ELEMENT_SOLID defines 4-noded tetrahedrons, you can easily convert to 10-noded tetrahedrons using the command *ELEMENT_SOLID_TET4TOTET10. Because the characteristic length of a 10-noded tetrahedron is half that of a 4-noded tetrahedron, the time step for the tetrahedrons will be smaller by a factor of 2. The parameter TET10 in 971, when set to 1 in *CONTROL_OUTPUT, causes the full 10-node connectivity to be written to the d3plot and d3part databases.

14. **1-Point Nodal Pressure Tetrahedron: Type 13.** Element type 13 is identical with type 10 but with additional averaging of nodal pressures, which significantly lowers volumetric locking. Therefore, it is well suited for applications with incompressible and nearly incompressible material behavior, i.e. rubber materials or ductile metals with isochoric plastic deformations (e.g. bulk forming). Compared to the standard tetrahedron (type 10), a speed penalty of max. 25 % can be observed. In implicit, all material models supported by type 10 are also supported for this element, while for explicit currently material models *MAT_001, 003, 006, 007, 015, 024, 027, 077, 081, 082, 091, 092, 098, 103, 106, 120, 123, 124, 128, 129, 181, 183, 187, 224, 225, and 244 are fully supported. For other materials this element behaves like the type 10 tetrahedron.

15. **Fully Integrated S/R Solid Elements for Elements with Poor Aspect Ratio: Types -1 and -2.** Solid formulations -1 and -2 may offer improved behavior over formulation 2 by accounting for poor element aspect ratios in a manner so as to reduce the transverse shear locking effects seen in formulation 2. Type -1 is a more computationally efficient implementation of type -2, but a side-effect is that type -1's resistance to a particular deformation mode, similar to an hourglass mode, is weakened. This side effect is not truly hourglassing behavior and so there is no hourglass energy and behavior is not affected by hourglass parameters.

16. **EFG Solid Elements: Types 41 and 42.** EFG element type 41 supports 4-node, 6-node and 8-node solid elements. For 3D tetrahedron *r*-adaptive analysis (ADPOPT=7 in *CONTROL_ADAPTIVE and ADPOPT=2 in *PART), if the initial mesh is not purely comprised of tetrahedrons, element type 41 should be used instead of 42 causing the mesh to be converted automatically into tetrahedron after the first time step. Element type 42 only supports 4-node tetrahedron mesh, and is optimized to achieve better computational efficiency compared to 41.

17. **Smoothed Particle Galerkin (SPG) method: Type 47.** In SPG method, nodes are converted into particles and 4-node, 6-node and 8-node *solid* elements are supported. The method is suitable for severe deformation problem and failure analysis.

Example:

(Note: NODE_SET option is available starting with the R3 release of Version 971)

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$$$   *SECTION_SOLID
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  A bolt modeled with solids was found to have excessive hourglassing.
$  Thus, the section (sid = 116) associated with the bolt part was used
$  to specify that a fully integrated Selectively-Reduced solid element
$  formulation be used to totally eliminate the hourglassing (elform = 2).
$
*SECTION_SOLID
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      sid    elform
       116         2
$
*PART
bolts
$      pid       sid       mid     eosid      hgid    adpopt
        17       116         5
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

SECTION_SPH_*{OPTION}

Available options include:

<BLANK>

ELLIPSE

INTERACTION (See Remark 3)

USER (See Remark 2)

Purpose: Define section properties for SPH particles.

NOTE: This feature is not supported for use in implicit calculations.

Card Sets. For each SPH section add one set of cards 1 or 2 (depending on the keyword option). This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	SECID	CSLH	HMIN	HMAX	SPHINI	DEATH	START	
Type	I/A	F	F	F	F	F	F	
Default	none	1.2	0.2	2.0	0.0	1.e20	0.0	

Ellipse Card. Additional card for ELLIPSE keyword option.

Card 2	1	2	3	4	5	6	7	8
Variable	HXCSLH	HYCSLH	HZCSLH	HXINI	HYINI	HZINI		
Type	F	F	F	F	F	F		

VARIABLE	DESCRIPTION
SECID	Section ID. SECID is referenced on the *PART card. A unique number or label must be specified.

VARIABLE **DESCRIPTION**

CSLH Constant applied to the smoothing length of the particles. The default value applies for most problems. Values between 1.05 and 1.3 are acceptable. Taking a value less than 1 is inadmissible. Values larger than 1.3 will increase the computational time. The default value is recommended.

HMIN Scale factor for the minimum smoothing length (See Remark 1)

HMAX Scale factor for the maximum smoothing length (See Remark 1)

SPHINI Optional initial smoothing length (overrides true smoothing length). This option applies to avoid LS-DYNA to calculate the smoothing length during initialization. In this case, the variable CSLH doesn't apply.

DEATH Time imposed SPH approximation is stopped.

START Time imposed SPH approximation is activated.

HXCSLH Constant applied for the smoothing length in the x-direction for the ellipse case.

HYCSLH Constant applied for the smoothing length in the y-direction for the ellipse case.

HZCSLH Constant applied for the smoothing length in the z-direction for the ellipse case.

HXINI Optional initial smoothing length in the x-direction for the ellipse case (overrides true smoothing length)

HYINI Optional initial smoothing length in the y-direction for the ellipse case (overrides true smoothing length)

HZINI Optional initial smoothing length in the z-direction for the ellipse case (overrides true smoothing length)

Remarks:

1. **Smoothing Length.** The SPH processor in LS-DYNA employs a variable smoothing length. LS-DYNA computes the initial smoothing length, h_0, for each SPH part by taking the maximum of the minimum distance between every particle. Every particle has its own smoothing length which varies in time according to the following equation:

$$\frac{d}{dt}h(t) = h(t)\nabla \cdot \mathrm{v}$$

where $h(t)$ is the smoothing length, and where $\nabla \cdot \mathbf{v}$ is the divergence of the

flow. The smoothing length increases as particles separate and reduces as the concentration increases. This scheme is designed to hold constant the number of particles in each neighborhood. In addition to being governed by the above evolution equation the smoothing length is constrained to be between a user-defined upper and lower value

$$\mathrm{HMIN} \times h_0 < h(t) < \mathrm{HMAX} \times h_0.$$

Defining a value of 1 for HMIN and 1 for HMAX will result in a constant smoothing length in time and space.

2. **USER Option.** The USER option allows the definition of customized subroutine for the variation of the smoothing length. A subroutine called *hdot* is defined in the file dyn21.F (Unix/linux) or lsdyna.f (Windows).

3. **Contact/Partial Interaction.** Combined with CONT=1 in the *CONTROL_SPH card, this keyword option activates a partial interaction between SPH parts through the normal interpolation method and partially interact through the contact option. All the SPH parts defined using this keyword will interact with each other through normal interpolation method automatically.

*SECTION_TSHELL

Purpose: Define section properties for thick shell elements.

Card Sets. For each TSHELL section include a set of the following cards. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	SECID	ELFORM	SHRF	NIP	PROPT	QR	ICOMP	TSHEAR
Type	I/A	I	F	F	F	F	I	I
Default	none	1	1.0	2	1	0	0	0

Angle Cards. If ICOMP = 1 specify NIP angles putting 8 on each card. Include as many cards as necessary.

Card 2	1	2	3	4	5	6	7	8
Variable	B1	B2	B3	B4	B5	B6	B7	B8
Type	F	F	F	F	F	F	F	F

VARIABLE	DESCRIPTION
SECID	Section ID. SECID is referenced on the *PART card. A unique number or label must be specified.
ELFORM	Element formulation: EQ.1: one point reduced integration (default), EQ.2: selective reduced 2 × 2 in plane integration. EQ.3: assumed strain 2 × 2 in plane integration, see remark below. EQ.5: assumed strain reduced integration with brick materials EQ.6: assumed strain reduced integration with shell materials EQ.7: assumed strain 2 × 2 in plane integration.
SHRF	Shear factor. A value of 5/6 is recommended (see Remark 4).

VARIABLE DESCRIPTION

NIP

Number of through thickness integration points for the thick shell. See the variable INTGRD in *CONTROL_SHELL for details of the through thickness integration rule.

EQ.0: set to 2 integration points.

PROPT

Printout option:

EQ.1.0: average resultants and fiber lengths,

EQ.2.0: resultants at plan points and fiber lengths,

EQ.3.0: resultants, stresses at all points, fiber lengths.

QR

Quadrature rule:

LT.0.0: absolute value is specified rule number,

EQ.0.0: Gauss (up to five points are permitted),

EQ.1.0: trapezoidal, not recommended for accuracy reasons.

ICOMP

Flag for layered composite material mode:

EQ.1: a material angle is defined for each through thickness integration point. For each layer one integration point is used.

TSHEAR

Flag for transverse shear strain or stress distribution (see Remarks 4 and 5):

EQ.0.0: Parabolic,

EQ.1.0: Constant through thickness.

B1

β_1, material angle at first integration point. The same procedure for determining material directions is use for thick shells that is used for the 4 node quadrilateral shell.

B2

β_2, material angle at second integration point

B3

β_3, material angle at third integration point

⋮ ⋮

BNIP

β_{NIP}, material angle at eighth integration point

Remarks:

1. **Thick Shell Element Formulations.** Thick shell elements are bending elements that have 4 nodes on the bottom face and 4 on the top face. Thick shell element formulations 1, 2 and 6 are extruded thin shell elements and use thin shell material models and have an uncoupled stiffness in the z-direction. Thick shell element formulations 3, 5, and 7 are layered brick elements that use 3D brick material models. Element forms 3 and 5, and 6 are distortion sensitive and should not be used in situations where the elements are badly shaped. A single thick shell element through the thickness will capture bending response, but with element types 3, at least two are recommended to avoid excessive softness.

2. **Formulation 1 Quadrature Quirk.** When using Gauss quadrature with element formulation 1 the number of integration point is automatically switched to 3 when NIP = 2 and 5 when NIP = 4.

3. **Implicit Time Integration.** Thick shell elements are available for implicit analysis with the exception of thick shell formulation 1. If an element of type 1 is specified in an implicit analysis, it is internally switched to type 2

4. **SHRF Field.** For ELFORM=1, 2 and 6, the transverse shear stiffness is scaled by the SHRF parameter. Since the strain is assumed to be constant through the thickness, setting SHRF=5/6 is recommended to obtain the correct shear energy. For ELFORM=3 and 5, the SHRF parameter is not used, except for material types 33, 36, 133, 135, and 243. For ELFORM=3, the shear stiffness is assumed constant through the thickness. For ELFORM=5, 6, and 7, the shear distribution is assumed either parabolic if TSHEAR=0, or constant if TSHEAR=1. The parabolic assumption is good when the elements are used in a single layer to model a shell type structure, but the constant option may be better when elements are stacked one on top of the other.

5. **Modeling Composites.** Thick shell elements of all formulations can be used to model layered composites, but element formulations 5 and 6 use assumed strain to capture the complex Poisson's effects and through thickness stress distribution in layered composites. To define the layers of a composite, use QR < 0 to point to *INTEGRATION_SHELL data. Alternatively, the *PART_-COMPOSITE_TSHELL keyword offers a simplified way to define the layers.

   When modeling composites, laminated shell theory may be used to correct the transverse shear strain if the shear stiffness varies by layer. Laminated shell theory is activated by setting LAMSHT = 4 or 5 on *CONTROL_SHELL. When laminated shell theory is active, the TSHEAR parameter works with all ELFORM values to select either a parabolic or constant shear stress distribution.

*SENSOR

The keyword *SENSOR provides a convenient way of activating and deactivating boundary conditions, airbags, discrete elements, joints, contact, rigid walls, single point constraints, and constrained nodes. The sensor capability is new in the second release of version 971 and will evolve in later releases to encompass many more LS-DYNA capabilities and replace some of the existing capabilities such as the airbag sensor logic. The keyword commands in this section are defined below in alphabetical order:

*SENSOR_CONTROL

*SENSOR_CPM_AIRBAG

*SENSOR_DEFINE_CALC-MATH

*SENSOR_DEFINE_ELEMENT

*SENSOR_DEFINE_FORCE

*SENSOR_DEFINE_FUNCTION

*SENSOR_DEFINE_MISC

*SENSOR_DEFINE_NODE

*SENSOR_SWITCH

*SENSOR_SWITCH_CALC-LOGIC

*SENSOR_SWITCH_SHELL_TO_VENT

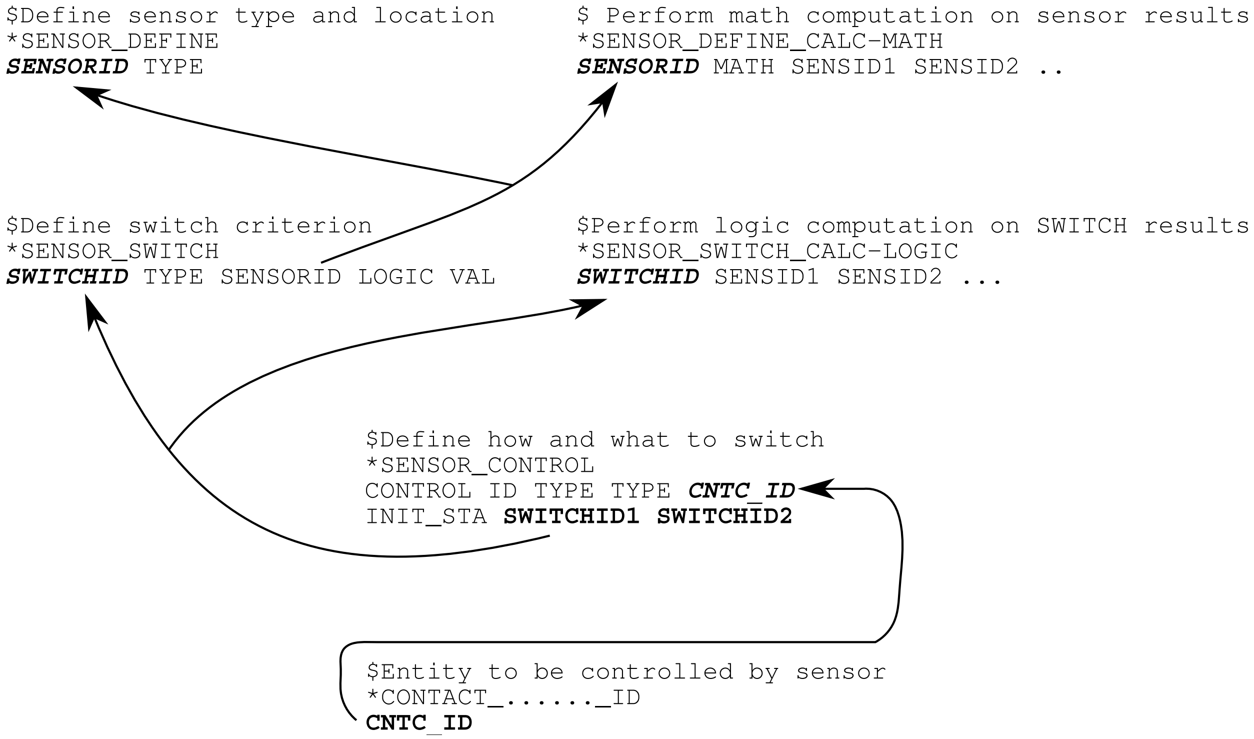
To define and utilize a sensor, three categories of sensor keyword commands are needed as shown in Figure 37-1.

1. Sensors are defined using the *SENSOR_DEFINE commands. Sensors provide a time history of model response that may be referred to by *SENSOR_SWITCH as a switching criterion. (Note: The time history of any sensor can be output using the SENSORD function in *DEFINE_CURVE_FUNCTION and *DATABASE_CURVOUT.)

   a) ***SENSOR_DEFINE(_ELEMENT,_FORCE,_MISC,_NODE)**

      These commands define a sensor's ID, type, and location..

   b) ***SENSOR_DEFINE_CALC-MATH, *SENSOR_DEFINE_FUNCTION**



```
$Define sensor type and location
*SENSOR_DEFINE
SENSORID TYPE
```

```
$ Perform math computation on sensor results
*SENSOR_DEFINE_CALC-MATH
SENSORID MATH SENSID1 SENSID2 ..
```

```
$Define switch criterion
*SENSOR_SWITCH
SWITCHID TYPE SENSORID LOGIC VAL
```

```
$Perform logic computation on SWITCH results
*SENSOR_SWITCH_CALC-LOGIC
SWITCHID SENSID1 SENSID2 ...
```

```
$Define how and what to switch
*SENSOR_CONTROL
CONTROL ID TYPE TYPE CNTC_ID
INIT_STA SWITCHID1 SWITCHID2
```

```
$Entity to be controlled by sensor
*CONTACT_......_ID
CNTC_ID
```

Figure 37-1. Relationship between sensor keyword definitions.

These commands define a sensor whose value is a mathmatical expression involving other sensors' values.

2. Sensor switching criterion definition using the *SENSOR_SWITCH keyword, which can be combined with the logical calculation command *SENSOR_SWITCH_CALC-LOGIC for more complicated definitions. The logic value yielded by this category of commands can be referred by *SENSOR_CONTROL to determine if a status switch condition is met.

   a) ***SENSOR_SWITCH**

   This command compares the numerical value from *SENSOR_DEFINE or *SENSOR_DEFINE_CALC-MATH with the given criterion to see if a switching condition is met.

   b) ***SENSOR_SWITCH_CALC-LOGIC**

   This command performs logical calculation on the information from SENSOR_SWITCH.

3. Sensor control definition, *SENSOR_CONTROL. This category of commands determines how and what to switch based on the logical values from *SENSOR_SWITCH and/or *SENSOR_SWITCH_CALC-LOGIC.

*SENSOR_CONTROL

Purpose: This command uses switches (*SENSOR_SWITCH) to toggle on or off the effects of other LS-DYNA keywords such as *CONTACT, or *AIRBAG.

Card Sets. For each sensor control add a pair of cards 1 and 2. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	CNTLID	TYPE	TYPEID	TIMEOFF	NREP			
Type	I	A	I	I	I			

Card 2	1	2	3	4	5	6	7	8
Variable	INITSTT	SWIT1	SWIT2	SWIT3	SWIT4	SWIT5	SWIT6	SWIT7
Type	A	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
CNTLID	Sensor control ID.
TYPE	Entity to be controlled:

EQ."AIRBAG":	*AIRBAG
EQ."BAGVENTPOP":	Opening and closing the airbag venting holes (see Remark 1)
EQ."BELTPRET":	Belt pretensioner firing (see Remark 2)
EQ."BELTRETRA":	Locking the belt retractor (see Remark 2)
EQ."BELTSLIP":	Controling the slippage of slip ring element (see Remark 3)
EQ."CONTACT":	*CONTACT
EQ."CONTACT2D":	*CONTACT_2D
EQ."CPM"	*AIRBAG_PARTICLE
EQ."DEF2RIG":	*DEFORMABLE_TO_RIGID_AUTOMATIC (see Remark 4)

VARIABLE **DESCRIPTION**

EQ.“ELESET”: Element set, see “ESTYP” below.

EQ.“FUNCTION”: *DEFINE_CURVE_FUNCTION (see Remarks 5 & 6)

EQ.“JOINT”: *CONSTRAINED_JOINT

EQ.“JOINTSTIF”: *CONSTRAINED_JOINT_STIFFNESS

EQ.“M PRESSURE”: *LOAD_MOVING_PRESSURE

EQ.“POREAIR”: *MAT_ADD_PORE_AIR

EQ.“PRESC-MOT”: *BOUNDARY_PRESCRIBED_MOTION

EQ.“PRESC-ORI”: *BOUNDARY_PRESCRIBED_ORIENTATION_RIGID

EQ.“PRESSURE”: *LOAD_SEGMENT_SET

EQ.“RWALL”: *RIGID_WALL

EQ.“SPC”: *BOUNDARY_SPC

EQ.“SPOTWELD”: *CONSTRAINED_SPOTWELD

TYPEID ID of entity to be controlled if TYPE is not set to FUNCTION. If TYPE is set to FUNCTION, see Remark 5. For TYPE = POREAIR, TYPEID is the ID of the part containing material with pore air.

TIMEOFF Flag for offset of time in curve:

EQ.0: No offset is applied.

EQ.1: Offset the abscissa of the time-dependent curve by the time value at which the sensor is triggered.

Type Of Control	Curve Affected
PRESSURE	*LOAD_SEGMENT
PRESC-MOT	*BOUNDARY_PRESCRIBED_MOTION
PRESC-ORI	*BOUNDARY_PRESCRIBED_ORIENTATION_RIGID

NREP Number of repeat of cycle of switches, SWITn, defined on the 2nd card. For example, a definition of SWITn like “601, 602, 601, 602, 601, 602” can be replaced by setting NREP to 3 and SWITn to ”601, 602”. Setting NREP = -1 repeats the cycle for infinite number of times. Default is 0.

VARIABLE	DESCRIPTION
ESTYP	Type of element set to be controlled. With initial status set to "ON", all the elements included in set TYPEID can be eroded when the controller status is changed to "OFF". When TYPEID is not defined, all elements of type ESTYP in the whole system will be eroded. EQ."BEAM": Beam element set. EQ."DISC": Discrete element set EQ."SHELL": Thin shell element set EQ."SOLID": Solid element set EQ."TSHELL": Thick shell element set
INITSTT	Initial status: EQ.On: Initial status is on EQ.Off: Initial status is off
SWIT*n*	ID of n^{th} switch. At the start of the calculation SWIT1 is *active*, meaning that it controls the state of the feature specified in TYPEID. After SWIT1 triggers, then SWIT2 becomes active; after SWIT2 triggers, then SWIT3 becomes active; this process will continue until the entire stack of switches has been exhausted.

Remarks:

1. **Activation of Bag Venting.** BAGVENTPOP activates (opens) or deactivates (closes) the venting holes of *AIRBAG_HYBRID and *AIRBAG_WANG_NEFSKE. It overwrites the definitions of PVENT of *AIRBAG_HYBRID and PPOP of *AIRBAG_WANG_NEFSKE. More than one SWIT can be input to open/close initially closed/opened holes, and then reclose/reopen the holes.

2. **Seatbelt Retractors.** The locking (or firing) of seatbelt retractor (or pretensioner) can be controlled through either general sensor, option BELTRETRA (or BELTPRET), or seatbelt sensors, *ELEMENT_SEATBELT_SENSOR. When BELTRETRA (or BELTPRET) is used, the SBSID*i* in *ELEMENT_SEATBELT_RETRACTOR (or PRETENSIONER) should be left blank.

3. **Seatbelt Slip Ring.** For one-way slip ring, a non-zero DIRECT in *ELEMENT_-SEATBELT_SLIPRING, BELTSLIP activates the constraint of one-way slippage when the status of SENSOR_CONTROL is on. When the SENSOR_CONTROL is turned off, the one-way slippage constraint is deactivated, therefore allowing slippage in both directions.

To model a two-way slip ring, BELTSLIP allow slippage in both directions when the status of SENSOR_CONTROL is on. When the status of SENSOR_-CONTROL is off, the slip ring lockup happens, no slippage is allowed then.

4. **Switching Between Rigid and Deformable.** DEF2RIG provides users more flexibility controlling material switch between rigid and deformable. Status of ON trigger the switch and deformable material becomes rigid. Rigidized material can then return to deformable status when status becomes OFF. As many as 7 SWITs can be input, any of them will change the status triggered by its preceding SWIT or the initial condition, INTSTT.

5. **Function for Sensor Control.** When the input parameter TYPE of *SENSOR_-CONTROL is set to "FUNCTION", the function "SENSOR(cntlid)" as described in *DEFINE_CURVE_FUNCTION takes on a value that depends on the current status of the *SENSOR_CONTROL. That status is either on or off at any given point in time. If the status is on, the value of function SENSOR(cntlid) is simply set to the integer value 1. If the status is off, the value of function SENSOR(cntlid) is set to the input parameter TYPEID (an integer) as specified in *SENSOR_CONTROL. To help clarify this relationship between *SENSOR_-CONTROL and *DEFINE_CURVE_FUNCTION, consider the following example.

6. **Example of Function Sensor Control.** Suppose a *SENSOR_CONTROL defined with CNTLID=101, TYPE="FUNCTION", and TYPEID = -2 has a status of off. Then a *DEFINE_CURVE_FUNCTION defined as “2+3*sensor(101)” will have a value of 2 + 3(-2) = -4. On the other hand, if the status of the *SENSOR_CONTROL changes to on, the *DEFINE_CURVE_FUNCTION takes on a value of 2 + 3(1) = 5.

*SENSOR_CPM_AIRBAG

Purpose: This command will associate a CPM airbag with a sensor switch (see *SENSOR_SWITCH). When the condition flag is raised, the specified CPM airbag will deploy. All time dependent curves used for the CPM airbag are shifted by the activation time including the *AIRBAG_PARTICLE curves for the inflator and vent as well as the *MAT_FABRIC curves for TSRFAC.

Card 1	1	2	3	4	5	6	7	8
Variable	CPMID	SWITID	TBIRTH	TDEATH	TDR	DEFPS	RBPID	
Type	I	I	F	F	F	I	I	

VARIABLE	DESCRIPTION
CPMID	Bag ID of *AIRBAG_PARTICLE_ID
SWITID	Switch ID of *SENSOR_SWITCH
TBIRTH	If SWITID is set, TBIRTH is not active. If SWITID is 0, TBIRTH is the activation time for the bag with ID = CPMID. All of the time dependent curves that are used in this bag will be offset by the value of TBIRTH.
TDEATH	Disable the CPMID bag when the simulation time exceeds this value.
TDR	If TDR is greater than 0 the bag with ID = CPMID will be rigid starting at first cycle and switch to deformable at time TDR.
DEFPS	Part set ID specifiying which parts of the bag with ID = CPMID are deformable.
RBPID	Part ID of the master rigid body to which the part is merged.

*SENSOR_DEFINE_CALC-MATH

Purpose: Defines a new sensor with a unique ID. The values associated with this sensor are computed by performing mathematical calculations with the information obtained from sensors defined by the *SENSOR_DEFINE_*OPTION*.

Math Sensor Cards. Include one additional card for each math sensor. This input ends at the next keyword ("*") card.

Card	1	2	3	4	5	6	7	8
Variable	SENSID	CALC	SENS1	SENS2	SENS3	SENS4	SENS5	SENS6
Type	I	A	I	I	I	I	I	I

VARIABLE **DESCRIPTION**

SENSID Sensor ID.

CALC Mathematical calculation, See Table 37-2.

SENS*i* *i*th Sensor ID

Remarks:

All sensors, SENS*i*, defined with either SENSOR_DEFINE_NODE_SET or SENSOR_DEFINE_ELEMENT_SET, must refer to either the same node set or the same element set.

Example:

```
$
$ assume set_2 to have 100 solid elements
*SENSOR_DEFINE_ELEMENT_SET
$ this sensor traces xx-strain of all 100 solid elements in set-2
        91     SOLID        -2        XX    STRAIN
*SENSOR_DEFINE_ELEMENT_SET
$ this sensor traces yy-strain of all 100 solid elements in set-2
        92     SOLID        -2        YY    STRAIN
*SENSOR_DEFINE_ELEMENT_SET
$ this sensor traces zz-strain of all 100 solid elements in set-2
        93     SOLID        -2        ZZ    STRAIN
*SENSOR_DEFINE_CALC-MATH
$ this sensor traces strain magnitudes of all 100 solid elements in set-2
       104  SQRTSQRE        91        92        93         0         0         0
*SENSOR_SWITCH
$ Because ELEMID of *sensor_define_element_set was input as "-2", SWITCH-1 will be
$ turned on if at least one of 100 elements has a strain magnitude>2.0E-4
$ On the other hand, If ELEMID was input as "2", SWITCH-1 will be turned on if
$ all 100 elements have strain magnitudes>2.0E-4
         1    SENSOR       104        GT    2.0E-4         0     0.001
$
```

FUNCTION	DESCRIPTION	MATHEMTAICAL FORM
ABSSUM	Absolute value of the sum of sensor values	$\lvert \mathrm{SENS1} + \mathrm{SENS2} + \cdots \rvert$
MIN	The minimum of sensor values	$\min(\mathrm{SENS1}, \mathrm{SENS2}, \dots)$
MAX	The maximum of sensor values	$\max(\mathrm{SENS1}, \mathrm{SENS2}, \dots)$
MAXMAG	The maximum of magnitude of sensor values	$\max(\lvert \mathrm{SENS1} \rvert, \lvert \mathrm{SENS2} \rvert, \dots)$
MINMAG	The minimum of the magnitude of sensor values	$\min(\lvert \mathrm{SENS1} \rvert, \lvert \mathrm{SENS2} \rvert, \dots)$
MULTIPLY	Multiplication of sensor values; negative for division (performed left to right)	$\mathrm{SENS1} \times \mathrm{SENS2} \times \cdots$
SQRE	Summation of squared values of sensor values	$\mathrm{SENS1}^2 + \mathrm{SENS2}^2 + \cdots$
SQRTSQRE	Square root of the sum of squared values	$\sqrt{\mathrm{SENS1}^2 + \mathrm{SENS2}^2 + \cdots}$
SQRT	Summation of square root of sensor values; negative for subtracting values	$\sqrt{\mathrm{SENS1}} + \sqrt{\mathrm{SENS2}} + \cdots$
SUMABS	Summation of absolute sensor values	$\lvert \mathrm{SENS1} \rvert + \lvert \mathrm{SENS2} \rvert + \cdots$
SUM	Summation of sensor values; negative for subtracting values	$\mathrm{SENS1} + \mathrm{SENS2} + \cdots$

Table 37-2. Available mathematical functions.

SENSOR_DEFINE_ELEMENT_*{OPTION}

Available options include:

<BLANK>

SET

Purpose: Define a strain gage type element sensor that checks the stress, strain, or resultant force of an element or element set.

Element Sensor Cards. Include one additional card for each element sensor. This input ends at the next keyword ("*") card.

Card	1	2	3	4	5	6	7	8
Variable	SENSID	ETYPE	ELEMID	COMP	CTYPE	LAYER	SF	PWR
Type	I	A	I	A	A	A/I	R	R

Optional card for SET option.

Card	1	2	3	4	5	6	7	8
Variable	SETOPT							
Type	A							

VARIABLE	DESCRIPTION
SENSID	Sensor ID.
ETYPE	Element type. Available options include:

EQ.BEAM:	beam element.
EQ.SHELL:	shell element
EQ.SOLID:	solid element
EQ.DISC-ELE:	discrete element
EQ.SEATBELT:	seatbelt element
EQ.TSHELL:	thick shell element

VARIABLE	DESCRIPTION
ELEMID	Element ID or element set ID when option_SET is active. In case of option_SET, a positive ELEMID requires all elements in set ELEMID to meet the switch condition to switch the status of related *SENSOR_SWITCH. If ELEMID is negative, the status of related *SENSOR_SWITCH will be changed if at least one of elements in set "-ELEMID" meets the switch condition.
COMP	Component type. The definition of component, and its related coordinate system, is consistent with that of elout. Leave blank for discrete elements. Available options for elements other than discrete element include: EQ.XX: *x*-normal component for shells and solids EQ.YY: *y*-normal component for shells and solids EQ.ZZ: *z*-normal component for shells and solids EQ.XY: *xy*-shear component for shells and solids EQ.YZ: *yz*-shear component for shells and solids EQ.ZX: *zx*-shear component for shells and solids EQ.AXIAL: axial EQ.SHEARS: local *s*-direction EQ.SHEART: local *t*-direction
CTYPE	Sensor type. Available options include: EQ.STRAIN: strain component for shells and solids EQ.STRESS: stress component for shells and solids EQ.FORCE: force resultants for beams, seatbelt, or translational discrete element; moment resultant for rotational discrete element EQ.MOMENT: moment resultants for beams EQ.DLEN: change in length for discrete or seatbelt element EQ.FAIL: failure of element, sensor value = 1 when element fails, = 0 otherwise.

VARIABLE **DESCRIPTION**

LAYER

Layer of integration point in shell or thick shell element. Options include:

EQ.BOT: component at lower surface meaning the integration point with the smallest through-the-thickness local coordinate

EQ.TOP: component at upper surface meaning the integration point with the largest through-the-thickness local coordinate

When CTYPE = STRESS, LAYER could be an integer “I” to monitor the stress of the I’the integration point.

SF, PWR

Optional parameters, scale factor and power, for users to adjust the resultant sensor value. The resultant sensor value is

$$[\mathrm{SF} \times (\mathrm{Original\ Value})]^{\mathrm{PWR}}$$

SETOPT

Option to process set of data when SET option is specified. More details can be found in *SENSOR_DEFINE_NODE_SET. When SETOPT is defined, a single value will be reported, which could be

EQ.AVG: the average value of the dataset

EQ.MAX: the maximum value of the dataset

EQ.MIN: the minimum value of the dataset

EQ.SUM: the sum of the dataset

*SENSOR_DEFINE_FORCE

Purpose: Define a force transducer type sensor.

Force Sensor Cards. Include one additional card for each force sensor. This input ends at the next keyword ("*") card.

Card	1	2	3	4	5	6	7	8
Variable	SENSID	FTYPE	TYPEID	VID	CRD			
Type	I	A	I	A/I	I			

VARIABLE **DESCRIPTION**

SENSID Sensor ID.

FTYPE Force type. See Table 37-3.

TYPEID ID defined in the associated KEYWORD command. See Table 37-3.

VID Vector along which the forces is measured.

EQ.X: *x*-direction in coordinate system CRD.

EQ.Y: *y*-direction in coordinate system CRD.

EQ.Z: *z*-direction in coordinate system CRD.

EQ.XMOMENT: *x*-direction moment for JOINT, JOINTSTIF, PRESC-MOT or SPC.

EQ.YMOMENT: *y*-direction moment for JOINT, JOINTSTIF, PRESC-MOT or SPC.

EQ.ZMOMENT: *z*-direction moment for JOINT, JOINTSTIF, PRESC-MOT or SPC.

VID∈{INT}: vector ID n in coordinate system CRD.

CRD Optional coordinate system, defined by *DEFINE_COORDINATE_NODES, to which vector VID is attached. If blank the global coordinate system is assumed.

FTYPE	TYPEID (Enter ID defined in following KEYWORD commands)	OUTPUT	ASCII FILE
AIRBAG	*AIRBAG	Airbag pressure	ABSTAT
CONTACT	*CONTACT	Contact force on the slave side	RCFORC
CONTACT2D	*CONTACT_2D	Contact force on the slave side	RCFORC
CPM	*AIRBAG_PARTICLE	Airbag pressure	AB-STAT_CPM
JOINT	*CONSTRAINED_JOINT	Joint force	JNTFORC
JOINTSTIF	*CONSTRAINED_JOINT_STIFFNESS	Joint stiffness force	JNTFORC
PRESC-MOT	*BOUNDARY_PRESCRIBED_MOTION	Prescribed motion force	BNDOUT
RWALL	*RIGIDWALL	Rigid wall force	RWFORC
SPC	*BOUNDARY_SPC	SPC reaction force	SPCFORC
SPOTWELD	*CONSTRAINED_POINTS	Spot weld force	SWFORC
X-SECTION	*DATABASE_CROSS_SECTION	Section force	SECFORC

Table 37-3. Force transducer type sensor

*SENSOR_DEFINE_FUNCTION

Purpose: Defines a new sensor with a unique ID. The value associated with this sensor is computed by performing mathematical calculations defined in *DEFINE_FUNCTION, with the information obtained from other sensors defined by the *SENSOR_DEFINE_*OPTION*.

Card 1	1	2	3	4	5	6	7	8
Variable	SENSID	FUNC	SENS1	SENS2	SENS3	SENS4	SENS5	SENS6
Type	I	I	I	I	I	I	I	I

Sensor Cards. Additional Cards needed when SENS1 < -5. Include as many cards as needed to specify all |SENS1| cards.

Card 2	1	2	3	4	5	6	7	8
Variable	SENSi	SENSi+1	SENSi+2	SENSi+3	SENSi+4	SENSi+5	SENSi+6	SENSi+7
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
SENSID	Sensor ID
FUNC	Function ID
SENS1	1st Sensor ID, the value of which will be used as the 1st argument of function FUNC. If defined as negative, the absolute value of SENS1, |SENS1|, is the number of sensors to be input. If |SENS1| > 5, additional cards will be needed to input the ID of all sensors. The number of sensor is limited to 15.
SENSi	ith Sensor ID, the value of which will be used as the ith argument of function FUNC

*SENSOR_DEFINE_MISC

Purpose: Trace the value of a miscellaneous item. This card replaces *SENSOR_DEFINE_ANGLE.

Force Sensor Cards. Include one additional card for each miscellaneous sensor. This input ends at the next keyword ("*") card.

Card	1	2	3	4	5	6	7	8
Variable	SENSID	MTYPE		I1	I2	I3	I4	I5
Type	I	A		I/A	I/A	I/A	I/A	I/A

VARIABLE	DESCRIPTION
SENSID	Sensor ID.

VARIABLE **DESCRIPTION**

MTYPE Entity to be traced:

EQ.ANGLE: Angular accelerometer sensor tracing the angle between two lines, 0≤ θ ≤180. The fields I1 and I2 are node numbers defining the 1st line, while I3 and I4 are node numbers defining the 2nd line.

EQ.CURVE: The value of a time-dependent curve defined by *DEFINE_CURVE_FUNCTION or *DEFINE_CURVE. I1 is the curve ID.

EQ.RETRACTOR: The seatbelt retractor payout rate is traced. I1 is the retractor ID.

EQ.RIGIDBODY: Accelerometer sensor tracing the kinematics of a rigid body with id I1. The I2 field specifies which kinematical component is to be traced. It may be set to “TX”, “TY”, or “TZ” for *X*, *Y*, and *Z* translations and to “RX”, “RY”, or “RZ” for the *X*, *Y*, and *Z* components of the rotation. The I3 field specifies the kinematics type: “D” for displacement, “V” for velocity and “A” for acceleration. Output is calculated with respect to the global coordinate system when the I4 field is set to “0”, its default value; the local rigid-body coordinate system is used when I4 is set to “1”.

EQ.TIME: The current analysis time is traced.

I1, …, I5 See MTYPE.

SENSOR_DEFINE_NODE_*{OPTION}

Available options include:

<BLANK>

SET

Purpose: Define an accelerometer type sensor. This command outputs the relative linear acceleration, velocity, or relative coordinate of node-1 with respect to node-2 along vector VID.

Node Sensor Cards. Include one additional card for each node sensor. This input ends at the next keyword ("*") card.

Card	1	2	3	4	5	6	7	8
Variable	SENSID	NODE1	NODE2	VID		CTYPE	SETOPT	
Type	I	I	I	I		A	A	

VARIABLE	DESCRIPTION
SENSID	Sensor ID.
NODE1,2	Nodes defining the accelerometer. NODE1 is a node set ID when option_SET is active. In case of option_SET, and when SETOPT is not defined, a positive NODE1 requires all nodes in set NODE1 to meet the switch condition to switch the status of related *SENSOR_SWITCH. If NODE1 is negative, the status of related *SENSOR_SWITCH will be changed if at least one of nodes in set "-NODE1" meets the switch condition.
VID	ID of vector along which the nodal values are measured, see *DEFINE_VECTOR. The magnitude of nodal values (coordinate, velocity or acceleration) will be output if VID is 0 or undefined.
CTYPE	Output component type (character string). EQ.ACC: acceleration EQ.VEL: velocity EQ.COORD: coordinate EQ.TEMP: temperature

VARIABLE	DESCRIPTION
SETOPT	Option to process set of data when SET option is specified. When SETOPT is specified, a single value will be reported, which could be EQ.AVG: the average value of the dataset EQ.MAX: the maximum value of the dataset EQ.MIN: the minimum value of the dataset EQ.SUM: the sum of the dataset

Remarks:

1. **Time Evolution of Vector VID.** The vector direction is determined by *DEFINE_VECTOR. This vector direction is updated with time only if the coordinate system CID (see *DEFINE_VECTOR) is defined using *DEFINE_COORDINATE_NODES and the parameter FLAG is set to 1. Otherwise, the vector direction is fixed.

2. **SETOPT.** When SETOPT is not defined for SET option, a list of nodal data will be reported, one for each node in the node set. These reported nodal values can be processed by *SENSOR_DEFINE_FUNCTION and *SENSOR_DEFINE_CALC-MATH, and then result in a list of processed values. These nodal values can be used to determine if the status of SENSOR_SWITCH will be changed. Depending on the sign of NODE1, it can take only one single data point or the whole data sets to meet the switch condition to change the status of the related SENSOR_SWITCH. It should be note that all sensor definitions referred to by these two processing commands must have the same number of data points. The reported nodal values cannot be accessed by commands like *DEFINE_CURVE_FUNCTION, see SENSORD option.

   When SETOPT is defined, the nodal values of all nodes in the node set will be processed, depending on the definition of SETOPT, the resulting value is reported as the single sensor value. The reported value can be processed by both *SENSOR_DEFINE_FUNCTION and *SENSOR_DEFINE_CALC-MATH as well as other regular sensors. This reported value can also be accessed by * DEFINE_CURVE_FUNCTION using SENSORD option. If the reference node, node2, is needed, NODE2 has to be a node set containing the same number of nodes as node set “-NODE1”. The nodes in sets “-NODE1” and “NODE2” have to be arranged in the same sequence so that the nodal value of nodes in set “-NODE1” can be measured with respect to the correct node in set NODE2.

3. When NODE1 is a node that belongs to a rigid body and CTTYPE=”ACC”, the acceleration recorded by the sensor is not updated every time step but rather

only when nodal output is written according to *DATABASE commands. As an alternative, *SENSOR_DEFINE_MISC with MTYPE=”RIGIDBODY” updates the sensor data every time step.

*SENSOR_SWITCH

Purpose: This command compares the value of a sensor, *SENSOR_DEFINE or SENSOR_CALC-MATH, to a given criterion to check if the switch condition is met. It output a logic value of TRUE or FALSE.

Sensor Switch Cards. Include one additional card for each sensor switch. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	SWITID		SENSID	LOGIC	VALUE	FILTRID	TIMWIN	
Type	I		I	A	F	I	F	

VARIABLE	DESCRIPTION
SWITID	Switch ID can be referred directly by *SENSOR_CONTROL to control the status of entities like CONTACT and AIRBAG, or can be referred to by *SENSOR_SWITCH_CALC-LOGIC for logic computation.
SENSID	ID of the sensor whose value will be compared to the criterion to determine if a switch condition is met.
LOGIC	Logic operator, could be either LT (<) or GT (>).
VALUE	Critical value
FILTER	Filter ID (optional). Filters may be defined using *DEFINE_FILTER.
TIMWIN	Trigger a status change when the value given by the sensor is less than or greater than (depending on LOGIC) the VALUE for a duration defined by TIMWIN.

*SENSOR_SWITCH_CALC-LOGIC

Purpose: This command performs a logic calculation for the logic output of up to seven *SENSOR_SWITCH or *SENSOR_SWITCH_CALC-LOGIC definitions. The output is a logic value of either TRUE or FALSE.

Log Cards. Include one additional card for each logic rule. This input ends at the next keyword (“*”) card.

Card	1	2	3	4	5	6	7	8
Variable	SWITID	SWIT1	SWIT2	SWIT3	SWIT4	SWIT5	SWIT6	SWIT7
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
SWITID	Switch ID can be referred directly by *SENSOR_CONTROL to control the status of entities like CONTACT and AIRBAG, or can be referred to by *SENSOR_SWITCH_CALC-LOGIC for logic computation.
SWITn	Input a positive sensor switch ID for "AND" and negative sensor switch ID for "OR". SWIT1 must always be positive.

This keyword implements standard Boolean logic.

$$\text{true} = 1,$$
$$\text{false} = 0,$$
$$\text{and} = \text{multiplication},$$
$$\text{or} = \text{addition}$$

An expression evaluating to 0 is false, while any expression that evaluates to greater than 0 is true, and, therefore, set to 1.

Example:

Consider 5 switches defined as follows:

$$\text{switch}(11) = \text{true}$$
$$\text{switch}(12) = \text{false}$$
$$\text{switch}(13) = \text{true}$$
$$\text{switch}(14) = \text{true}$$

$$\text{switch}(15) = \text{false.}$$

To evaluate the expression

$$[\text{switch}(11) \text{ or switch}(12) \text{ or switch}(13)] \text{ and } [\text{switch}(14) \text{ or switch}(15)]$$

and assign the value to switch(103), the following would apply:

```
*SENSOR_SWITCH_CALC-LOGIC
101,11,-12,-13
102,14,-15
103,101,102
```

This translates into

$$\begin{aligned}
\text{switch}(101) &= \text{switch}(11) \text{ or switch}(12) \text{ or switch}(13) \\
&= \min((1\ +\ 0\ +\ 1),1) \\
&= 1 \text{ (true)} \\
\text{switch}(102) &= \text{switch}(14) \text{ or switch}(15) \\
&= \min((1+0),1) \\
&= 1 \text{ (true)} \\
\text{switch}(103) &= \text{switch}(101) \text{ and switch}(102) \\
&= \min((1 \times 1),1) \\
&= 1 \text{ (true)}
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\overbrace{[\text{switch}(11) \text{ or } s\text{witch}(12) \text{ or switch}(13)]}^{\text{switch}(101)=\text{true}} \text{ AND } \overbrace{[\text{switch}(14) \text{ or switch}(15)]}^{\text{switch}(102)=\text{true}} \\
&= \text{switch}(103) \\
&= \text{true}
\end{aligned}$$

*SENSOR_SWITCH_SHELL_TO_VENT

Purpose: This option will treat the failed shell elements as vent hole for the airbag defined by *AIRBAG_PARTICLE. The mass escaped from the vent will be reported in `abstat_cpm` file.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	ITYPE	C23					
Type	I	I	F					

Shell Fail Time Cards. Optional Cards for setting time at which shells in a shell list (see *SHELL_LIST) change into vents. This card may be repeated up time 15 times. This input ends at the next keyword ("*") cards.

Optional	1	2	3	4	5	6	7	8
Variable	SSID	FTIME	C23V					
Type	I	F	F					
Default	none	0.	C23					

VARIABLE **DESCRIPTION**

ID Part set ID/Part ID.

TYPE

EQ.0: Part

EQ.1: Part set

C23 Vent Coefficient (Default = 0.7)

LT.0: User defined load curve ID. The vent coefficient will be determined by this pressure-vent_coeff curve.

SSID ID of *SET_SHELL_LIST

FTIME Time to convert shell list to vent. (Default is from t = 0.)

VARIABLE	DESCRIPTION
C23V	Vent Coefficient (Default = C23) LT.0: User defined load curve ID. The vent coefficient will be determined by this pressure-vent_coeff curve.

*SET

The keyword *SET provides a convenient way of defining groups of nodes, parts, elements, and segments. The sets can be used in the definitions of contact interfaces, loading conditions, boundary conditions, and other inputs. The keyword provides also a convenient way of defining groups of vibration modes to be used in frequency domain analysis. Each set type must have a unique numeric identification. The keyword control cards in this section are defined in alphabetical order:

SET_BEAM_*{OPTION}_{OPTION}

*SET_BEAM_ADD

*SET_BEAM_INTERSECT

*SET_BOX

SET_DISCRETE_*{OPTION}_{OPTION}

*SET_DISCRETE_ADD

SET_MODE_*{OPTION}

*SET_MULTIMATERIAL_GROUP_LIST

SET_NODE_*{OPTION}_{OPTION}

SET_NODE_ADD_*{OPTION}

*SET_NODE_INTERSECT

SET_PART_*{OPTION}_{OPTION}

*SET_PART_ADD

SET_SEGMENT_*{OPTION}_{OPTION}

*SET_SEGMENT_ADD

*SET_SEGMENT_INTERSECT

SET_2D_SEGMENT_*{OPTION}_{OPTION}

SET_SHELL_*{OPTION}_{OPTION}

*SET_SHELL_ADD

*SET_SHELL_INTERSECT

SET_SOLID_*{OPTION}_{OPTION}

*SET_SOLID_ADD

*SET_SOLIDT_INTERSECT

SET_TSHELL_*{OPTION}_{OPTION}

An additional option**_TITLE** may be appended to all the ***SET** keywords. If this option is used then an addition line is read for each section in 80a format which can be used to describe the set. At present LS-DYNA does make use of the title. Inclusion of titles gives greater clarity to input decks.

The **GENERAL** option is available for set definitions. In this option, the commands are executed in the order defined. For example, the delete option cannot delete a node or element unless the node or element was previously added via a command such as BOX or ALL.

The **COLLECT** option allows for the definition of multiple sets that share the same ID and combines them into one large set whenever this option is found. If two or more like sets definitions share the same IDs, they are combined if and only if the**_COLLECT** option is specified in each definition. If the**_COLLECT** option is not specified for one or more like set definitions that share identical ID's an error termination will occur. For include files using ***INCLUDE_TRANSFORM** where set offsets are specified, the offsets are not applied for the case where the**_COLLECT** option is present.

SET_BEAM_*{OPTION1}_{OPTION2}

For *OPTION1* the available options are:

<BLANK>

GENERATE

GENERATE_INCREMENT

GENERAL

For *OPTION2* the available option is:

COLLECT

The GENERATE and GENERATE_INCREMENT options will generate block(s) of beam element ID's between a starting ID and an ending ID. An arbitrary number of blocks can be specified to define the set.

Purpose: Define a set of beam elements or a set of seat belt elements (see BSID of *DATABASE_CROSS_SECTION_SET).

Card 1	1	2	3	4	5	6	7	8
Variable	SID							
Type	I							
Default	none							

Beam Element ID Cards. This Card 2 format applies to the case of an unset (<BLANK>) keyword option. Set one value per element in the set. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	K1	K2	K3	K4	K5	K6	K7	K8
Type	I	I	I	I	I	I	I	I

Beam Element Range Cards. This Card 2 format applies to the GENERATE keyword option. Set one pair of BNBEG and BNEND values per block of elements. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	B1BEG	B1END	B2BEG	B2END	B3BEG	B3END	B4BEG	B4END
Type	I	I	I	I	I	I	I	I

Beam Element Range with Increment Cards. This Card 2 format applies to the GENERATE_INCREMENT keyword option. For each block of elements add one card to the deck. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	BBEG	BEND	INCR					
Type	I	I	I					

Generalized Beam Element Range Cards. This Card 2 format applies to the GENERAL keyword option. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2…	1	2	3	4	5	6	7	8
Variable	OPTION	E1	E2	E3	E4	E5	E6	E7
Type	A	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
SID	Set ID
K1	First beam element
K2	Second beam element
⋮	⋮
B[N]BEG	First beam element ID in block N.

VARIABLE	DESCRIPTION
B[N]END	Last beam element ID in block N. All defined ID's between and including B[N]BEG to B[N]END are added to the set. These sets are generated after all input is read so that gaps in the element numbering are not a problem. B[N]BEG and B[N]END may simply be limits on the ID's and not element ID's.
BBEG	First beam element ID in block.
BEND	Last beam element ID in block.
INCR	Beam ID increment. Beam IDs BBEG, BBEG + INCR, BBEG + 2 × INCR, and so on through BEND are added to the set.
OPTION	Option for GENERAL. See table below.
E1, …, E7	Specified entity. Each card must have the option specified. See table below.

The General Option:

The "OPTION" column in the table below enumerates the allowed values for the "OPTION" variable in Card 2 for the GENERAL option. Likewise, the variables E1, …, E7 refer to the GENERAL option Card 2.

Each of the following operations accept up to 7 arguments, but they may take fewer. Values of "En" left unspecified are ignored.

OPTION	DESCRIPTION
ALL	All beam elements will be included in the set.
ELEM	Elements E1, E2, E3, ... will be included.
DELEM	Elements E1, E2, E3, ... previously added will be excluded.
PART	Elements of parts E1, E2, E3, ... will be included.
DPART	Elements of parts E1, E2, E3, ... previously added will be excluded.
BOX	Elements inside boxes E1, E2, E3, ... will be included. (see *DEFINE_BOX)
DBOX	Elements inside boxes E1, E2, E3, ... previously added will be excluded.

*SET_BEAM_ADD

Purpose: Define a beam set by combining beam sets.

Card 1	1	2	3	4	5	6	7	8
Variable	SID							
Type	I							
Default	none							

Beam Element Set Cards. Each card can be used to specify up to 8 beam element sets. Include as many cards of this kind as necessary. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	BSID1	BSID2	BSID3	BSID4	BSID5	BSID6	BSID7	BSID8
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
SID	Set ID of new beam set. All beam sets should have a unique set ID.
BSID[N]	The Nth beam set ID on the card

*SET_BEAM_INTERSECT

Purpose: Define a beam set as the intersection, ∩, of a series of beam sets. The new beam set, SID, contains only the elements common to of all beam sets listed on the cards of format 2.

Card 1	1	2	3	4	5	6	7	8
Variable	SID							
Type	I							
Default	none							

Beam Set Cards. Each card can be used to specify up to 8 beam sets. Include as many cards of this kind as necessary. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	BSID1	BSID2	BSID3	BSID4	BSID5	BSID6	BSID7	BSID8
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
SID	Set ID of new beam set. All beam sets should have a unique set ID.
BSID[N]	The Nth beam set ID on card2

*SET_BOX

Purpose: Define a set of boxes. The new box set, SID, contains a set of box IDs listed on the cards of format 2. Refer box ID in *DEFINE_BOX.

Card 1	1	2	3	4	5	6	7	8
Variable	SID							
Type	I							
Default	none							

Box Set Cards. Each card can be used to specify up to 8 box IDs. Include as many cards of this kind as necessary. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	BID1	BID2	BID3	BID4	BID5	BID6	BID7	BID8
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
SID	Set ID of new box set. All box sets should have a unique set ID.
BID[N]	The Nth box ID on card2

SET_DISCRETE_*{OPTION1}_{OPTION2}

For *OPTION1* the available options are:

<BLANK>

GENERATE

GENERAL

For *OPTION2* the available option is:

COLLECT

The option GENERATE will generate a block of discrete element ID's between a starting ID and an ending ID. An arbitrary number of blocks can be specified to define the set.

Purpose: Define a set of discrete elements.

Card 1	1	2	3	4	5	6	7	8
Variable	SID							
Type	I							
Default	none							

Discrete Element ID Cards. This Card 2 format applies to the case of an unset (<BLANK>) keyword option. Set one value per element in the set. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	K1	K2	K3	K4	K5	K6	K7	K8
Type	I	I	I	I	I	I	I	I

Discrete Element Range Cards. This Card 2 format applies to the GENERATE keyword option. Set one pair of BNBEG and BNEND values per block of elements. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	B1BEG	B1END	B2BEG	B2END	B3BEG	B3END	B4BEG	B4END
Type	I	I	I	I	I	I	I	I

Generalized Discrete Element Range Cards. This Card 2 format applies to the GENERAL keyword option. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	OPTION	E1	E2	E3	E4	E5	E6	E7
Type	A	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
SID	Set ID
K1	First discrete element
K2	Second discrete element
⋮	⋮
B[N]BEG	First discrete element ID in block N.
B[N]END	Last discrete element ID in block N. All defined ID's between and including B[N]BEG to B[N]END are added to the set. These sets are generated after all input is read so that gaps in the element numbering are not a problem. B[N]BEG and B[N]END may simply be limits on the ID's and not element ID's.
OPTION	Option for GENERAL. See table below.
E1, …, E7	Specified entity. Each card must have the option specified. See table below.

The General Option:

The "OPTION" column in the table below enumerates the allowed values for the "OPTION" variable in Card 2 for the GENERAL option. Likewise, the variables E1, …, E7 refer to the GENERAL option Card 2.

Each of the following operations accept up to 7 arguments, but they may take fewer. Values of "En" left unspecified are ignored.

OPTION	DESCRIPTION
ALL	All discrete elements will be included in the set.
ELEM	Elements E1, E2, E3, ... will be included.
DELEM	Elements E1, E2, E3, ... previously added will be excluded.
PART	Elements of parts E1, E2, E3, ... will be included.
DPART	Elements of parts E1, E2, E3, ... previously added will be excluded.
BOX	Elements inside boxes E1, E2, E3, ... will be included. (see *DEFINE_BOX)
DBOX	Elements inside boxes E1, E2, E3, ... previously added will be excluded.

*SET_DISCRETE_ADD

Purpose: Define a discrete set by combining discrete sets.

Card 1	1	2	3	4	5	6	7	8
Variable	SID							
Type	I							
Default	none							

Discrete Element Set Cards. Each card can be used to specify up to 8 discrete element sets. Include as many cards of this kind as necessary. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	DSID1	DSID2	DSID3	DSID4	DSID5	DSID6	DSID7	DSID8
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
SID	Set ID of new discrete element set. All discrete element sets must have a unique ID.
DSID[N]	The Nth discrete set ID on card2

SET_MODE_*{OPTION}

Available options include:

<BLANK>

LIST

LIST_GENERATE

The last option, LIST_GENERATE, will generate a block of mode ID's between a starting ID and an ending ID. An arbitrary number of blocks can be specified to define the set.

Purpose: Define a set of modes.

Card 1	1	2	3	4	5	6	7	8
Variable	SID							
Type	I							
Default	none							

Mode ID Cards. This Card 2 format applies to the for keyword option set to LIST *or* for an unset (<BLANK>) keyword option. Set one value per mode in the set. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	MID1	MID2	MID3	MID4	MID5	MID6	MID7	MID8
Type	I	I	I	I	I	I	I	I

Mode Range Cards. This Card 2 format applies to the GENERATE keyword option. Set one pair of BNBEG and BNEND values per block of modes. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	M1BEG	M1END	M2BEG	M2END	M3BEG	M3END	M4BEG	M4END
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
SID	Set identification. All mode sets should have a unique set ID.
MID[N]	Mode ID N.
M[N]BEG	First mode ID in block N.
M[N]END	Last mode ID in block N. All defined ID's between and including M[N]BEG and M[N]END are added to the set.

Remarks:

1. The available mode ID's can be found in ASCII file `eigout`, or binary database `d3eigv`.

*SET_MULTI

Note that this keyword's name has been shortened. Its older long form, however, is still also valid.

*SET_MULTI-MATERIAL_GROUP_LIST

Purpose: This command defines an ALE multi-material set ID (AMMSID) which contains a collection of one or more ALE multi-material group ID(s) (AMMGID). This provides a means for selecting any specific ALE multi-material(s). Application includes, for example, a selection of any particular fluid(s) to be coupled to a fluid-structure interaction.

Card 1	1	2	3	4	5	6	7	8
Variable	AMSID							
Type	I							
Default	0							

Multi-Material Group ID Cards. Set one value per element in the set. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	AMGID1	AMGID2	AMGID3	AMGID4	AMGID5	AMGID6	AMGID7	AMGID8
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

VARIABLE	DESCRIPTION
AMSID	An ALE multi-material set ID (AMSID) which contains a collection of one or more ALE multi-material group ID(s) (AMGID).
AMGID1	The 1st ALE multi-material group ID (AMGID = 1) defined by the 1st data line of the *ALE_MULTI-MATERIAL_GROUP card.
⋮	⋮

VARIABLE	DESCRIPTION
AMGID8	The 8th ALE multi-material group ID (AMGID = 1) defined by the 8th data line of the *ALE_MULTI-MATERIAL_GROUP card.

Remarks:

1. Refer to an example in the *CONSTRAINED_LAGRANGE_IN_SOLID section.

*SET_NODE_*{OPTION1}*_*{OPTION2}*

For *OPTION1* the available options are:

<BLANK>

LIST

COLUMN

LIST_GENERATE

LIST_GENERATE_INCREMENT

GENERAL

LIST_SMOOTH

For *OPTION2* the available option is:

COLLECT

The LIST option generates a set for a list of node IDs. The LIST_GENERATE and LIST_GENERATE_INCREMENT options will generate block(s) of node IDs between a starting ID and an ending ID. An arbitrary number of blocks can be specified to define the node set. The option LIST_SMOOTH is used to define a local region on a distorted tooling mesh to be smoothed. The LIST_SMOOTH option is documented in the **Local smoothing of tooling mesh** section of the *INTERFACE_COMPENSATION_NEW card's documentation. The COLUMN option is for setting nodal attributes, which pass data to other keyword cards, on a node-by-node basis.

Purpose: Define a nodal set with some identical or unique attributes.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	DA1	DA2	DA3	DA4	SOLVER		
Type	I	F	F	F	F	A		
Default	none	0.	0.	0.	0.	MECH		
Remark		1	1	1	1	3		

Node ID Cards. This Card 2 format applies to LIST and LIST_SMOOTH keyword options. Additionally, it applies to the case of an unset (<BLANK>) keyword option. Set one value per node in the set. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	NID1	NID2	NID3	NID4	NID5	NID6	NID7	NID8
Type	I	I	I	I	I	I	I	I

Node ID with Column Cards. This Card 2 format applies to the COLUMN keyword option. Include one card per node in the set. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	NID	A1	A2	A3	A4			
Type	I	F	F	F	F			
Remark		2	2	2	2			

Node ID Range Cards. This Card 2 format applies to the LIST_GENERATE keyword option. Set one pair of BNBEG and BNEND values per block of nodes. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	B1BEG	B1END	B2BEG	B2END	B3BEG	B3END	B4BEG	B4END
Type	I	I	I	I	I	I	I	I

Node ID Range with Increment Cards. This Card 2 format applies to the LIST_GENERATE_INCREMENT keyword option. For each block of nodes add one card to the deck. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	BBEG	BEND	INCR					
Type	I	I	I					

Generalized Node ID Range Cards. This Card 2 format applies to the GENERAL keyword option. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	OPTION	E1	E2	E3	E4	E5	E6	E7
Type	A	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
SID	Set identification. All node sets should have a unique set ID.
DA1	First nodal attribute default value, see remark 1 below.
DA2	Second nodal attribute default value
DA3	Third nodal attribute default value
DA4	Fourth nodal attribute default value
SOLVER	Name of solver using this set (MECH, CESE, etc.)
NID*i*	Node ID *i*
NID	Nodal ID
A1	First nodal attribute, see remark 2 below.
A2	Second nodal attribute
A3	Third nodal attribute
A4	Fourth nodal attribute

VARIABLE	DESCRIPTION
B*n*BEG	First node ID in block *n*.
B*n*END	Last node ID in block *n*. All defined ID's between and including B*n*BEG to B*n*END are added to the set. These sets are generated after all input is read so that gaps in the node numbering are not a problem. B*n*BEG and B*n*END may simply be limits on the ID's and not nodal ID's.
BBEG	First node ID in block.
BEND	Last node ID in block.
INCR	Node ID increment. Node IDs BBEG, BBEG + INCR, BBEG + 2 × INCR, and so on through BEND are added to the set.
OPTION	Option for GENERAL. See table below.
E1, …, E7	Specified entity. Each card must have the option specified. See table below.

The General Option:

The "OPTION" column in the table below enumerates the allowed values for the "OPTION" variable in Card 2 for the GENERAL option. Likewise, the variables E1, …, E7 refer to the GENERAL option Card 2.

Each of the following operations accept up to 7 arguments, but they may take fewer. Values of "En" left unspecified are ignored.

OPTION	DESCRIPTION
ALL	All nodes will be included in the set.
NODE	Nodes E1, E2, E3, … will be included.
DNODE	Nodes E1, E2, E3, … previously added will be excluded.
PART	Nodes of parts E1, E2, E3, … will be included.
DPART	Nodes of parts E1, E2, E3, … previously added will be excluded.
BOX	Nodes inside boxes E1, E2, E3, … will be included. (see *DEFINE_BOX)

OPTION	DESCRIPTION
DBOX	Nodes inside boxes E1, E2, E3, … previously added will be excluded.
VOL	Nodes inside contact volumes E1, E2, E3, … will be included. (see *DEFINE_CONTACT_VOLUME)
DVOL	Nodes inside contact volumes E1, E2, E3, … previously added will be excluded.
SET_*XXXX*	Include nodal points of element sets defined by SET_*XXXX*_LIST, where *XXXX* could be SHELL, SOLID, BEAM, TSHELL and SPRING
SALECPT	Nodes inside a box in Structured ALE mesh. E1 here is the S-ALE mesh ID (MSHID). E2, E3, E4, E5, E6, E7 correspond to XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX. They are the minimum and the maximum nodal indices along each direction in S-ALE mesh. This option is only to be used for Structured ALE mesh and should not be used in a mixed manner with other "_GENERAL" options. Please refer to *ALE_STRUCTURED_MESH_CONTROL_-POINTS and *ALE_STRUCTURED_MESH_CONTROL for more details.
SALEFAC	Nodes that are on the face of a Structured ALE mesh. E1 gives the S-ALE mesh ID (MSHID). E2, E3, E4, E5, E6, E7 correspond to *-x*, *+x*, *-y*, *+y*, *-z*, *+z* faces. Assigning 1, for instance, to these 6 values would include all the surface segments at these faces in the segment set. This option is only to be used for Structured ALE mesh and should not be used in a mixed manner with other "_GENERAL" options. Please refer to *ALE_STRUCTURED_MESH_CONTROL_-POINTS and *ALE_STRUCTURED_MESH_CONTROL for more details.

Remarks:

1. Nodal attributes can be assigned to pass data to other keywords. For example, for contact option, *CONTACT_TIEBREAK_NODES_TO_SURFACE the attributes are:

DA1 = NFLF ⇒ Normal failure force,

DA2 = NSFL ⇒ Shear failure force,

DA3 = NNEN ⇒ Exponent for normal force,

DA4 = NMES ⇒ Exponent for shear force.

2. The default nodal attributes can be overridden on these cards; otherwise, A1 = DA1, etc.

3. This field is used by a non-mechanics solver to create a set defined on that solver's mesh. By default, the set refers to the mechanics mesh.

4. The option *SET_NODE_LIST_SMOOTH is used for localized tooling surface smoothing, and is used in conjunction with keywords *INTERFACE_COMPENSATION_NEW_LOCAL_SMOOTH, *INCLUDE_COMPENSATION_-ORIGINAL_RIGID_TOOL, and *INCLUDE_COMPENSATION_NEW_RIGID_-TOOL. This option is available in R6 Revision 73850 and later releases

SET_NODE_ADD_*{OPTION}

Available options include:

<BLANK>

ADVANCED

Purpose: Define a node set by combining node sets or for the ADVANCED option by combining, NODE, SHELL, SOLID, BEAM, SEGMENT, DISCRETE and THICK SHELL sets.

Card 1	1	2	3	4	5	6	7	8
Variable	NSID	DA1	DA2	DA3	DA4	SOLVER		
Type	I	F	F	F	F	A		
Default	none	none	none	none	none	MECH		
Remark						1		

Node Set Cards. This Card 2 format is used when the keyword option is left unset (<BLANK>). Each card can be used to specify up to 8 node set IDs. Include as many cards of this kind as necessary. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	NSID1	NSID2	NSID3	NSID4	NSID5	NSID6	NSID7	NSID8
Type	I	I	I	I	I	I	I	I

Node Set Advanced Cards. This Card 2 format is used when the keyword option is set to ADVANCED. Each card can be used to specify up to 4 set IDs (node sets, beam sets, etc…). Include as many cards of this kind as necessary. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	SID1	TYPE1	SID2	TYPE2	SID3	TYPE3	SID4	TYPE4
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
NSID	Set ID of new node set. All node sets should have a unique set ID.
DA1	First nodal attribute default value, see remark 1 below.
DA2	Second nodal attribute default value
DA3	Third nodal attribute default value
DA4	Fourth nodal attribute default value
SOLVER	Name of solver using this set (MECH, CESE, etc.)
NSID[N]	The Nth node set ID on Card 2 in LIST format.
SID[N]	The Nth set ID on Card 2 in ADVANCED format.
TYPE[N]	Type set for SID[N]: EQ.1: Node set EQ.2: Shell set EQ.3: Beam set EQ.4: Solid set EQ.5: Segment set EQ.6: Discrete set EQ.7: Thick shell set

Remarks:

1. This field is used by a non-mechanics solver to create a set defined on that solver's mesh. By default, the set refers to the mechanics mesh.

*SET_NODE_INTERSECT

Purpose: Define a node set as the intersection, ∩, of a series of node sets. The new node set, NSID, contains all common elements of all node sets listed on all cards in format 2.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	DA1	DA2	DA3	DA4	SOLVER		
Type	I	F	F	F	F	A		
Default	none	none	none	none	none	MECH		
Remark						1		

Node Set Cards. For each SID in the intersection specify one field. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	NSID1	NSID2	NSID3	NSID4	NSID5	NSID6	NSID7	NSID8
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
SID	Set ID of new node set. All node sets should have a unique set ID.
DAi	Nodal attribute of the i'th node.
SOLVER	Name of solver using this set (MECH, CESE, etc.)
NSIDn	The *n*th node set ID.

Remarks:

1. This field is used by a non-mechanics solver to create a set defined on that solver's mesh. By default, the set refers to the mechanics mesh.

SET_PART_*{OPTION1}_{OPTION2}

For *OPTION1* available options are:

<BLANK>

LIST

COLUMN

LIST_GENERATE

LIST_GENERATE_INCREMENT

For *OPTION2* the available option is:

COLLECT

The LIST_GENERATE and LIST_GENERATE_INCREMENT options will generate block(s) of part IDs between a starting ID and an ending ID. An arbitrary number of blocks can be specified to define the part set.

Purpose: Define a set of parts with optional attributes. For the column option, see *AIRBAG or *CONSTRAINED_RIGID_BODY_STOPPERS.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	DA1	DA2	DA3	DA4	SOLVER		
Type	I	F	F	F	F	A		
Default	none	0.				MECH		
Remark		1	1	1	1	3		

Part ID Cards. This Card 2 format applies to the LIST keyword option. Additionally, it applies to the case of an unset (<BLANK>) keyword option. Set one value per part in the set. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	PID1	PID2	PID3	PID4	PID5	PID6	PID7	PID8
Type	I	I	I	I	I	I	I	I

Part ID with Column Cards. This Card 2 format applies to the COLUMN keyword option. Include one card per part in the set. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	PID	A1	A2	A3	A4			
Type	I	F	F	F	F			
Remark		2	2	2	2			

Part ID Range Cards. This Card 2 format applies to the GENERATE keyword option. Set one pair of BNBEG and BNEND values per block of part IDs. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	B1BEG	B1END	B2BEG	B2END	B3BEG	B3END	B4BEG	B4END
Type	I	I	I	I	I	I	I	I

Part ID Range with Increment Cards. This Card 2 format applies to the LIST_GENERATE_INCREMENT keyword option. For each block of parts add one card to the deck. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	BBEG	BEND	INCR					
Type	I	I	I					

VARIABLE	DESCRIPTION
SID	Set ID. All part sets should have a unique set ID.
DA1	First attribute default value, see remark 1 below.
DA2	Second attribute default value
DA3	Third attribute default value
DA4	Fourth attribute default value
SOLVER	Name of solver using this set (MECH, CESE, etc.)
PID	Part ID
PID1	First part ID
PID2	Second part ID
⋮	⋮
A1	First part attribute, see remark 2 below.
A2	Second part attribute
A3	Third part attribute
A4	Fourth part attribute
B[N]BEG	First part ID in block N.
B[N]END	Last part ID in block N. All defined ID's between and including B[N]BEG to B[N]END are added to the set. These sets are generated after all input is read so that gaps in the part numbering are not a problem. B[N]BEG and B[N]END may simply be limits on the ID's and not part ID's.

VARIABLE	DESCRIPTION
BBEG	First part ID in block.
BEND	Last part ID in block.
INCR	Part ID increment. Part IDs BBEG, BBEG+INCR, BBEG + 2 × INCR, and so on through BEND are added to the set.

Remarks:

1. Part attributes can be assigned for some input types. For example, for airbags a time delay, DA1 = T1, can be defined before pressure begins to act along with a time delay, DA2 = T2, before full pressure is applied, (default T2 = T1), and for the constraint option, *CONSTRAINED_RIGID_BODY_STOPPERS one attribute can be defined: DA1, the closure distance which activates the stopper constraint.

2. The default part attributes can be overridden on the part cards; otherwise, A1 = DA1, etc.

3. This field is used by a non-mechanics solver to create a set defined on that solver’s mesh. By default, the set refers to the mechanics mesh.

*SET_PART_ADD

Purpose: Define a part set by combining part sets.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	DA1	DA2	DA3	DA4	SOLVER		
Type	I	F	F	F	F	A		
Default	none					MECH		
Remark		1,2	1,2	1,2	1,2	3		

Part Set Cards. Each card can be used to specify up to 8 part set IDs. Include as many cards of this kind as necessary. This input ends at the next keyword ("*") card.

Card 2…	1	2	3	4	5	6	7	8
Variable	PSID1	PSID2	PSID3	PSID4	PSID5	PSID6	PSID7	PSID8
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
SID	Set ID. All part sets should have a unique set ID.
DA1	First attribute default value, see Remarks 1 and 2 below.
DA2	Second attribute default value
DA3	Third attribute default value
DA4	Fourth attribute default value
SOLVER	Name of solver using this set (MECH, CESE, etc.)

VARIABLE	DESCRIPTION
PSID[N]	The Nth part set ID GT.0: PSIDn is added to SID, LT.0: all part sets with ID between PSID(i-1) and |PSIDi|, including PSID(i-1) and |PSIDi|, will be added to SID. PSID(i-1) has to be > 0 and has a magnitude smaller or equal to |PSIDi | when PSIDi < 0.

Remarks:

1. Part attributes can be assigned for some input types. For example, for airbags a time delay, DA1 = T1, can be defined before pressure begins to act along with a time delay, DA2 = T2, before full pressure is applied, (default T2 = T1), and for the constraint option, *CONSTRAINED_RIGID_BODY_STOPPERS one attribute can be defined: DA1, the closure distance which activates the stopper constraint.

2. The default values for the part attributes are given in the contributing *SET_-PART_{*OPTION*} commands. Nonzero values of DA1, DA2, DA3, or DA4 in *SET_PART_ADD will override the respective default values.

3. This field is used by a non-mechanics solver to create a set defined on that solver's mesh. By default, the set refers to the mechanics mesh.

*SET_SEGMENT_{OPTION1}_{OPTION2}

For *OPTION1* the available options are:

<BLANK>

GENERAL

For *OPTION2* the available option is

COLLECT

Purpose: Define set of segments with optional identical or unique attributes. For three-dimensional geometries, a segment can be triangular or quadrilateral. For two-dimensional geometries, a segment is a line defined by two nodes and the GENERAL option does not apply.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	DA1	DA2	DA3	DA4	SOLVER		
Type	I	F	F	F	F	A		
Default	none	0.	0.	0.	0.	MECH		
Remarks		1	1	1	1	4		

Segment Cards. For each segment in the set include on card of this format. Set N3 = N4 for triangular segments. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	N1	N2	N3	N4	A1	A2	A3	A4
Type	I	I	I	I	F	F	F	F
Remarks				2	3	3	3	3

Generalized Part ID Range Cards. This Card 2 format applies to the GENERAL keyword option. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	OPTION	E1	E2	E3	E4	E5	E6	E7
Type	A	I	I	I	I or F	I or F	I or F	I or F

VARIABLE	DESCRIPTION
SID	Set ID. All segment sets should have a unique set ID.
DA1	First segment attribute default value, see remark 1 below.
DA2	Second segment attribute default value
DA3	Third segment attribute default value
DA4	Fourth segment attribute default value
SOLVER	Name of solver using this set (MECH, CESE, etc.)
N1	Nodal point n_1
N2	Nodal point n_2
N3	Nodal point n_3
N4	Nodal point n_4, see Remark 2 below.
A1	First segment attribute, see Remark 3 below.
A2	Second segment attribute
A3	Third segment attribute
A4	Fourth segment attribute
NFLS	Normal failure stress
SFLS	Shear failure stress. Failure criterion:
OPTION	Option for GENERAL. See table below.

VARIABLE	DESCRIPTION
E1, …, E7	Specified entity. Each card must have an option specified. See table below.

The General Option:

The “OPTION” column in the table below enumerates the allowed values for the “OPTION” variable in Card 2 for the GENERAL option. Likewise, the variables E1, …, E7 refer to the GENERAL option Card 2.

Each of the following operations accept up to 7 arguments, but they may take fewer. Values of “En” left unspecified are ignored.

OPTION	DESCRIPTION
ALL	All exterior segments will be included in the set.
BOX	Generate segments inside boxes having IDs E1, E2, and E3 with attributes having values E4, E5, E6, and E7. For shell elements one segment per shell is generated. For solid elements only those segments wrapping the solid part and pointing outward from the part will be generated.
BOX_SHELL	Generate segments inside boxes having IDs E1, E2, and E3 with attributes having values E4, E5, E6, and E7. The segments are only generated for shell elements. One segment per shell is generated.
BOX_SLDIO	Generate segments inside boxes having IDs E1, E2, and E3 with attributes having values E4, E5, E6, and E7. Both exterior segments and inter-element segments are generated.
BOX_SOLID	Generate segments inside boxes having IDs E1, E2, and E3 with attributes having values E4, E5, E6, and E7. The segments are only generated for exterior solid elements
PART	Generate segments of parts E1, E2, and E3 with attributes E4, E5, E6, and E7. For shell elements one segment per shell is generated. For solid elements only those segments wrapping the solid part and pointing outward from the part will be generated. PART could refer to beam parts when defining 2D segments for traction application.

OPTION DESCRIPTION

PART_IO
Generate segments from parts E1, E2, E3 with attributes E4, E5, E6, and E7. Same as the PART option above except that inter-element segments inside parts will be generated as well. This option is sometimes useful for single surface contact of solid elements to prevent negative volumes.

PSLDFi
Generate segments from the i'th face of solid parts E1, E2, E3 with attributes E4, E5, E6, and E7. See table below for face definition.

SEG
Create segment with node IDs E1, E2, E3, and E4.

VOL
Generate segments inside contact volume IDs E1, E2, and E3 with attributes having values E4, E5, E6, and E7. See BOX option for other details.

VOL_SHELL
Generate segments for shells inside contact volume IDs E1, E2, and E3 with attributes having values E4, E5, E6, and E7

VOL_SLDIO
Generate segments for solid elements inside contact volume IDs E1, E2, and E3 with attributes E4, E5, E6, and E7. See BOX_SLDIO for other details.

VOL_SOLID
Generate segments for solid elements inside contact volume IDs E1, E2, and E3 with attributes E4, E5, E6, and E7. See BOX_SOLID for other details.

SET_SHELL
Generate segments for shell elements in SET_SHELL_LIST with IDs E1, E2, and E3 with attributes E4, E5, E6, and E7.

SET_SOLID
Generate segments for solid elements in SET_SOLID_LIST with IDs E1, E2, and E3 with attributes E4, E5, E6, and E7.

SET_SLDIO
Generate segments for solid elements in SET_SOLID_LIST with IDs E1, E2, and E3 with attributes E4, E5, E6, and E7. Both exterior & interior segments are generated.

SET_SLDFi
Generate segments from the ith face of solid elements in SET_SOLID_LIST with IDs E1, E2, and E3 with attributes E4, E5, E6, and E7. See table below for face definition.

SET_TSHELL
Generate segments for thick shell elements in SET_TSHELL_LIST with IDs of E1, E2, and E3 with attributes E4, E5, E6, and E7. Only exterior segments are generated.

OPTION **DESCRIPTION**

SET_TSHIO Generate segments for thick shell elements in SET_TSHELL_LIST with IDs of E1, E2, and E3 with attributes E5, E5, E6, and E7. Both exterior & interior segments are generated.

DBOX Segments inside boxes with IDs E1, …, E7 will be excluded.

DBOX_SHELL Shell related segments inside boxes of IDs E1, …, E7 will be excluded.

DBOX_SOLID Solid related segments inside boxes of IDs E1, …, E7 will be excluded.

DPART Segments of parts with IDs E1, …, E7 will be excluded.

DSEG Segment with node IDs E1, E2, E3, and E4 will be deleted.

DVOL Segments inside contact volumes having IDs E1, …, E7 will be excluded.

DVOL_SHELL Shell related segments inside contact volumes having IDs E1, …, E7 will be excluded.

DVOL_SOLID Solid related segments inside contact volumes having IDs E1, …, E7 will be excluded.

SALECPT Segments inside a box in Structured ALE mesh. E1 here is the S-ALE mesh ID (MSHID). E2, E3, E4, E5, E6, E7 correspond to XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX. They are the minimum and the maximum nodal indices along each direction in S-ALE mesh. This option is only to be used for Structured ALE mesh and should not be used in a mixed manner with other "_GENERAL" options.

Please refer to *ALE_STRUCTURED_MESH_CONTROL_POINTS and *ALE_STRUCTURED_MESH_CONTROL for more details.

OPTION	DESCRIPTION
SALEFAC	Segments on the face of Structured ALE mesh. E1 here is the S-ALE mesh ID (MSHID). E2, E3, E4, E5, E6, E7 correspond to -X, +X, -Y, +Y, -Z, +Z faces. Assigning 1 to these 6 values would include all the surface segments at these faces in the segment set. This option is only to be used for Structured ALE mesh and should not be used in a mixed manner with other "_GENERAL" options. Please refer to *ALE_STRUCTURED_MESH_CONTROL_POINTS and *ALE_STRUCTURED_MESH_CONTROL for more details.

Remarks:

1. Segment attributes can be assigned for some input types. For example, for the contact options.

   The attributes for the SLAVE surface are:

   DA1 (NFLS) = Normal failure stress, *CONTACT_TIEBREAK_SURFACE contact only,

   DA2 (SFLS) = Shear failure stress, *CONTACT_TIEBREAK_SURFACE contact only,

   DA3 (FSF) = Coulomb friction scale factor,

   DA4 (VSF) = Viscous friction scale factor,

   and the attributes for the MASTER surface are:

   DA3 (FSF) = Coulomb friction scale factor,

   DA4 (VSF) = Viscous friction scale factor.

   For airbags, see *AIRBAG, a time delay, DA1 = T1, can be defined before pressure begins to act on a segment along with a time delay, DA2 = T2, before full pressure is applied to the segment, (default T2 = T1), and for the constraint option,

2. To define a triangular segment make N4 equal to N3.

3. The default segment attributes can be overridden on these cards, otherwise, A1 = DA1, A2 = DA2, etc.

4. This field is used by a non-mechanics solver to create a set defined on that solver's mesh. By default, the set refers to the mechanics mesh.

FACE	Hexahedron	Pentahedron	Tetrahedron
1	N1, N5, N8, N4	N1, N2, N5	N1, N2, N4
2	N2, N3, N7, N6	N4, N6, N3	N2, N3, N4
3	N1, N2, N6, N5	N1, N4, N3, N2	N1, N3, N2
4	N4, N8, N7, N3	N2, N3, N6, N5	N1, N4, N3
5	N1, N4, N3, N2	N1, N5, N6, N4	
6	N5, N6, N7, N8		

Table 4.1 Face definition of solid elements

*SET_SEGMENT_ADD

Purpose: Define a segment set by combining segment sets.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	SOLVER						
Type	I	A						
Default	none	MECH						
Remark		1						

Segment Set Cards. Each card can be used to specify up to 8 segment set IDs. Include as many cards of this kind as necessary. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	SSID1	SSID2	SSID3	SSID4	SSID5	SSID6	SSID7	SSID8
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
SID	Set ID of new segment set. All segment sets should have a unique set ID.
SOLVER	Name of solver using this set (MECH, CESE, etc.)
SSID[N]	The Nth segment set ID on card2.

Remarks:

1. This field is used by a non-mechanics solver to create a set defined on that solver's mesh. By default, the set refers to the mechanics mesh.

*SET_SEGMENT_INTERSECT

Purpose: Define a segment set as the intersection, ∩, of a series of segment sets. The new segment set, SID, contains all segments common to the sets listed on all of the cards in format 2.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	SOLVER						
Type	I	A						
Default	none	MECH						
Remark		1						

Segment Set Cards. For each SID in the intersection specify one field. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	SSID1	SSID2	SSID3	SSID4	SSID5	SSID6	SSID7	SSID8
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
SID	Set ID of new segment set. All segment sets should have a unique set ID.
SOLVER	Name of solver using this set (MECH, CESE, etc.)
SSID[N]	The Nth segment set ID

Remarks:

1. This field is used by a non-mechanics solver to create a set defined on that solver's mesh. By default, the set refers to the mechanics mesh.

*SET_2D_SEGMENT_*{OPTION1}*_*{OPTION2}*

For *OPTION1* the available options are:

<BLANK>

SET

For *OPTION2* the available option is:

COLLECT

Purpose: Define a set of boundary line segments in two-dimensional axisymmetric, plane stress, and plane strain geometries with optional attributes. This command does not apply to beam formulations 7 and 8. It is sometimes convenient for two-dimensional parts which are subject to adaptivity because the segments in the set are updated as the geometry adapts.

Card Sets. For each set include a pair of cards 1 and 2. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	DA1	DA2	DA3	DA4			
Type	I	F	F	F	F			
Default	none	0.	0.	0.	0.			
Remarks		1	1	1	1			

Card 2	1	2	3	4	5	6	7	8
Variable	PID/PSID							
Type	I							
Remarks	2							

VARIABLE	DESCRIPTION
SID	Set ID. All segment sets should have a unique set ID.
DA1	First segment attribute default value, see remark 1 below.
DA2	Second segment attribute default value
DA3	Third segment attribute default value
DA4	Fourth segment attribute default value
PID/PSID	Part ID or part set ID if SET option is specified.

Remarks:

1. The boundary along $r = 0$ isn't included in axisymmetric problems.

2. The common boundary between parts in the part set PSID is not included in the boundary segments.

*SET_SHELL_*{OPTION1}*_*{OPTION2}*

For *OPTION1* the available options are:

<BLANK>

LIST

COLUMN

LIST_GENERATE

LIST_GENERATE_INCREMENT

GENERAL

For *OPTION2* the available option is:

COLLECT

The LIST_GENERATE and LIST_GENERATE_INCREMENT options will generate block(s) of shell element IDs between a starting ID and an ending ID. An arbitrary number of blocks can be specified to define the shell element set.

Purpose: Define a set of shell elements with optional identical or unique attributes.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	DA1	DA2	DA3	DA4			
Type	I	F	F	F	F			
Default	none	0.	0.	0.	0.			
Remarks		1	1	1	1			

Shell Element ID Cards. This Card 2 format applies to LIST keyword option. Additionally, it applies to the case of an unset (<BLANK>) keyword option. Set one value per element in the set. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	EID1	EID2	EID3	EID4	EID5	EID6	EID7	EID8
Type	I	I	I	I	I	I	I	I
Remarks	2	2	2	2	2	2	2	2

Shell Element ID with Column Cards. This Card 2 format applies to the COLUMN keyword option. Include one card per shell element in the set. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	EID	A1	A2	A3	A4			
Type	I	F	F	F	F			
Remarks		3	3	3	3			

Shell Element ID Range Cards. This Card 2 format applies to the GENERATE keyword option. Set one pair of BNBEG and BNEND values per block of shell element IDs. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	B1BEG	B1END	B2BEG	B2END	B3BEG	B3END	B4BEG	B4END
Type	I	I	I	I	I	I	I	I

Shell Element ID Range with Increment Cards. This Card 2 format applies to the LIST_GENERATE_INCREMENT keyword option. For each block of shell elements add one card to the deck. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	BBEG	BEND	INCR					
Type	I	I	I					

Generalized Shell Element ID Range Cards. This Card 2 format applies to the GENERAL keyword option. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	OPTION	E1	E2	E3	E4	E5	E6	E7
Type	A	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
SID	Set ID. All shell sets should have a unique set ID.
DA1	First attribute default value, see remark 1.
DA2	Second attribute default value
DA3	Third attribute default value
DA4	Fourth attribute default value
EID1	First shell element ID, see remark 2.
EID2	Second shell element ID
⋮	⋮
EID	Element ID
A1	First attribute
A2	Second attribute
A3	Third attribute

VARIABLE	DESCRIPTION
A4	Fourth attribute
B*n*BEG	First shell ID in shell block *n*.
B*n*END	Last shell ID in block *n*. All defined ID’s between and including B*n*BEG to B*n*END are added to the set. These sets are generated after all input is read so that gaps in the element numbering are not a problem. B*n*BEG and B*n*END may simply be limits on the ID’s and not element IDs.
BBEG	First shell element ID in block.
BEND	Last shell element ID in block.
INCR	Shell element ID increment. Shell element IDs BBEG, BBEG + INCR, BBEG + 2 × INCR, and so on through BEND are added to the set.
OPTION	Option for GENERAL. See table below.
E1, …, E7	Specified entity. Each card must have the option specified. See table below.

The General Option:

The “OPTION” column in the table below enumerates the allowed values for the “OPTION” variable in Card 2 for the GENERAL option. Likewise, the variables E1, …, E7 refer to the GENERAL option Card 2.

Each of the following operations accept up to 7 arguments, but they may take fewer. Values of “En” left unspecified are ignored.

OPTION	DESCRIPTION
ALL	All shell elements will be included in the set.
ELEM	Shell elements E1, E2, E3, … will be included.
DELEM	Shell elements E1, E2, E3, … previously added will be excluded.
PART	Shell elements of parts E1, E2, E3, … will be included.
DPART	Shell elements of parts E1, E2, E3, … previously added will be excluded.

OPTION	DESCRIPTION
BOX	Shell elements inside boxes E1, E2, E3, … will be included. (see *DEFINE_BOX)
DBOX	Shell elements inside boxes E1, E2, E3, … previously added will be excluded.

Remarks:

1. Shell attributes can be assigned for some input types.

   For example, for contact options, the attributes for the SLAVE surface are:

   DA1 (NFLS) = Normal failure stress, *CONTACT_TIEBREAK_SURFACE contact only,

   DA2 (SFLS) = Shear failure stress, *CONTACT_TIEBREAK_SURFACE contact only,

   DA3 (FSF) = Coulomb friction scale factor,

   DA4 (VSF) = Viscous friction scale factor,

   and the attributes for the MASTER surface are:

   DA1 (FSF) = Coulomb friction scale factor,

   DA2 (VSF) = Viscous friction scale factor.

2. The default attributes are taken.

3. The default shell attributes can be overridden on these cards; otherwise, A1 = DA1, etc.

*SET_SHELL_ADD

Purpose: Define a shell set by combining shell sets.

Card 1	1	2	3	4	5	6	7	8
Variable	SID							
Type	I							
Default	none							

Shell Element Set Cards. Each card can be used to specify up to 8 shell element set IDs. Include as many cards as necessary. This input ends at the next keyword (“*”) card.

Card 2	1	2	3	4	5	6	7	8
Variable	SSID1	SSID2	SSID3	SSID4	SSID5	SSID6	SSID7	SSID8
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
SID	Set ID of new shell set. All shell sets should have a unique set ID.
SSID[N]	The Nth shell set ID on card2

*SET_SHELL_INTERSECT

Purpose: Define a shell set as the intersection, ∩, of a series of shell sets. The new shell set, SID, contains all shells common to all sets on the cards of format 2.

Card 1	1	2	3	4	5	6	7	8
Variable	SID							
Type	I							
Default	none							

Shell Element Set Cards. For each shell element SID in the intersection input one field. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	SSID1	SSID2	SSID3	SSID4	SSID5	SSID6	SSID7	SSID8
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
SID	Set ID of new shell set. All shell sets should have a unique set ID.
SSID[N]	The Nth shell set ID

SET_SOLID_*{OPTION1}_{OPTION2}

For *OPTION1* the available options are:

<BLANK>

GENERATE

GENERATE_INCREMENT

GENERAL

For *OPTION2* the available option is:

COLLECT

The GENERATE and GENERATE_INCREMENT options will generate block(s) of solid element
IDs between a starting ID and an ending ID. An arbitrary number of blocks can be specified to define the solid element set.

Purpose: Define a set of solid elements.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	SOLVER						
Type	I	A						
Default	none	MECH						
Remark		1						

Solid Element ID Cards. This Card 2 format applies to the case of an unset (<BLANK>) keyword option. Set one value per solid element in the set. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	K1	K2	K3	K4	K5	K6	K7	K8
Type	I	I	I	I	I	I	I	I

Solid Element ID Range Cards. This Card 2 format applies to the GENERATE keyword option. Set one pair of BNBEG and BNEND values per block of solid elements. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	B1BEG	B1END	B2BEG	B2END	B3BEG	B3END	B4BEG	B4END
Type	I	I	I	I	I	I	I	I

Solid Element ID Range with Increment Cards. This Card 2 format applies to the GENERATE_INCREMENT keyword option. For each block of solid elements add one card to the deck. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	BBEG	BEND	INCR					
Type	I	I	I					

Generalized Solid Element ID Range Cards. This Card 2 format applies to the GENERAL keyword option. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	OPTION	E1	E2	E3	E4	E5	E6	E7
Type	A	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
SID	Set ID. All solid sets should have a unique set ID.
SOLVER	Name of solver using this set (MECH, CESE, etc.)
K1	First element ID
K2	Second element ID
⋮	⋮
K8	Eighth element ID
B[N]BEG	First solid element ID in block N.
B[N]END	Last solid element ID in block N. All defined ID's between and including B[N]BEG to B[N]END are added to the set. These sets are generated after all input is read so that gaps in the element numbering are not a problem. B[N]BEG and B[N]END may simply be limits on the ID's and not element IDs.
BBEG	First solid element ID in block.
BEND	Last solid element ID in block.
INCR	Solid ID increment. Solid IDs BBEG, BBEG + INCR, BBEG + 2 × INCR, and so on through BEND are added to the set.
OPTION	Option for GENERAL. See table below.
E1, ..., E7	Specified entity. Each card must have the option specified. See table below.

The General Option:

The "OPTION" column in the table below enumerates the allowed values for the "OPTION" variable in Card 2 for the GENERAL option. Likewise, the variables E1, …, E7 refer to the GENERAL option Card 2.

Each of the following operations accept up to 7 arguments, but they may take fewer. Values of "En" left unspecified are ignored.

OPTION	DESCRIPTION
ALL	All solid elements will be included in the set.
ELEM	Elements E1, E2, E3, ... will be included.
DELEM	Elements E1, E2, E3, ... previously added will be excluded.
PART	Elements of parts E1, E2, E3, ... will be included.
DPART	Elements of parts E1, E2, E3, ... previously added will be excluded.
BOX	Elements inside boxes E1, E2, E3, ... will be included. (see *DEFINE_BOX)
DBOX	Elements inside boxes E1, E2, E3, ... previously added will be excluded.
SALECPT	Elements inside a box in Structured ALE mesh. E1 here is the S-ALE mesh ID (MSHID). E2, E3, E4, E5, E6, E7 correspond to XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX. They are the minimum and the maximum nodal indices along each direction in S-ALE mesh. This option is only to be used for Structured ALE mesh and should not be used in a mixed manner with other "_GENERAL" options. Please refer to *ALE_STRUCTURED_MESH_CONTROL_POINTS and *ALE_STRUCTURED_MESH_CONTROL for more details.

OPTION	DESCRIPTION
SALEFAC	Elements on the face of Structured ALE mesh. E1 here is the S-ALE mesh ID (MSHID). E2, E3, E4, E5, E6, E7 correspond to -X, +X, -Y, +Y, -Z, +Z faces. Assigning 1 to these 6 values would include all the boundary elements at these faces in the segment set. This option is only to be used for Structured ALE mesh and should not be used in a mixed manner with other "_GENERAL" options. Please refer to *ALE_STRUCTURED_MESH_CONTROL_-POINTS and *ALE_STRUCTURED_MESH_CONTROL for more details.

Remarks:

1. This field is used by a non-mechanics solver to create a set defined on that solver's mesh. By default, the set refers to the mechanics mesh.

*SET_SOLID_ADD

Purpose: Define a solid set by combining solid sets.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	SOLVER						
Type	I	A						
Default	none	MECH						
Remark		1						

Node Set Cards. Each card can be used to specify up to 8 solid set IDs. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	SSID1	SSID2	SSID3	SSID4	SSID5	SSID6	SSID7	SSID8
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
SID	Set ID of new solid set. All solid sets should have a unique set ID.
SOLVER	Name of solver using this set (MECH, CESE, etc.)
SSID[N]	The Nth solid set ID.

Remarks:

1. This field is used by a non-mechanics solver to create a set defined on that solver's mesh. By default, the set refers to the mechanics mesh.

*SET_SOLID_INTERSECT

Purpose: Define a solid set as the intersection, ∩, of a series of solid sets. The new solid set, SID, contains all common elements of all solid sets SSID*n*.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	SOLVER						
Type	I	A						
Default	none	MECH						
Remark		1						

Solid Element Set Cards. For each solid element SID in the intersection input one field. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 2…	1	2	3	4	5	6	7	8
Variable	SSID1	SSID2	SSID3	SSID4	SSID5	SSID6	SSID7	SSID8
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
SID	Set ID of new solid set. All solid sets should have a unique set ID.
SOLVER	Name of solver using this set (MECH, CESE, etc.)
SSIDN	The Nth solid set ID on card2

Remarks:

1. This field is used by a non-mechanics solver to create a set defined on that solver's mesh. By default, the set refers to the mechanics mesh.

*SET_TSHELL_{OPTION1}_{OPTION2}

For *OPTION1* the available options are:

<BLANK>

GENERATE

GENERAL

For *OPTION2* the available option is:

COLLECT

The option GENERATE will generate a block of thick shell element ID's between a starting ID and an ending ID. An arbitrary number of blocks can be specified to define the set.

Purpose: Define a set of thick shell elements.

Card 1	1	2	3	4	5	6	7	8
Variable	SID							
Type	I							
Default	none							

Thick Shell Element ID Cards. This Card 2 format applies to the case of an unset (<BLANK>) keyword option. Set one value per thick shell element in the set. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	K1	K2	K3	K4	K5	K6	K7	K8
Type	I	I	I	I	I	I	I	I

Thick Shell Element ID Range Cards. This Card 2 format applies to the GENERATE keyword option. Set one pair of BNBEG and BNEND values per block of thick shell elements. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2…	1	2	3	4	5	6	7	8
Variable	B1BEG	B1END	B2BEG	B2END	B3BEG	B3END	B4BEG	B4END
Type	I	I	I	I	I	I	I	I

Generalized Thick Shell Element ID Range Cards. This Card 2 format applies to the GENERAL keyword option. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 2…	1	2	3	4	5	6	7	8
Variable	OPTION	E1	E2	E3	E4	E5	E6	E7
Type	A	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
SID	Set ID. All tshell sets should have a unique set ID.
K1	First thick shell element ID
K2	Second thick shell element ID
⋮	⋮
K8	Eighth thick shell element ID
B[N]BEG	First thick shell element ID in block N.
B[N]END	Last thick shell element ID in block N. All defined ID's between and including B[N]BEG to B[N]END are added to the set. These sets are generated after all input is read so that gaps in the element numbering are not a problem. B[N]BEG and B[N]END may simply be limits on the ID's and not element IDs.
OPTION	Option for GENERAL. See table below.

VARIABLE	DESCRIPTION
E1, ..., E7	Specified entity. Each card must have the option specified. See table below.

The General Option:

The "OPTION" column in the table below enumerates the allowed values for the "OPTION" variable in Card 2 for the GENERAL option. Likewise, the variables E1, …, E7 refer to the GENERAL option Card 2.

Each of the following operations accept up to 7 arguments, but they may take fewer. Values of "En" left unspecified are ignored.

OPTION	DESCRIPTION
ALL	All thick shell elements will be included in the set.
ELEM	Elements E1, E2, E3, ... will be included.
DELEM	Elements E1, E2, E3, ... previously added will be excluded.
PART	Elements of parts E1, E2, E3, ... will be included.
DPART	Elements of parts E1, E2, E3, ... previously added will be excluded.
BOX	Elements inside boxes E1, E2, E3, ... will be included. (see *DEFINE_BOX)
DBOX	Elements inside boxes E1, E2, E3, ... previously added will be excluded.

*TERMINATION

The keyword *TERMINATION provides an alternative way of stopping the calculation before the termination time is reached. The termination time is specified on the *CONTROL_TERMINATION input and will terminate the calculation whether or not the options available in this section are active. Different types of termination may be defined:

*TERMINATION_BODY

Purpose: Terminate calculation based on rigid body displacements. For *TERMINATION_BODY the analysis terminates when the center of mass displacement of the rigid body specified reaches either the maximum or minimum value (stops 1, 2 or 3) or the displacement magnitude of the center of mass is exceeded (stop 4). If more than one condition is input, the analysis stops when any of the conditions is satisfied. Termination by other means than *TERMINATION input is controlled by the *CONTROL_TERMINATION control card. Note that this type of termination is not active during dynamic relaxation.

Part Cards. Add one card for each part having termination criterion. Include as many cards as necessary. This input terminates at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	STOP	MAXC	MINC				
Type	I	I	F	F				
Default	none	none	-	-				

VARIABLE	DESCRIPTION
PID	Part ID of rigid body, see *PART_*OPTION*.
STOP	Stop criterion: EQ.1: global x direction, EQ.2: global y direction, EQ.3: global z direction, EQ.4: stop if displacement magnitude is exceeded.
MAXC	Maximum (most positive) displacement, options 1, 2, 3 and 4: EQ.0.0: MAXC set to 1.0e21.
MINC	Minimum (most negative) displacement, options 1, 2 and 3 above only: EQ.0.0: MINC set to -1.0e21.

*TERMINATION_CONTACT

Purpose: The analysis terminates when the magnitude of the contact interface resultant force is zero. If more than one contact condition is input, the analysis stops when any of the conditions is satisfied. Termination by other means than *TERMINATION input is controlled by the *CONTROL_TERMINATION control card. Note that this type of termination is not active during dynamic relaxation and does not apply to 2D contact types.

Contact ID Cards. Add one card for contact ID having a termination criterion. Include as many cards as necessary. This input terminates at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	CID	ACTIM	DUR	THRES	DOF			
Type	I	F	F	F	I			
Default	none	none	-	0.0	0			

VARIABLE	DESCRIPTION
CID	Contact ID. The contact ID is defined by the ordering of the contact input unless the TITLE option which allows the CID to be defined is used in the *CONTACT section.
ACTIM	Activation time.
DUR	Time duration of null resultant force prior to termination. This time is tracked only after the activation time is reached and the contact resultant forces are zero. EQ.0.0: Immediate termination after null force is detected.
THRES	Any measured force magnitude below or equal to this specified threshold is taken as a null force. Default = 0.0
DOF	Option to consider only the force magnitude in the x, y, or z global directions corresponding to DOF = 1,2, and 3, respectively.

*TERMINATION_CURVE

Purpose: Terminate the calculation when the load curve value returns to zero. This termination can be used with the contact option *CONTACT_AUTO_MOVE. In this latter option, the load curve is modified to account for the movement of the master surface.

Load Curve Card. For each load curve used as a termination criterion add a card. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	LCID	ATIME						
Type	I	F						
Default	none	Remark 1		-				

VARIABLE	DESCRIPTION
LCID	Load curve ID governing termination.
ATIME	Activation time. After this time the load curve is checked. If zero, see remark 1 below.

Remarks:

1. If ATIME = 0.0, termination will occur after the load curve value becomes nonzero and then returns to zero.

TERMINATION_DELETED_SHELLS_*{OPTION}

Available options include:

<BLANK>

SET

Purpose: Terminate the calculation when the number of deleted shells for a specified part ID exceeds the value defined here. This input has no effect for a part ID that is left undefined. Generally, this option should be used with the NFAIL1 and NFAIL4 parameters that are defined in the *CONTROL_SHELL control information.

When using the SET option, termination will occur when NDS elements are deleted in any one of the parts in the part set PSID.

Part (set) Cards. Include one card for each part having a termination criterion based on shell deletion. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID/PSID	NDS						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
PID / PSID	Part ID or if option SET is active, part set ID.
NDS	Number of elements that must be deleted for the specified part ID's, before an error termination occurs.

TERMINATION_DELETED_SOLIDS_*{OPTION}

Available options include:

<BLANK>

SET

Purpose: Terminate the calculation when the number of deleted solids for a specified part ID exceeds the value defined here. This input has no effect for a part ID that is left undefined.

When using the SET option, termination will occur when NDS elements are deleted in any one of the parts in the part set PSID.

Part (set) Cards. Include one card for each part having a termination criterion based on solid element deletion. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID/PSID	NDS						
Type	I	I						
Default	none	1						

VARIABLE	DESCRIPTION
PID/PSID	Part ID or if option SET is active, part set ID.
NDS	Number of elements that must be deleted for the specified part ID's, before an error termination occurs.

*TERMINATION_NODE

Purpose: Terminate calculation based on nodal point coordinates. The analysis terminates for *TERMINATION_NODE when the current position of the node specified reaches either the maximum or minimum value (stops 1, 2 or 3), or picks up force from any contact surface (stops 4). Termination by other means than *TERMINATION is controlled by the *CONTROL_TERMINATION control card. Note that this type of termination is not active during dynamic relaxation.

Node Cards. Include one card for each node having a termination criterion. Include as many cards as desired. This input terminates at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	NID	STOP	MAXC	MINC				
Type	I	I	F	F				
Default	none	none	-	-				

VARIABLE **DESCRIPTION**

NID Node ID, see *NODE_*OPTION*.

STOP Stop criterion:

EQ.1: global x direction,

EQ.2: global y direction,

EQ.3: global z direction,

EQ.4: stop if node touches contact surface.

MAXC Maximum (most positive) coordinate (options 1, 2 and 3) above only.

MINC Minimum (most negative) coordinate (options 1, 2 and 3) above only.

*TERMINATION_SENSOR

Purpose: Terminates the calculation when the switch condition defined in *SENSOR_-SWITCH is met.

Switch ID Cards. Include one card for each switch controlling termination. Include as many cards as desired. This input ends at the next keyword (“*”) card.

Card 1	1	2	3	4	5	6	7	8
Variable	SWID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
SWID	ID of *SENSOR_SWITCH which will terminate the calculation when its condition is met. Only one *TERMINATION_SENSOR is allowed. If more than one *TERMINATION_SENSOR is defined; only the last one is effective.

Remarks:

An example allowing more than one sensor_switch to terminate calculation:

```
*SENSOR_DEFINE_ELEMENT
$ Axial force of beam element 1
44,BEAM,1,AXIAL,FORCE
*SENSOR_DEFINE_ELEMENT
$ Axial force of beam element 2
55,BEAM,21,AXIAL,FORCE
*SENSOR_SWITCH
$a switch condition is  met when the axial force of beam-1 >5.0
11,SENSOR,44,GT,5.
*SENSOR_SWITCH
$a switch condition is  met when the axial force of beam-2 >10.0
22,SENSOR,55,GT,10.
*SENSOR_SWITCH
$ a switch condition is  met when time >50.
33,TIME, , 50
*SENSOR_SWITCH_CALC-LOGIC
$ a switch condition is met if both conditions
$ of switch-11 and switch-33 are met, I.e.,
$ axial force of beam-1>5.0 and time>50
44,11,33
*SENSOR_SWITCH_CALC-LOGIC
$ a switch condition is met if both conditions
$ of switch-22 and switch-33 are met, I.e.,
$ axial force of beam-2>10.0 and time>50
55,33,22
*SENSOR_SWITCH_CALC-LOGIC
$ a switch condition is met if the conditions
$ of switch-44 or switch-55 is met, I.e.,
$ axial force of beam-1>5.0 and time>50 or
$ axial force of beam-2>10.0 and time>50
66,44,-55
*TERMINATION_SENSOR
$ job will be terminated when the switch condition of switch-66 is met, I.e.,
$ axial force of beam-1>5.0 and time>50 or
$ axial force of beam-2>10.0 and time>50
66
```

*TITLE

*TITLE

Purpose: Define job title.

Card 1	1	2	3	4	5	6	7	8
Variable	TITLE							
Type	C							
Default	LS-DYNA USER INPUT							

VARIABLE	DESCRIPTION
TITLE	Heading to appear on output and in output files.

*USER

USER_INTERFACE_*OPTION

Available options include:

CONTROL

FRICTION

FORCES

CONDUCTIVITY

Purpose: Define user defined input and allocate storage for user defined subroutines for the contact algorithms. See also *CONTROL_CONTACT. The CONTROL option above allows the user to take information from the contact interface for further action, e.g., stopping the analysis. A sample user subroutine is provided in Appendix F.

The FRICTION option may be used to modify the Coulomb friction coefficients in contact types 3, 5, or 10 (*CONTACT_SURFACE_TO_SURFACE, *CONTACT_-NODES_TO_SURFACE, or *CONTACT_ONE_WAY_SURFACE_TO_SURFACE) according to contact information or to use a friction coefficient database. A sample user-defined friction subroutine is provided in Appendix G. For the subroutine to be called, the static friction coefficient FS on Card 2 of *CONTACT must be any nonzero value, and shell thickness offsets must be invoked in the contact by setting SHLTHK to 1 or 2 using *CONTROL_CONTACT or Opt. Card B in *CONTACT. The array length USRFRC in *CONTROL_CONTACT should be set to a value no less than the sum of the number of history variables NOC and the number of user-defined input parameters in *USER_INTERFACE_FRICTION.

The CONDUCTIVITY option is used to define heat transfer contact conductance properties for thermal contacts.

Card 1	1	2	3	4	5	6	7	8
Variable	IFID	NOC	NOCI	NHSV				
Type	I	I	I	I				
Default	none	none	none	0				

Initialization Cards. Use as many cards as necessary to set NOCI variables.

Card 2	1	2	3	4	5	6	7	8
Variable	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8
Type	F	F	F	F	F	F	F	F

VARIABLE	DESCRIPTION
IFID	Interface number
NOC	Number of history variables for interface. The number should not exceed the length of the array defined on *CONTROL_CONTACT. See Remarks.
NOCI	Initialize the first NOCI history variables in the input. NOCI must be smaller or equal to NOC.
NHSV	Number of history variables per interface node (only for friction and conductivity interface).
UC1	First user defined input parameter.
UC2	Second user defined input parameter.
⋮	⋮
UC[N]	Last user defined input parameter, where N = NOCI.

The FORCES option is used to collect contact nodal forces from specified contact ID list for user subroutines.

Card 1	1	2	3	4	5	6	7	8
Variable	NCONT							
Type	I							
Default	none							

Contact ID Cards.

Card 2	1	2	3	4	5	6	7	8
Variable	CID1	CID2	CID3	CID4	CID5	CID6	CID7	CID8
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
IFID	Interface number
NCONT	Number of contact ID. If NCONT LE. 0, all contacts will be used.
CID1	First contact user ID.
CID2	Second contact user ID.
⋮	⋮
CID[N]	Last contact user ID, where N = NCONT.

Remarks:

The (NOC) interface variables (of which NOCI are initialized) are passed as arguments to the user defined subroutine. See Appendix G for the full list of arguments passed to the subroutine.

This keyword is not supported by segment based contact which is invoked by setting SOFT = 2 on optional card A of the *CONTACT card.

*USER_LOADING

Purpose: Provide a means of applying pressure and force boundary conditions. The keyword *USER_LOADING activates this option. Input here is optional with the input being read until the next "*" keyword appears. The data read here is to be stored in a common block provided in the user subroutine, LOADUD. This data is stored and retrieved from the restart files.

Parameter Cards. Add one card for each input parameter. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1…	1	2	3	4	5	6	7	8
Variable	PARM1	PARM2	PARM3	PARM4	PARM5	PARM6	PARM7	PARM8
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
PARM[N]	This is the Nth user input parameter.

*USER_LOADING_SET

Purpose: Provides a means to apply user-defined loading to a set of nodes or segments. Loading could be nodal force, body force, temperature distribution, and pressure on segment or beam.

Set Cards. Add a card for each set to which a load is applied. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	LTYPE	LCID	CID	SF1	SF2	SF3	IDULS
Type	I	A	I	I	F	F	F	I
Default	none	none	none	global	none	none	none	Seq. #

VARIABLE **DESCRIPTION**

SID
ID of the set to which user-defined loading will be applied. Set type depends on the type of loading, see LTYPE.

LTYPE
Loading type:

EQ."FORCEN": Force a will be applied to node set SID. The load is to be given in units of force.

EQ."BODYFN": Body force density will be applied to node set SID. The load is to be given in units of force per volume.

EQ."TEMPTN": Temperature will be assigned to node set SID. This option cannot be coexist with *LOAD_THERMAL_VARIABLE. In other word, users can only use either this option or *LOAD_THERMAL_VARIABLE to specify temperature distribution, not both of them,

EQ."PRESSS": Pressure will be applied to segment set SID. The load is to be given in units of force per area.

EQ."PRESSB": Pressure in units of force per length will be applied to beam set SID.

VARIABLE	DESCRIPTION
LCID	Load curve, a function of time. Its current value, crv, is passed to user subroutine LOADSETUD.
CID	Optional coordinate system along which scale factors SF_i is defined. Global system is the default system.
SF[i]	Scale factor of loading magnitude, when LTYPE LTYPE.EQ.“FORCEN”: SFi is the factor along i^{th} direction of CID. For example, set SF1 = 1. and the others to zero if the load is to be applied in the positive *x*-direction. This applies whether the global or a local coordinate system is used. LTYPE.EQ.“BODYFN”: See “EQ.FORCEN” LTYPE.EQ.“PRESSS”: SF1 is used as the scale factor, SF2 and SF3 are ignored, LTYPE.EQ.“PRESSB”: Scale factor along *r*, *s*, *t* axis of beam.
IDULS	Each USER_LOADING_SET can be assigned a unique ID, which is passed to user subroutine LOADSETUD and allows multiple loading definitions by using a single user subroutine, LOADSETUD. If no value is input, LS-DYNA will assign a sequence number to each USER_LOADING_SET based on its definition sequence.

Remarks:

*USER_LOADING_SET activates the loading defined in user subroutine LOADSETUD, part of dyn21.F. When both *USER_LOADING_SET and *USER_LOADING are defined, *USER_LOADING is only used to define user-defined parameters, PARM*n*; not to activate user subroutine LOADUD. Therefore only loading defined in LOADSETUD will be applied.

More than one loading definitions can be defined and assigned a unique ID, that enables multiple loading to be taken care of by a single subroutine, LOADSETUD, as shown below:

```
subroutine loadsetud(time,lft,llt,crv,iduls,parm)
c
c     Input (not modifiable)
c       x   : coordinate of node or element center
c       d   : displacement of node or element center
c       v   : velocity of node or element center
```

```
c       temp: temperature of node or element center
c       crv : value of LCID at current time
c     isuls : id of user_loading_set
c     parm: parameters defined in *USER_LOADING
c     Output (defined by user)
c       udl : user-defined load value
      include 'nlqparm'
C_TASKCOMMON (aux8loc)
      common/aux8loc/
     & x1(nlq),x2(nlq),x3(nlq),v1(nlq),v2(nlq),v3(nlq),
     & d1(nlq),d2(nlq),d3(nlq),temp(nlq),udl(nlq),tmp(nlq,12)
c
c    sample code
c    if (iduls.eq.100) then
c      do i=lft,llt
c       your code here
c        udl(i)=..........
c      enddo
c    elseif (iduls.eq.200) then
c      do i=lft,llt
c        udl(i)=..........
c      enddo
c    endif
      return
      end
```

Restart Input Data

In general three categories of restart actions are possible with LS-DYNA and are outlined in the following discussion:

1. A simple restart occurs when LS-DYNA was interactively stopped before reaching the termination time. Then, by specifying the R=**rtf** command line option on the execution line, LS-DYNA restarts the calculation from the termination point. The calculation will pick up at the specified termination time. see INTRODUCTION, Execution Syntax. No additional input deck is required.

2. For small modifications of the restart run LS-DYNA offer a "small restart" capability which can

   a) reset termination time,

   b) reset output printing interval,

   c) reset output plotting interval,

   d) delete contact surfaces,

   e) delete elements and parts,

   f) switch deformable bodies to rigid,

   g) switch rigid bodies to deformable,

   h) change damping options.

   All modifications to the problem made with the restart input deck will be reflected in subsequent restart dumps. All the members of the file families are consecutively numbered beginning from the last member prior to termination.

   For a small restart run a *small input deck* replaces the standard input deck on the execution line which must have at least the following command line arguments:

   LS-DYNA I=**restartinput** R=**D3DUMP*nn***

   where D3DUMP*nn* (or whatever name is chosen for the family member) is the *n*th restart file from the last run where the data is taken. LS-DYNA automatically detects that a small input deck is used since the I=**restartinput** file may contain *only* the *restart* keywords (excluding *STRESS_INITIALIZATION):

   CHANGE_*OPTION

*CONTROL_DYNAMIC_RELAXATION

*CONTROL_SHELL

*CONTROL_TERMINATION

*CONTROL_TIMESTEP

*DAMPING_GLOBAL

DATABASE_*OPTION

DATABASE_BINARY_*OPTION

DELETE_*OPTION

*INTERFACE_SPRINGBACK_LSDYNA

RIGID_DEFORMABLE_*OPTION

STRESS_INITIALIZATION_*{OPTION}

TERMINATION_*OPTION

*TITLE

*KEYWORD

*CONTROL_CPU

DEFINE_*OPTION

SET_*OPTION

The user has to take care that nonphysical modifications to the input deck are avoided; otherwise, complete nonsense may be the result.

3. If many modifications are desired a *full restart* may be the appropriate choice. A full restart is selected by including a full model along with a *STRESS_INITIALIZATION keyword card and possibly other restart cards. As mentioned in the *Restart Analysis* subsection of the *Introduction* portion of the manual, either all parts or some subset of parts can be made for the stress initialization.

Remarks:

a) In a full restart, only those nodes and elements defined in the full restart deck will be present in the analysis after the full restart is initiated. But as a convenience, any of those nodes or elements can be deleted using the *DELETE command.

b) In a small restart, velocities of nodes come from the dump file by default but those velocities can be changed using *CHANGE_VELOCITY_....

c) In a full restart, velocities of pre-existing nodes come from the dump file by default but those velocities can be changed using *CHANGE_VELOCITY_.... To set the starting velocities for new nodes in a full restart, use *INITIAL_VELOCITY_....

d) Pre-existing contacts, in general, carry forward seamlessly using data from the d3dump (or d3full if MPP) database. It is important that the contact ID(s) in the full restart input deck match the contact ID(s) in the original input deck if the intent is for the contacts to be initialized using data from the d3dump/d3full database. EXCEPTION: In the special case of MPP, a *CONTACT_AUTOMATIC_GENERAL contact in the full restart input deck is treated as a brand new contact and is not initialized using data from d3full.

e) Only sets utilizing element IDs and node IDs are permitted in a small restart deck; part IDs are not recognized. Sets referenced by other commands in a small restart deck must be defined in the small restart deck.

CHANGE_*OPTION

Purpose: Change solution options.

Available options include:

- BOUNDARY_CONDITION
- CONTACT_SMALL_PENETRATION
- CURVE_DEFINITION
- OUTPUT
- RIGIDWALL_GEOMETRIC
- RIGIDWALL_PLANAR
- RIGID_BODY_CONSTRAINT
- RIGID_BODY_INERTIA
- RIGID_BODY_STOPPER
- STATUS_REPORT_FREQUENCY
- THERMAL_PARAMETERS
- VELOCITY
- VELOCITY_GENERATION
- VELOCITY_NODE
- VELOCITY_RIGID_BODY
- VELOCITY_ZERO

Boundary Condition Cards. This card 1 format is for the ***BOUNDARY_CONDITION*** keyword option. Add one card for each boundary condition. This card imposes *additional* boundary conditions. It does not remove previously imposed conditions (for example, this option will not free a fixed node). This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	NID	BCC						
Type	I	I						

VARIABLE	DESCRIPTION
NID	Nodal point ID, see also *NODE.
BCC	New translational boundary condition code: EQ.1: constrained x displacement, EQ.2: constrained y displacement, EQ.3: constrained z displacement, EQ.4: constrained x and y displacements, EQ.5: constrained y and z displacements, EQ.6: constrained z and x displacements, EQ.7: constrained x, y, and z displacements.

Small Penetration Check Cards. This Card 1 format is for the ***CONTACT_SMALL_PENETRATION*** keyword option. Set one value for each contact surface ID where the small penetration check is to be turned on. The input terminates at the next keyword ("*") card. See the PENCHK variable in *CONTACT.

Card 1	1	2	3	4	5	6	7	8
Variable	ID1	ID2	ID3	ID4	ID5	ID6	ID7	ID8
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
ID*n*	Contact ID for surface number *n*.

Load Curve Redefinition Cards. This Card 1 format is for the ***CURVE_DEFINITION*** keyword option. *The new load curve must contain the same number of points as the curve it replaces.* The curve should be defined according to the DEFINE_CURVE section of the manual. This input terminates when the next "*" card is encountered. Offsets and scale factors are ignored.

Card 1	1	2	3	4	5	6	7	8
Variable	LCID							
Type	I							

VARIABLE	DESCRIPTION
LCID	Load curve ID

ASCII Output Overwrite Card. This format applies to the ***OUTPUT*** keyword option.

Card 1	1	2	3	4	5	6	7	8
Variable	IASCII							
Type	I							

VARIABLE **DESCRIPTION**

IASCII

Flag to control manner of outputing ASCII data requested by *DATABASE_*OPTION* commands in a full restart deck:

EQ.0: Full restart overwrites existing ASCII output (default),

EQ.1: Full restart appends to existing ASCII output.

Rigidwall modification. The format for the ***RIGIDWALL_GEOMETRIC*** and ***RIGIDWALL_PLANAR*** cards is identical to the original cards (see *RIGIDWALL_GEOMETRIC and *RIGIDWALL_PLANAR), however there are restrictions on the entries that may be changed: only those entries that define the size and orientation of the rigid walls may be changed, but not any of the others (e.g., the type).

Rigid Body Constraint Modification Cards. This format for Card 1 applies to the ***RIGID_BODY_CONSTRAINT*** keyword option. This option can change translation and rotational boundary condition on a rigid body. This input ends at the next keyword ("*") card. See CONSTRAINTED_RIGID_BODIES.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	TC	RC					
Type	I	I	I					

VARIABLE	DESCRIPTION
PID	Part ID, see *PART.
TC	Translational constraint: EQ.0: no constraints, EQ.1: constrained x displacement, EQ.2: constrained y displacement, EQ.3: constrained z displacement, EQ.4: constrained x and y displacements, EQ.5: constrained y and z displacements, EQ.6: constrained z and x displacements, EQ.7: constrained x, y, and z displacements.
RC	Rotational constraint: EQ.0: no constraints, EQ.1: constrained x rotation, EQ.2: constrained y rotation, EQ.3: constrained z rotation, EQ.4: constrained x and y rotations, EQ.5: constrained y and z rotations, EQ.6: constrained z and x rotations, EQ.7: constrained x, y, and z rotations.

Card sets for *RIGID_BODY_INERTIA* keyword option. This option supports changing the mass and inertia properties of a rigid body. Include as many pairs of the following two cards as necessary. This input ends at the next keyword ("*") card. The inertia tensor is specified relative to the coordinate system set in *MAT_RIGID at the start of the calculation, which is fixed in the rigid body and tracks the rigid body rotation.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	PID	TM					
Type	I	I	F					

Card 2	1	2	3	4	5	6	7	8
Variable	IXX	IXY	IXZ	IYY	IYZ	IZZ		
Type	F	F	F	F	F	F		

VARIABLE	DESCRIPTION
ID	ID for this change inertia input.
PID	Part ID, see *PART.
TM	Translational mass.
IXX	I_{xx}, xx component of inertia tensor.
IXY	I_{xy}
IXZ	I_{xz}
IYY	I_{yy}
IYZ	I_{yz}
IZZ	I_{zz}

Card sets for the *RIGID_BODY_STOPPERS* keyword option. This option is for redefining existing stoppers. Include as many pairs of cards as necessary. This input terminates when the next "*" card is encountered. See *CONSTRAINED_RIGID_BODY_STOPPERS. Note that new stopper definitions cannot be introduced in this section. Existing stoppers can be modified.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	LCMAX	LCMIN	PSIDMX	PSIDMN	LCVMNX	DIR	VID
Type	I	I	I	I	I	I	I	I
Default	required	0	0	0	0	0	required	0

Card 2	1	2	3	4	5	6	7	8
Variable	BIRTH	DEATH						
Type	F	F						
Default	0	10^{28}						

VARIABLE **DESCRIPTION**

PID Part ID of master rigid body, see *PART.

LCMAX Load curve ID defining the maximum coordinate as a function of time:

EQ.0: no limitation of the maximum displacement. New curves can be defined by the *DEFINE_CURVE within the present restart deck. (Not applicable for small deck restart).

LCMIN Load curve ID defining the minimum coordinate as a function of time:

EQ.0: no limitation of the minimum displacement. New curves can be defined by the *DEFINE_CURVE within the present restart deck. (Not applicable for small deck restart).

PSIDMX Optional part set ID of rigid bodies that are slaved in the maximum coordinate direction to the master rigid body. This option requires additional input by the *SET_PART definition.

VARIABLE	DESCRIPTION
PSIDMN	Optional part set ID of rigid bodies that are slaved in the minimum coordinate direction to the master rigid body. This option requires additional input by the *SET_PART definition.
LCVMNX	Load curve ID which defines the maximum absolute value of the velocity that is allowed within the stopper: EQ.0: no limitation of the minimum displacement.
DIR	Direction stopper acts in: EQ.1: x-translation, EQ.2: y-translation, EQ.3: z-translation, EQ.4: arbitrary, defined by vector VID, EQ.5: x-axis rotation, EQ.6: y-axis rotation, EQ.7: z-axis rotation, EQ.8: arbitrary, defined by vector VID.
VID	Vector for arbitrary orientation of stopper. The vector must be defined by a *DEFINE_VECTOR within the present restart deck.
BIRTH	Time at which stopper is activated.
DEATH	Time at which stopper is deactivated.

Remarks:

The optional definition of part sets in minimum or maximum coordinate directions allows the motion to be controlled in an arbitrary direction.

D3HSP Interval Change Card. This card format applies to the ***STATUS_REPORT_FREQUENCY*** keyword option.

Card 1	1	2	3	4	5	6	7	8
Variable	IKEDIT							
Type	I							

VARIABLE	DESCRIPTION
IKEDIT	Problem status report interval steps in the D3HSP output file: EQ.0: interval remains unchanged.

Card set for the *THERMAL_PARAMETERS* keyword option. This option is for changing the parameters used by a thermal or coupled structural/thermal analysis. See *CONTROL_THERMAL. Add the two following cards to the deck (they do not repeat).

Card 1	1	2	3	4	5	6	7	8
Variable	TS	DT	TMIN	TMAX	DTEMP	TSCP		
Type	I	F	F	F	F	F		

Card 2	1	2	3	4	5	6	7	8
Variable	REFMAX	TOL						
Type	I	F						

VARIABLE	DESCRIPTION
TS	Thermal time step code: EQ.0: No change, EQ.1: Fixed time step, EQ.2: variable time step.
DT	Thermal time step on restart: EQ.0: No change.
TMIN	Minimum thermal time step: EQ.0: No change.
TMAX	Maximum thermal time step: EQ.0: No change.

VARIABLE	DESCRIPTION
DTEMP	Maximum temperature change in a thermal time step: EQ.0: No change.
TSCP	Time step control parameter (0.0 < TSCP < 1.0): EQ.0: No change.
REFMAX	Maximum number of reformations per thermal time step: EQ.0: No change.
TOL	Non-linear convergence tolerance: EQ.0: No change.

Node Set Velocity Card Sets. The formats for Cards 1 and 2 apply to the ***VELOCITY*** and ***VELOCITY_ONLY*** keyword options. These options are for setting velocity fields on node sets at restart. For each node set add one pair of the following cards. This input ends at the next keyword ("*") card. Undefined nodes (not listed on a set velocity card) will have their nodal velocities reset to zero if a *CHANGE_VELOCITY definition is encountered in the restart deck. However, if any of the *CHANGE_VELOCITY definitions have ONLY appended, then only the specified nodes will have their nodal velocities modified.

Card 1	1	2	3	4	5	6	7	8
Variable	NSID							
Type	I							
Default	none							
Remark	1							

Card 2	1	2	3	4	5	6	7	8
Variable	VX	VY	VZ	VXR	VYR	VZR		
Type	F	F	F	F	F	F		
Default	0.	0.	0.	0.	0.	0.		

VARIABLE	DESCRIPTION
NSID	Nodal set ID containing nodes for initial velocity.
VX	Velocity in x-direction.
VY	Velocity in y-direction.
VZ	Velocity in z-direction.
VXR	Rotational velocity about the x-axis.
VYR	Rotational velocity about the y-axis.
VZR	Rotational velocity about the z-axis.

Remarks:

1. If a node is initialized on more than one input card set, then the last set input will determine its velocity, unless it is specified on a *CHANGE_VELOCITY_NODE card.

2. Undefined nodes will have their nodal velocities set to zero if a *CHANGE_VELOCITY definition is encountered in the restart deck.

3. If both *CHANGE_VELOCITY and *CHANGE_VELOCITY_ZERO cards are defined then all velocities will be reset to zero.

Velocity generation cards. The velocity generation cards for the ***VELOCITY_GENERATION*** option are identical to the standard velocity generation cards (see *INITIAL_VELOCITY_GENERATION), and all parameters may be changed.

Nodal Point Velocity Cards. This format applies to the ***VELOCITY_NODE*** and ***VELOCITY_NODE_ONLY*** keyword options. These option support changing nodal velocities. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	NID	VX	VY	VZ	VXR	VYR	VZR	
Type	I	F	F	F	F	F	F	
Default	none	0.	0.	0.	0.	0.	0.	

VARIABLE	DESCRIPTION
NID	Node ID
VX	Translational velocity in x-direction.
VY	Translational velocity in y-direction.
VZ	Translational velocity in z-direction.
VXR	Rotational velocity about the x-axis.
VYR	Rotational velocity about the y-axis.
VZR	Rotational velocity about the z-axis.

Remarks:

1. Undefined nodes (not listed on a point velocity card) will have their nodal velocities reset to zero if a *CHANGE_VELOCITY_NODE definition is encountered in the restart deck. However, if any of the *CHANGE_VELOCITY or CHANGE_VELOCITY_NODE definitions have_ONLY appended, then only the specified nodes will have their nodal velocities modified.

2. If a node is initialized on more than one input card set, then the last set input will determine its velocity, unless it is specified on a *CHANGE_VELOCITY_-NODE card.

3. If both *CHANGE_VELOCITY and *CHANGE_VELOCITY_ZERO cards are defined then all velocities will be reset to zero.

Rigid Body Velocity Cards. This Card 1 format applies to the ***VELOCITY_RIGID_BODY*** keyword option. This option allows for setting the velocity components of a rigid body at restart. Include as many of these cards as desired. This input ends at the next keyword (“*”) card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	VX	VY	VZ	VXR	VYR	VZR	
Type	I	F	F	F	F	F	F	
Default	none	0.	0.	0.	0.	0.	0.	

VARIABLE	DESCRIPTION
PID	Part ID of rigid body.
VX	Translational velocity in x-direction.
VY	Translational velocity in y-direction.
VZ	Translational velocity in z-direction.
VXR	Rotational velocity about the x-axis.
VYR	Rotational velocity about the y-axis.
VZR	Rotational velocity about the z-axis.

Remarks:

1. Rotational velocities are defined about the center of mass of the rigid body.

2. Rigid bodies not defined in this section will not have their velocities modified.

Restarting the Model at Rest. The ***VELOCITY_ZERO*** option resets the velocities to zero at the start of the restart. There are no data cards associated with *CHANGE_VELOCITY_ZERO.

*CONTROL_DYNAMIC_RELAXATION

Purpose: Define controls for dynamic relaxation.

Card 1	1	2	3	4	5	6	7	8
Variable	NRCYCK	DRTOL	DRFCTR	DRTERM	TSSFDR	IRELAL	EDTTL	IDRFLG
Type	I	F	F	F	F	I	F	I
Default	250	0.001	0.995	∞	TSSFAC	0	0.0	0
Remarks	1	1	1	1	1			1

VARIABLE	DESCRIPTION
NRCYCK	Number of iterations between convergence checks, for dynamic relaxation option (default = 250).
DRTOL	Convergence tolerance for dynamic relaxation option (default = 0.001).
DRFCTR	Dynamic relaxation factor (default = .995).
DRTERM	Optional termination time for dynamic relaxation. Termination occurs at this time or when convergence is attained (default = infinity).
TSSFDR	Scale factor for computed time step during dynamic relaxation. If zero, the value is set to TSSFAC defined on *CONTROL_TERMINATION. After converging, the scale factor is reset to TSSFAC.
IRELAL	Automatic control for dynamic relaxation option based on algorithm of Papadrakakis [1981].
EDTTL	Convergence tolerance on automatic control of dynamic relaxation.
IDRFLG	Dynamic relaxation flag for stress initialization: EQ.0: not active, EQ.1: dynamic relaxation is activated.

Remarks:

1. If a dynamic relaxation relaxation analysis is being restarted at a point before convergence was obtained, then NRCYCK, DRTOL, DRFCTR, DRTERM and TSSFDR will default to their previous values, and IDRFLG will be set to 1.

2. If dynamic relaxation is activated after a restart from a normal transient analysis LS-DYNA continues the output of data as it would without the dynamic relaxation being active. This is unlike the dynamic relaxation phase at the beginning of the calculation when a separate database is not used. Only load curves that are flagged for dynamic relaxation are applied after restarting.

*CONTROL_SHELL

Purpose: Change failure parameters NFAIL1 and NFAIL4 if necessary. These parameters must be nonzero in the initial run.

Card 1	1	2	3	4	5	6	7	8
Variable								
Type								

Card 2	1	2	3	4	5	6	7	8
Variable						NFAIL1	NFAIL4	PSNFAIL
Type						I	I	I

VARIABLE	DESCRIPTION
NFAIL1	Flag to check for highly distorted under-integrated shell elements, print a message, and delete the element or terminate. Generally, this flag is not needed for one point elements that do not use the warping stiffness. A distorted element is one where a negative jacobian exists within the domain of the shell, not just at integration points. The checks are made away from the integration points to enable the bad elements to be deleted before an instability leading to an error termination occurs. This test will increase CPU requirements for one point elements. EQ.1: print message and delete element. EQ.2: print message, write d3dump file, and terminate GT.2: print message and delete element. When NFAIL1 elements are deleted then write D3dump file and terminate. These NFAIL1 failed elements also include all shell elements that failed for other reasons than distortion. Before the d3dump file is written, NFAIL1 is doubled, so the run can immediately be continued if desired.

VARIABLE **DESCRIPTION**

NFAIL4

Flag to check for highly distorted fully-integrated shell elements, print a message, and delete the element or terminate. Generally, this flag is recommended. A distorted element is one where a negative jacobian exists within the domain of the shell, not just at integration points.

The checks are made away from the integration points to enable the bad elements to be deleted before an instability leading to an error termination occurs.

EQ.1: print message and delete element.

EQ.2: print message, write d3dump file, and terminate

GT.2: print message and delete element. When NFAIL4 elements are deleted then write d3dump file and terminate. These NFAIL4 failed elements also include all shell elements that failed for other reasons than distortion. Before the d3dump file is written, NFAIL4 is doubled, so the run can immediately be continued if desired.

PSNFAIL

Optional shell part set ID specifying which part IDs are checked by the NFAIL1, NFAIL4, and W-MODE options. If zero, all shell part IDs are included.

*CONTROL_TERMINATION

Purpose: Stop the job.

Card	1	2	3	4	5	6	7	8
Variable	ENDTIM	ENDCYC						
Type	F	I						

VARIABLE **DESCRIPTION**

ENDTIM Termination time:

EQ.0.0: Termination time remains unchanged.

ENDCYC Termination cycle. The termination cycle is optional and will be used if the specified cycle is reached before the termination time.

EQ.0.0: Termination cycle remains unchanged.

Remarks:

This is a reduced version of the *CONTROL_TERMINATION card used in the initial input deck.

*CONTROL_TIMESTEP

Purpose: Set time step size control using different options.

Card	1	2	3	4	5	6	7	8
Variable	DUMMY	tssfac	ISDO	DUMMY	DT2MS	LCTM		
Type	F	F	I	F	F	I		

VARIABLE **DESCRIPTION**

DUMMY Dummy field, see remark 1 below.

TSSFAC Scale factor for computed time step.

EQ.0.0: TSSFAC remains unchanged.

ISDO Basis of time size calculation for 4-node shell elements, ISDO 3-node shells use the shortest altitude for options 0,1 and the shortest side for option 2. This option has no relevance to solid elements, which use a length based on the element volume divided by the largest surface area:

EQ.0: characteristic length = area/(longest side),

EQ.1: characteristic length = area/(longest diagonal),

EQ.2: based on bar wave speed and MAX [shortest side, area/longest side]. THIS LAST OPTION CAN GIVE A MUCH LARGER TIME STEP SIZE THAT CAN LEAD TO INSTABILITIES IN SOME APPLICATIONS, ESPECIALLY WHEN TRIANGULAR ELEMENTS ARE USED.

DUMMY Dummy field, see remark 1 below.

DT2MS New time step for mass scaled calculations. Mass scaling must be active in the time zero analysis.

EQ.0.0: DT2MS remains unchanged.

LCTM Load curve ID that limits maximum time step size:

EQ.0: LCTM remains unchanged.

Remarks:

1. This a reduced version of the *CONTROL_TIMESTEP used in the initial analysis. The dummy fields are included to maintain compatibility. If using free format input then a 0.0 should be entered for the dummy values.

*DAMPING_GLOBAL

Purpose: Define mass weighted nodal damping that applies globally to the deformable nodes.

Card	1	2	3	4	5	6	7	8
Variable	LCID	VALDMP						
Type	I	F						
Default	0	0.0						

VARIABLE	DESCRIPTION
LCID	Load curve ID which specifies node system damping: EQ.n: system damping is given by load curve n. The damping force applied to each node is f = -d(t) mv, where d(t) is defined by load curve n.
VALDMP	System damping constant, d (this option is bypassed if the load curve number defined above is nonzero).

DATABASE_*OPTION

Options for ASCII files include. If a file is not specified in the restart deck then the output interval for the file will remain unchanged.

SECFORC	Cross section forces.
RWFORC	Wall forces.
NODOUT	Nodal point data.
ELOUT	Element data.
GLSTAT	Global data.
DEFORC	Discrete elements.
MATSUM	Material energies.
NCFORC	Nodal interface forces.
RCFORC	Resultant interface forces.
DEFGEO	Deformed geometry file
SPCFORC	Set dt for spc reaction forces.
SWFORC	Nodal constraint reaction forces (spot welds and rivets).
ABSTAT	Set dt for airbag statistics.
NODFOR	Set dt for nodal force groups.
BNDOUT	Boundary condition forces and energy
RBDOUT	Set dt for rigid body data.
GCEOUT	Set dt for geometric contact entities.
SLEOUT	Set dt for sliding interface energy.
JNTFORC	Set dt for joint force file.
SBTOUT	Set dt for seat belt output file.
AVSFLT	Set dt for AVS database.
MOVIE	Set dt for MOVIE.
MPGS	Set dt for MPGS.
TPRINT	Set dt for thermal file.

Card	1	2	3	4	5	6	7	8
Variable	DT							
Type	F							

VARIABLE	DESCRIPTION
DT	Time interval between outputs: EQ.0.0: output interval is unchanged.

Remarks:

To terminate output to a particular file set DT to a high value.

If IACCOP = 2 was specified in *CONTROL_OUTPUT, the best results are obtained in the NODOUT file by keeping the same DT on restart. When DT is changed for NODOUT, oscillations may occur around the restart time. If DT is larger than initially specified in the original input file, more memory is required to store the time states for the averaging than was originally allocated. A warning message is printed, and the filtering is applied using the available memory. When DT is smaller than initially specified, more oscillations may appear in the output than earlier in the calculation because the frequency content of the averaged output increases as DT decreases.

DATABASE_BINARY_*OPTION

Options for binary output files with the default names given include:

D3PLOT	Dt for complete output states.
D3THDT	Dt for time history data for element subsets.
D3DUMP	Binary output restart files. Define output frequency in cycles
RUNRSF	Binary output restart file. Define output frequency in cycles.
INTFOR	Dt for contact surface Interface database.

Card	1	2	3	4	5	6	7	8
Variable	DT/CYCL							
Type	F							

VARIABLE **DESCRIPTION**

DT Time interval between outputs.

EQ.0.0: Time interval remains unchanged.

CYCL Output interval in time steps.

EQ.0.0: output interval remains unchanged.

DELETE_*OPTION

Available options are:

ALECPL

CONTACT

CONTACT_2DAUTO

ENTITY

PART

ELEMENT_BEAM

ELEMENT_SHELL

ELEMENT_SOLID

ELEMENT_TSHELL

FSI

Purpose: Delete contact surfaces, ALE FSI couplings, parts, or elements by a list of IDs. There are two contact algorithms for two-dimensional problems: the line-to-line contact and the automatic contact defined by part ID's. Each uses their own numbering.

ID Cards. This card 1 format applies to the ALECPL, CONTACT, CONTACT_2DAUTO, ENTITY, FSI and PART options. Include as many cards as necessary to input desired IDs. This input ends at the next keyword ("*") card.

Card	1	2	3	4	5	6	7	8
Variable	ID1	ID2	ID3	ID4	ID5	ID6	ID7	ID8
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
IDn	Contact ID/Coupling ID/Part ID

Remarks:

The FSI option corresponds to ALE couplings defined with *CONSTRAINED_LAGRANGE_IN_SOLID. The ALECPL option corresponds to ALE couplings defined with *ALE_COUPLING_NODAL_CONSTRAINT. For CONTACT, FSI, and ALECPL options, a negative ID implies that the absolute value gives the contact surface/FSI/ALECPL coupling which is to be activated.

Element set cards. This card 1 format applies to the four ELEMENT options. This input ends at the next keyword ("*") card.

Card	1	2	3	4	5	6	7	8
Variable	ESID							
Type	I							

VARIABLE	DESCRIPTION
ESID	Element set ID, see *SET_SOLID, *SET_BEAM, *SET_SHELL, *SET_TSHELL.

*INTERFACE_SPRINGBACK_LSDYNA

Purpose: Define a material subset for output to a stress initialization file “dynain”. The dynain file contains keyword commands that can be included in a subsequent input deck to initialize deformation, stress, and strain in parts. This file can be used, for example, to do an implicit springback analysis after an explicit forming analysis.

Part Set ID Cards.

Card 1	1	2	3	4	5	6	7	8
Variable	PSID	NSHV						
Type	I	I						

Constraint Cards. Optional cards that list of nodal points that are constrained in the dynain file. This input ends at the next keyword (“*”) card.

Card 2	1	2	3	4	5	6	7	8
Variable	NID	TC	RC					
Type	I	F	F					
Default	none	0.	0.					

VARIABLE	DESCRIPTION
PSID	Part set ID for springback, see *SET_PART.
NSHV	Number of shell or solid history variables (beyond the six stresses and effective plastic strain) to be initialized in the interface file. For solids, one additional state variable (initial volume) is also written. If NSHV is nonzero, the element formulations, calculational units, and constitutive models should not change between runs. If NHSV exceeds the number of integration point history variables required by the constitutive model, only the number required is written; therefore, if in doubt, set NHSV to alarge number.
NID	Node ID

VARIABLE	DESCRIPTION
TC	Translational constraint: EQ.0: no constraints, EQ.1: constrained x displacement, EQ.2: constrained y displacement, EQ.3: constrained z displacement, EQ.4: constrained x and y displacements, EQ.5: constrained y and z displacements, EQ.6: constrained z and x displacements, EQ.7: constrained x, y, and z displacements.
RC	Rotational constraint: EQ.0: no constraints, EQ.1: constrained x rotation, EQ.2: constrained y rotation, EQ.3: constrained z rotation, EQ.4: constrained x and y rotations, EQ.5: constrained y and z rotations, EQ.6: constrained z and x rotations, EQ.7: constrained x, y, and z rotations.

RIGID_DEFORMABLE_*OPTION

Available options include:

CONTROL

D2R (Deformable to rigid part switch)

R2D (Rigid to deformable part switch)

Purpose: Define parts to be switched from rigid to deformable and deformable to rigid in a restart. It is only possible to switch parts on a restart if part switching was activated in the time zero analysis. See *DEFORMABLE_TO_RIGID for details of part switching.

*RIGID_DEFORMABLE_CONTROL

Card	1	2	3	4	5	6	7	8
Variable	NRBF	NCSF	RWF	DTMAX				
Type	I	I	I	F				
Default	0	0	0	none				

VARIABLE	DESCRIPTION
NRBF	Flag to delete or activate nodal rigid bodies. If nodal rigid bodies or generalized, weld definitions are active in the deformable bodies that are switched to rigid, then the definitions should be deleted to avoid instabilities: EQ.0: no change, EQ.1: delete, EQ.2: activate.
NCSF	Flag to delete or activate nodal constraint set. If nodal constraint/spot weld definitions are active in the deformable bodies that are switched to rigid, then the definitions should be deleted to avoid instabilities: EQ.0: no change, EQ.1: delete, EQ.2: activate.
RWF	Flag to delete or activate rigid walls: EQ.0: no change, EQ.1: delete, EQ.2: activate.
DTMAX	Maximum permitted time step size after restart.

*RIGID_DEFORMABLE_D2R

Part ID Cards. Include one card for each part. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	MRB						
Type	I	I						
Default	none	0						

VARIABLE	DESCRIPTION
PID	Part ID of the part which is switched to a rigid material.
MRB	Part ID of the master rigid body to which the part is merged. If zero, the part becomes either an independent or master rigid body.

*RIGID_DEFORMABLE_R2D

Termination of this input is when the next "*" card is read.

Part ID Cards. Include one card for each part. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
PID	Part ID of the part which is switched to a deformable material.

STRESS_INITIALIZATION_*{OPTION}

This keyword causes a full deck restart. For a full deck restart the input deck must contain the full model. The stress initialization feature allows all or selected parts to be initialized from the previous calculation using data from the d3dump or runrsf databases.

The options that are available with this keyword are:

<BLANK>

DISCRETE

SEATBELT

Optional Part Cards. If no part cards are included in the deck then all parts, seatbelts and discrete parts in the new input deck that existed in the previous input deck (with or without the same part IDs) are initialized from the d3dump or runrsf database. Otherwise for each part to be initialized from the restart data include an addition card in format 1. This input terminates at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PIDO	PIDN						
Type	I	I						
Default	none	PIDO						

VARIABLE	DESCRIPTION
PIDO	Old part ID, see *PART.
PIDN	New part ID, see *PART: EQ.0: New part ID is the same as the old part ID.

Remarks:

If one or more of the above cards are defined then discrete and seatbelt elements will not be initialized unless the additional option cards *STRESS_INITIALIZATION_DISCRETE and *STRESS_INITIALIZATION_SEATBELT are defined.

*STRESS_INITIALIZATION_DISCRETE

Initialize all discrete parts from the old parts. No further input is required with this card. This card is not required if *STRESS_INITIALIZATION is specified without further input.

*STRESS_INITIALIZATION_SEATBELT

Initialize all seatbelt parts from the old parts. No further input is required with this card. This card is not required if *STRESS_INITIALIZATION is specified without further input.

TERMINATION_*OPTION

Purpose: Stops the job depending on some displacement conditions.

Available options include:

NODE

BODY

Caution: The inputs are different for the nodal and rigid body stop conditions. The nodal stop condition works on the global coordinate position, while the body stop condition works on the relative global translation. The number of termination conditions cannot exceed the maximum of 10 or the number specified in the original analysis.

The analysis terminates for *TERMINATION_NODE when the current position of the node specified reaches either the maximum or minimum value (stops 1, 2 or 3), or picks up force from any contact surface (stop 4). For *TERMINATION_BODY the analysis terminates when the center of mass displacement of the rigid body specified reaches either the maximum or minimum value (stops 1, 2 or 3) or the displacement magnitude of the center of mass is exceeded (stop 4). If more than one condition is input, the analysis stops when any of the conditions is satisfied.

NOTE: This input completely overrides the existing termination conditions defined in the time zero run.

Termination by other means is controlled by the *CONTROL_TERMINATION control card.

Node/Part Cards. Include an additional card in format 1 for each node or part with a termination criterion

Card	1	2	3	4	5	6	7	8
Variable	NID/PID	STOP	MAXC	MINC				
Type	I	I	F	F				
Default	none	none	-	-				

For the NODE option:

VARIABLE	DESCRIPTION
NID	Node ID
STOP	Stop criterion: EQ.1: global x direction, EQ.2: global y direction, EQ.3: global z direction, EQ.4: stop if node touches contact surface.
MAXC	Maximum (most positive) coordinate, options 1, 2 and 3 above only.
MINC	Minimum (most negative) coordinate, options 1, 2 and 3 above only.

For the BODY option:

VARIABLE	DESCRIPTION
PID	Part ID of rigid body

VARIABLE **DESCRIPTION**

STOP Stop criterion:

EQ.1: global x direction,

EQ.2: global y direction,

EQ.3: global z direction,

EQ.4: stop if displacement magnitude is exceeded.

MAXC Maximum (most positive) displacement, options 1, 2, 3 and 4:

EQ.0.0: MAXC set to 1.0e21

MINC Minimum (most negative) displacement, options 1, 2 and 3 above only:

EQ.0.0: MINC set to -1.0e21

*TITLE

Purpose: Define job title.

Card	1	2	3	4	5	6	7	8
Variable	TITLE							
Type	C							
Default	LS-DYNA USER INPUT							

VARIABLE	DESCRIPTION
TITLE	Heading to appear on output.

REFERENCES

Abbo, A.J., and S.W. Sloan, "A Smooth Hyperbolic Approximation to the Mohr-Coulomb Yield Criterioin," *Computers and Structures*, Vol. 54, No. 1, (1995).

Allen, D.J., Rule, W.K., Jones, S.E., "Optimizing Material Strength Constants Numerically Extracted from Taylor Impact Data", *Experimental Mechanics*, Vol. 37, No 3, September (1997).

Allman, D.J., "A Compatible Triangular Element Including Vertex Rotations for Plane Elasticity Analysis," *Computers and Structures*, 19, 1-8, (1984).

Anagonye, A.U. and J.T. Wang, "A Semi-Empirical Method for Estimating the Effective Leak and Vent Areas of an Airbag", AMD-Vol. 237/BED-Vol. 45, pp. 195-217, (1999).

Anand, L. and M.E. Gurtin, "A theory of amorphous solids undergoing large deformations, with application to polymeric glasses," *International Journal of Solids and Structures*, 40, pp. 1465-1487 (2003)

Aretz, H. "Applications of a New Plane Stress Yield Function to Orthotropic Steel and Aluminum Sheet Metals," *Modeling and Simulation in Materials Science and Engineering*, 12, 491-509 (2004).

Argon, AS., "A theory for the low-temperature plastic deformation of glassy polymers", *Philosophical Magazine*, 28, 839-865 (1973).

Armstrong, P.J., and Frederick, C.O., "A Mathematical Representation of the Multiaxial Bauschinger Effect," CEGB Report, RD/B/N731, Berkeley Nuclear Laboratories (1966).

Arruda, E. and M. Boyce, "A Three-Dimensional Constitutive Model for the Large Stretch Behavior of Rubber Elastic Materials," *Journal of the Mechanics and Physics of Solids*, Vol. 41, No. 2, pp. 389-412, (1993).

Auricchio, F., R.L. Taylor and J. Lubliner, "Shape-memory alloys: macromodeling and numerical simulations of the superelastic behavior", *Computer Methods in Applied Mechanics and Engineering*, vol. 146, pp. 281-312, (1997).

Auricchio, F. and R.L. Taylor, "Shape-memory alloys: modeling and numerical simulations of the finite-strain superelastic behavior", *Computer Methods in Applied Mechanics and Engineering*, vol. 143, pp. 175-194, (1997).

Bahler AS: The series elastic element of mammalian skeletal muscle. *Am J Physiol* 213:1560-1564, (1967).

Baker, E.L., “An Explosives Products Thermodynamic Equation of State Appropriate for Material Acceleration and Overdriven Detonation: Theoretical Background and Fourmulation,” Technical Report ARAED-TR-911013, U.S. Army Armament Research, Development and Engineering Center, Picatinney Arsenal, New Jersey, 1991).

Baker, E.L. and J. Orosz, J., “Advanced Warheads Concepts: An Advanced Equation of State for Overdriven Detonation,” Technical Report ARAED-TR-911007, U.S. Army Armament Research, Development and Engineering Center, Picatinney Arsenal, New Jersey, (1991).

Baker, E.L. and L.I. Stiel, “Improved Quantitative Explosive Performance Prediction Using Jaguar,” 1997 Insensitive Munitions and Energetic Materials Technology Symposium, Tampa, FL, (1997).

Bammann, D.J. and E.C. Aifantis, “A Model for Finite-Deformation Plasticity,” *Acta Mechanica*, 70, 1-13 (1987).

Bammann, D.J. and G. Johnson, “On the Kinematics of Finite-Deformation Plasticity,” *Acta Mechanica, 69,* 97-117 (1987).

Bammann, D.J., “Modeling the Temperature and Strain Rate Dependent Large Deformation of Metals,” Proceedings of the 11th US National Congress of Applied Mechanics, Tuscon, AZ, (1989).

Bammann, D.J., M.L. Chiesa, A. McDonald, W.A. Kawahara, J.J. Dike, and V.D. Revelli, “Predictions of Ductile Failure in Metal Structures,” in AMD-Vol. 107, Failure Criteria and Analysis in Dynamic Response, Edited by. H.E. Lindberg, 7-12, (1990).

Bandak, F.A., private communications, U.S. Dept. of Trans., Division of Biomechanics Research, 400 7th St., S.W. Washington, D.C. 20590 (1991).

Bansal, S., Mobasher, B., Rajan, S., and Vintilescu, I., “Development of Fabric Constitutive Behavior for Use in Modeling Engine Fan Blade-Out Events.” J. Aerosp. Eng. 22, SPECIAL ISSUE: Ballistic Impact and Crashworthiness Response of Aerospace Structures, 249–259, (2009).

Barlat, F. and J. Lian, “Plastic Behavior and Stretchability of Sheet Metals. Part I: A Yield Function for Orthotropic Sheets Under Plane Stress Conditions,” *Int. J. of Plasticity*, Vol. 5, pp. 51-66 (1989).

Barlat, F., D.J. Lege, and J.C. Brem, “A Six-Component Yield Function for Anisotropic Materials,” *Int. J. of Plasticity*, 7, 693-712, (1991).

Barlat, F., Y. Maeda, K. Chung, M. Yanagawa, J.C. Brem, Y. Hayashida, D.J. Lege, K. Matsui, S.J. Murtha, S. Hattori, R.C. Becker, and S. Makosey, “Yield Function

Development for Aluminum Alloy Sheets", J. Mech. Phys. Solids, Vol. 45, No. 11-12, 1727-1763, (1997).

Barlat, F., Brem, J.C., Yoon, J.W., Chung, K., Dick, R.E., Lege, D.J., Pourboghrat, F., Choi, S.H., Chu, E., "Plane Stress Yield Function for Aluminum Alloy Sheets – Part 1: Theory, Int. J. Plast. 19, 1-23, (2003).

Basu, U., "Explicit finite element perfectly matched layer for transient three-dimensional elastic waves," *International Journal for Numerical Methods in Engineering*, vol. 77, pp. 151–176, (2009).

Basu, U. and Chopra, A.K., "Perfectly matched layers for time-harmonic elastodynamics of unbounded domains theory and finite-element implementation," *Computer Methods in Applied Mechanics and Engineering*, vol. 192, pp. 1337–1375, (2003).

Basu, U. and Chopra, A.K., "Perfectly matched layers for transient elastodynamics of unbounded domains," *International Journal for Numerical Methods in Engineering*, vol. 59, pp. 1039–1074, (2004). Erratum: Ibid. vol. 61, pp. 156–157, (2004).

Bathe, K.-J. Conserving energy and momentum in nonlinear dynamics: A simple implicit time integration scheme, *Computers and Structures*, 85, 437-445 (2007).

Bathe, K.-J. and Dvorkin, E.N. A four node plate bending element based on Mindlin-Reissner plate theory and a mixed interpolation*, Int. J. Num. Meth. Eng.*, 21, 367-383 (1985).

Bathe, K-J. and Nooh, G. Insight into an implicit time integration scheme for structural dynamics, *Computers and Structures*, 98/99, 1-6 (2012).

Batoz, J.L. and Ben Tahar, M. Evaluation of a new quadrilateral thin plate bending element, Int. *J. Num. Meth. Eng.*, 18, 1644-1677 (1982).

Batoz, J.-L. and M. Ben Tahar, Evaluation of a new quadrilateral thin plate bending element, *International Journal for Numerical Methods in Engineering,* 18, (1982), 1655-1677.

Bazeley, G.P., W.K. Cheung, R.M. Irons, and O.C. Zienkiewicz, "Triangular Elements in Plate Bending-Confirming and Nonconforming Solutions in Matrix Methods and Structural Mechanics," Proc. Conf. on Matrix Methods in Structural Analysis, Rept. AFFDL-R-66-80, Wright Patterson AFB, 547-576 (1965).

Belytschko, T. and Bindeman, L. P. "Assumed Strain Stabilization of the Eight Node Hexahedral Element," Comp. Meth. Appl. Mech. Eng. **105**, 225-260 (1993).

Belytschko, T.B. and A.H. Marchertas, “Nonlinear Finite Element Method for Plates and its Application to the Dynamic Response of Reactor Fuel Subassemblies,” *Trans, ASME J. Pressure Vessel Tech.*, 251-257 (1974).

Belytschko, T.B. and C.S. Tsay, “Explicit Algorithms for Nonlinear Dynamics of Shells,” AMD-Vol.48, ASME, 209-231 (1981).

Belytschko, T.B. and C.S. Tsay, “Explicit Algorithms for Nonlinear Dynamics of Shells,” *Comp. Meth. Appl. Mech. Eng.*, 43, 251-276, (1984).

Belytschko, T.B. and C.S. Tsay, “A Stabilization Procedure for the Quadrilateral Plate Element with One-Point Quadrature,” *Int. J. Num. Method. Eng.*, 19, 405-419 (1983).

Belytschko, T.B., H. Stolarski, and N. Carpenter, “A C° Triangular Plate Element with One-Point Quadrature,” *Int. J. Num. Meth. Eng.*, 20, 787-802 (1984).

Belytschko, T.B., I. Yeh, “The Splitting Pinball Method for Contact-Impact Problems,” *Comp. Meth. Appl. Mech. Eng.*, 105, 375-393, (1993).

Belytschko, T.B., L. Schwer, and M.J. Klein, “Large Displacement Transient Analysis of Space Frames,” *Int. J. Num. Eng.*, 11, 65-84 (1977).

Benson, D.J. and J.O. Hallquist, “A Simple Rigid Body Algorithm for Structural Dynamics Programs,” *Int. J. Numer. Meth. Eng.*, 22, (1986).

Benson, D.J. and J.O. Hallquist, “A Single Surface Contact Algorithm for the Postbuckling Analysis of Shell Structures,” Comp. Meths. Appl. Mech. Eng., 78, 141-163 (1990).

Benzeggagh, M.L. and Kenane, M., “Measurement of Mixed-mode Delamination Fracture Toughness of Unidirectional Glass/Epoxy Composites with Mixed-mode Bending Apparatus,” Composites Science and Technology, 56, 439-449 (1996).

Berstad, T., “Material Modeling of Aluminium for Crashworthiness Analysis”, Dr.Ing. Dissertation, Department of Structural Engineering, Norwegian University of Science and Technology, Trondheim, Norway, (1996).

Berstad, T., Hopperstad, O.S., Lademo, O.-G. and Malo, K.A., “Computational Model of Ductile Damage and Fracture in Shell Analysis”, Second European LS-DYNA Conference, Gothenburg, Sweden, (1999).

Berstad, T., Lademo, O.-G., Pedersen, K.O. and Hopperstad, O.S., “*Formability modeling with LS-DYNA*”, 8th International LS-DYNA User’s Conference, Detroit, May 3-5, 2004.

Berstad, T., Langseth, M. and Hopperstad, O.S., "Elasto-viscoplastic Constitutive Models in the Explicit Finite Element Code LS-DYNA3D," Second International LS-DYNA3D conference, San Francisco, (1994).

Bergström, J.S. and M.C. Boyce, "Constitutive modeling of the large strain time-dependent behavior of elastomers" *J. Mech. Phys. Solids*, 46, 931-954 (1998).

Bielak, J. and Christiano, P., "On the effective seismic input for non-linear soil-structure interaction systems," *Earthquake Engineering and Structural Dynamics*, vol. 12, pp. 107–119, (1984).

Bier, M., Sommer, S., "Simplified modeling of self-piercing riveted joints for crash simulation with a modified version of *CONSTRAINED_INTERPOLATION_SPOTWELD". 9th European LS-DYNA Conference, Manchester, (2013).

Bilkhu, S.S., M. Founas, and G.S. Nasholtz, "Material Modeling of Structural Foams in Finite Element Analysis Using Compressive Uniaxial and Triaxial Data," SAE (Nat. Conf.) Detroit 1993, pp. 4-34.

Blatz, P.J., and Ko, W.L., "Application of Finite Element Theory to the Deformation of Rubbery Materials," *Trans. Soc. of Rheology*, 6, 223-251 (1962).

Borrvall T., Bhalsod D., Hallquist J.O., and Wainscott B., "Current Status of Subcycling and Multiscale Simulations in LS-DYNA". 13th International LS-DYNA User's Conference, Dearborn MI, (2014).

Boyce, M.C., Parks, D.M., and Argon, A.S., "Large inelastic deformation of glassy polymers. Part I: Rate dependent constitutive model". *Mechanics of Materials*, 7, 15-33 (1988).

Boyce, M.C., Socrate, C. and Llana, P.G., "Constitutive model for the finite deformation stress-strain behavior of poly(ethylene terephthalate) above the glass transition". *Polymer*, 41, 2183-2201 (2000).

Brandt, J., On Constitutive Modelling of the Compaction and Sintering of Cemented Carbides, Linköping Studies in Science and Technology, Dissertations 515, 1998.

Brekelmans, W.A.M., Scheurs,P.J.G., and de Vree, J.H.P., 1991, "Continuum damage mechanics for softening of brittle materials", *Acta Mechanica*, vol 93, pp 133-143

Broadhouse, B.J., "The Winfrith Concrete Model in LS-DYNA3D," Report: SPD/D(95)363, Structural Performance Department, AEA Technology, Winfrith Technology Centre, U.K. (1995).

Broadhouse, B.J. and Neilson, A.J., "Modelling Reinforced Concrete Structures in DYNA3D", Safety and Engineering Division, United Kingdom Atomic Energy Authority, Winfrith, AEEW-M 2465, 1987.

Brode, H.L., "Height of Burst Effects at High Overpressure," RAND, RM-6301-DASA, DASA 2506, (1970).

Brown, B.E. and J.O. Hallquist, "TAURUS: An Interactive Post-Processor for the Analysis Codes NIKE3D, DYNA3D, TACO3D, and GEMINI," University of California, Lawrence Livermore National Laboratory, Rept. UCID-19392 (1982) Rev. 1 (1984).

Bruneau, M., Uang, C.M., Whittaker, A., Ductile Design of Steel Structures, McGraw Hill, (1998).

Burton, D.E. et al. "Physics and Numerics of the TENSOR Code," Lawrence Livermore National Laboratory, Internal Document UCID-19428, (July 1982).

Carney, K.S., S.A. Howard, B.A. Miller, and D.J. Benson. "New Representation of Bearings in LS-DYNA," 13th International LS-DYNA Users Conference, Dearborn, MI, June 8-10, 2014.

CEB Code 1993, Comite euro-international du beton, *CEB-FIP Model Code 1990*, Thomas Telford, London, (1993).

Chang, F.K. and K.Y. Chang, "A Progressive Damage Model for Laminated Composites Containing Stress Concentration," *J. of Composite Materials*, 21, 834-855 (1987a).

Chang, F.K. and K.Y. Chang, "Post-Failure Analysis of Bolted Composite Joints in Tension or Shear-Out Mode Failure," *J. of Composite Materials*, 21 809-823 (1987b).

Chang, F.S., "Constitutive Equation Development of Foam Materials," Ph.D. Dissertation, submitted to the Graduate School, Wayne State University, Detroit, Michigan (1995).

Chao, Y. J., Wang, K, Miller, K. W. and Zhu, X. K. "Dynamic Separation of Resistance Spot Welded joints: Part I – Experiments", *Exp Mech*, Vol. 50, Issue 7, pp 889-900 (2010).

Chen, W.F., and Baladi, G.Y., Soil Plasticity: Theory and Implementation, Elesvier, New York, (1985).

Cheng, H., Obergefell, L.A., and Rizer, A., March 1994, "Generator of Body (GEBOD) Manual," Report No. AL/CF-TR-1994-0051.

Chowdhury, S.R. and Narasimhan R., "A Cohesive Finite Element Formulation for Modeling Fracture and Delamination in Solids," *Sadhana*, 25(6), 561-587, (2000).

Christensen, R.M. "A Nonlinear Theory of Viscoelasticity for Application to Elastomers," *Journal of Applied Mechanics,* Volume 47, American Society of Mechanical Engineers, pages 762-768, December 1980.

Chu, C.C. and A. Needleman, "Void Nucleation Effects in Biaxially Stretched Sheets", ASME Journal of Engineering Materials and Technology, 102, 249-256 (1980).

Chung, K. and K. Shah, "Finite Element Simulation of Sheet Metal Forming for Planar Anisotropic Metals," *Int. J. of Plasticity*, 8, 453-476, (1992).

Cochran, S.G. and J. Chan, "Shock Initiation and Detonation Models in One and Two Dimensions," University of California, Lawrence Livermore National Laboratory, Rept. UCID-18024 (1979).

Cook, R. D., Concepts and Applications of Finite Element Analysis, John Wiley and Sons, Inc. (1974).

Couch, R., E. Albright, and N. Alexander, The Joy Computer Code, Lawrence Livermore National Laboratory, Internal Document Rept. UCID-19688, (January, 1983).

Cowper, G.R. and P.S. Symonds, Strain Hardening and Strain Rate Effects in the Impact Loading of Cantilever Beams, Brown University, Applied Mathematics Report, 1958.

CRAY-1 Computer System CFT Reference Manual, Cray Research Incorporated, Bloomington, NM., Publication No. 2240009 (1978).

Dal, H. and M. Kaliske, "Bergström-Boyce model for nonlinear finite rubber viscoelasticity: theoretical aspects and algorithmic treatment for the FE method" *Computational Mechanics*, 44(6), 809-823, (2009).

DeRuntz, J.A. Jr., "Reference Material for USA, The Underwater Shock Analysis Code, USA-STAGS, and USA-STAGS-CFA," Report LMSC-P032568, Computational Mechanics Laboratory, Lockheed Palo Alto Research Laboratory, Palo Alto, CA. (1993).

Desai, C.S., and H.J. Siriwardane, Constitutive Laws for Engineering Materials with Emphasis On Geologic Materials, Prentice-Hall, Chapter 10, (1984).

Deshpande, V.S. and N.A. Fleck, "Isotropic Models for Metallic Foams," *Journal of the Mechanics and Physics of Solids*, 48, 1253-1283, (2000).

Dick, R.E., and W.H. Harris, "Full Automated Rezoning of Evolving Geometry Problems," Numerical Methods in Industrial Forming Processes, Chenot, Wood, and Zienkiewicz, Editors, Bulkema, Rotterdam, 243-248, (1992).

Dilger, W.H., R. Koch, and R. Kowalczyk, "Ductility of Plain and Confined Concrete Under Different Strain Rates," *ACI Journal,* January-February, (1984).

Dobratz, B.M., "LLNL Explosives Handbook, Properties of Chemical Explosives and Explosive Simulants," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-52997 (1981).

Du Bois, P.A., "Numerical Simulation of Strandfoam" Daimler-Chrysler AG Abt. EP/CSV, Report (2001).

Dufailly, J., and Lemaitre, J., "Modeling very low cycle fatigue", *International Journal of damage mechanics*, 4, pp. 153-170 (1995).

Erhart, T., Andrade, F., Du Bois, P., "Short Introductionof a New Generalized Damage Model", 11th European LS-DYNA Conference, Salzburg, Austria, May 2017.

Erhart, T., "A New Feature to Model Shell-Like Structures with Stacked Elements", 10th European LS-DYNA Conference, Würzburg, Germany, June 2015.

Erhart, T., "An Overview of User Defined Interfaces in LS-DYNA", LS-DYNA Forum, Bamberg, Germany, 2010.

Englemann, B. E., R.G. Whirley, and G.L. Goudreau, "A Simple Shell Element Formulation for Large-Scale Elastoplastic Analysis," CED-Vol. 3. Analytical and Computational Models of Shells, A.K. Noor, T. Belytschko, and J.C. Simo, Editors, 1989, pp. 399-416.

Faßnacht, W., "Simulation der Rißbildung in Aluminiumgußbauteilen," Dissertation, Technishe Universität Darmstadt, (1999).

Feng, W.W. and Hallquist, J.O., "On Constitutive Equations for Elastomers and Elastomeric Foams", The 4th European LS-DYNA Conference, D-II-15, Ulm, Germany, May 2003.

Feucht, M., "Ein gradientenabhängiges Gursonmodell zur Beshreibung duktiler Schädigung mit Entfestigung," Dissertation, Technishe Universität Darmstadt, (1998).

Fiolka, M. and Matzenmiller, A., "Delaminationsberechnung von Faserverbundstrukturen", *PAMM Proc. Appl. Math. Mech.* 5, S.393-394 (2005).

Flanagan, D.P. and T. Belytschko, "A Uniform Strain Hexahedron and Quadrilateral and Orthogonal Hourglass Control," *Int. J. Numer. Meths. Eng.*, 17, 679-706 (1981).

Fleischer, M., Borrvall, T. and Bletzinger, K-U., "Experience from using recently implemented enhancements for Material 36 in LS-DYNA 971 performing a virtual tensile test", 6th European LS-DYNA Users Conference, Gothenburg, 2007.

Forghani A., "A Non-Local Approach to Simulation of Damage in Composite Structures", *PhD Thesis, Department of Civil Engineering, The University of British Columbia,, Vancouver, Canada,* (2011).

Forghani A., Zobeiry N., Vaziri R., Poursartip A., and Ellyin F., "A Non-Local Approach to Simulation of Damage in Laminated Composites." *Proc., ASC/CANCOM Conference,* Montreal, Canada (2011b).

Forghani A., Zobeiry N., Poursartip A., and Vaziri R., "A Structural Modeling Framework for Prediction of Damage Development and Failure of Composite Laminates". Accepted for publication in *Composites Sci. Technol.*

Freed AD., Einstein DR. and Vesely I., "Invariant Formulation for Dispersed Transverse Isotropy in Aortic Heart Valves – An Efficient Means for Modeling Fiber Splay", *Biomechan Model Mechanobiol*, 4, 100-117 (2005)

Fung, Y.C., Biomechanics, Springer, New York, 1993.

Fung, Y.C., Foundations of Solid Mechanics, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1965.

Gerlach, S., Fiolka M. and Matzenmiller, A., Modelling and analysis of adhesively bonded joints with interface elements for crash analysis, 4. LS-DYNA Forum, 20-21, (2005) Bamberg, DYNAmore GmbH, Stuttgart.

Ginsberg, M. and J. Johnson, "Benchmarking the Performance of Physical Impact Simulation Software on Vector and Parallel Computers," Applications Track of Supercomputing, IEEE monograph, Computer Society Press, March, 1989.

Giroux, E.D. HEMP User's Manual, University of California, Lawrence Livermore National Laboratory, Rept. UCRL-51079 (1973).

Goldberg, R., and D. Stouffer, "High Strain Rate Dependent Modeling Polymer Matrix Composites," NASA/TM-1999-209433 (1999).

Goudreau, G.L. and J.O. Hallquist, "Recent Developments in Large Scale Finite Element Lagrangian Hydrocode Technology," *J. Comp. Meths. Appl. Mechs. Eng.*, 30 (1982).

Goldak, J., Chakravarti, A., and Bibby, M., "A New Finite Element Model for Welding Heat Sources," *Metallurgical Transactions B*, vol. 15B, pp. 299-305, June, 1984.

Govindjee, S., Kay, J.G., and Simo, J.C. [1994], Anisotropic Modeling and Numerical Simulation of Brittle Damage in Concrete, Report No. UCB/SEMM-94/18, Department of Civil Engineering, University of California, Berkeley, CA 94720.

Govindjee, S., Kay, J.G., and Simo, J.C. [1995], "Anisotropic Modeling and Numerical Simulation of Brittle Damage in Concrete," *Int. J. Numer. Meth. Engng*, **38**, 3611-3633.

Graefe, H., W. Krummheuer, and V. Siejak, "Computer Simulation of Static Deployment Tests for Airbags, Air Permeability of Uncoated Fabrics and Steady State Measurements of the Rate of Volume Flow Through Airbags," SAE Technical Paper Series, 901750, Passenger Car Meeting and Expositition, Dearborn, Michigan, September 17-20, 1990.

Gran, J.K. and P.E. Senseny, "Compression Bending of Scale-Model Reinforced-Concrete Walls," *ASCE Journal of Engineering Mechanics*, Volume 122, Number 7, pages 660-668, July (1996).

Grassl, P., U. Nyström, R. Rempling, and K. Gylltoft, "A Damage-Plasticity Model for the Dynamic Failure of Concrete", 8th International Conference on Structural Dynamics, Leuven, Belgium, March (2011).

Grassl, P., D. Xenos, U. Nyström, R. Rempling, and K. Gylltoft, "CDPM2: A damage-plasticity approach to modelling the failure of concrete", *International Journal of Solids and Structures*, Vol. 50, Issue 24, pp. 3805-3816, (2013).

Grassl, P. and M. Jirásek, "Damage-Plastic Model for Concrete Failure", *International Journal of Solids and Structures*, Vol. 43, Issues 22-23, pp. 7166-7196, November (2006).

Guccione, J., A. McCulloch, and L. Waldman, "Passive Material Properties of Intact Ventricular Myocardium Determined from a Cylindrical Model", *ASME Journal of Biomechanical Engineering*, Vol. 113, pages 42-55, (1991).

Guccione JM, Waldman LK, McCulloch AD., "Mechanics of Active Contraction in Cardiac Muscle: Part II – Cylindrical Models of the Systolic Left Ventricle", J. Bio Mech, 115, 82-90, (1993).

Gurson, A.L., Plastic Flow and Fracture Behavior of Ductile Materials Incorporating Void Nucleation, Growth, and Interaction, Ph.D. Thesis, Brown University, (1975).

Gurson, A.L., “Continuum Theory of Ductile Rupture by Void Nucleation and Growth: Part I - Yield Criteria and Flow Rules for Porous Ductile Media”, *J. of Eng. Materials and Technology*, (1977).

Hallquist, J.O., Preliminary User’s Manuals for DYNA3D and DYNAP (Nonlinear Dynamic Analysis of Solids in Three Dimension), University of California, Lawrence Livermore National Laboratory, Rept. UCID-17268 (1976) and Rev. 1 (1979).[**a**]

Hallquist, J.O., A Procedure for the Solution of Finite Deformation Contact-Impact Problems by the Finite Element Method, University of California, Lawrence Livermore National Laboratory, Rept. UCRL-52066 (1976).

Hallquist, J.O., “A Numerical Procedure for Three-Dimensional Impact Problems,” *American Society of Civil Engineering*, Preprint 2956 (1977).

Hallquist, J.O., “A Numerical Treatment of Sliding Interfaces and Impact,” in: K.C. Park and D.K. Gartling (eds.) *Computational Techniques for Interface Problems*, AMD Vol. 30, ASME, New York (1978).

Hallquist, J.O., NIKE2D: An Implicit, Finite-Element Code for Analyzing the Static and Dynamic Response of Two-Dimensional Solids, University of California, Lawrence Livermore National Laboratory, Rept. UCRL-52678 (1979).[**b**]

Hallquist, J.O., User's Manual for DYNA2D – An Explicit Two-Dimensional Hydrodynamic Finite Element Code with Interactive Rezoning, University of California, Lawrence Livermore National Laboratory, Rept. UCID-18756 (1980).

Hallquist, J.O., User's Manual for DYNA3D and DYNAP (Nonlinear Dynamic Analysis of Solids in Three Dimensions), University of California, Lawrence Livermore National Laboratory, Rept. UCID-19156 (1981).[**a**]

Hallquist, J. O., NIKE3D: An Implicit, Finite-Deformation, Finite-Element Code for Analyzing the Static and Dynamic Response of Three-Dimensional Solids, University of California, Lawrence Livermore National Laboratory, Rept. UCID-18822 (1981).[**b**]

Hallquist, J.O., DYNA3D User's Manual (Nonlinear Dynamic Analysis of Solids in Three Dimensions), University of California, Lawrence Livermore National Laboratory, Rept. UCID-19156 (1982; Rev. 1: 1984; Rev. 2: 1986).

Hallquist, J.O., Theoretical Manual for DYNA3D, University of California, Lawrence Livermore National Laboratory, Rept. UCID-19501 (March, 1983).

Hallquist, J.O., DYNA3D User's Manual (Nonlinear Dynamic Analysis of Solids in Three Dimensions), University of California, Lawrence Livermore National Laboratory, Rept. UCID-19156 (1988, Rev. 4).

Hallquist, J.O., LS-DYNA User's Manual (Nonlinear Dynamic Analysis of Solids in Three Dimensions), Livermore Software Technology Corporation, Rept. 1007 (1990).

Hallquist, J.O., D.J. Benson, and G.L. Goudreau, "Implementation of a Modified Hughes-Liu Shell into a Fully Vectorized Explicit Finite Element Code," Proceedings of the International Symposium on Finite Element Methods for Nonlinear Problems, University of Trondheim, Trondheim, Norway (1985).

Hallquist, J.O. and D.J. Benson, "A Comparison of an Implicit and Explicit Implementation of the Hughes-Liu Shell," Finite Element Methods for Plate and Shell Structures, T.J.R. Hughes and E. Hinton, Editors, 394-431, Pineridge Press Int., Swanea, U.K. (1986).

Hallquist, J.O. and D.J. Benson, DYNA3D User's Manual (Nonlinear Dynamic Analysis of Solids in Three Dimensions), University of California, Lawrence Livermore National Laboratory, Rept. UCID-19156 (Rev. 2: 1986; Rev. 3: 1987).

Hallquist, J.O., D.W. Stillman, T.J.R. Hughes, C. and Tarver,"Modeling of Airbags Using MVMA/DYNA3D," LSTC Report (1990).

Hashin, Z, "Failure Criteria for Unidirectional Fiber Composites," *Journal of Applied Mechanics*, 47, 329 (1980).

Haßler, M., Schweizerhof, K., "On the influence of fluid-structure-interaction on the static stability of thin walled shell structures", *International Journal of Structural Stability*, Vol. 7, pp. 313–335, (2007).

Haßler, M., Schweizerhof, K., "On the static interaction of fluid and gas loaded multi-chamber systems in a large deformation finite element analysis", *Computer Methods in Applied Mechanics and Engineering*, Vol. 197, pp. 1725–1749, (2008).

Hänsel, C., P. Hora, and J. Reissner, "Model for the Kinetics of Strain-Induced Martensitic Phase Transformation at Isothermal Conditions for the Simulation of Sheet Metal Forming Processes with Metastable Austenitic Steels," Simulation of Materials Processing: Theory, Methods, and Applications, Huétink and Baaijens (eds), Balkema, Rotterdam, (1998).

Haward, R.N., and Thackray, G., "The use of a mathematical model to describe isothermal stress-strain curves in glassy thermoplastics". *Proc Roy Soc A*, 302, 453-472 (1968).

Herrmann, L.R. and F.E. Peterson, "A Numerical Procedure for Viscoelastic Stress Analysis," Seventh Meeting of ICRPG Mechanical Behavior Working Group, Orlando, FL, CPIA Publication No. 177, 1968.

Hill A.V., “The heat of shortening and the dynamic constants of muscle,” *Proc Roy Soc* B126:136-195, (1938).

Hill, R., “A Theory of the Yielding and Plastic Flow of Anisotropic Metals,” *Proceedings of the Royal Society of London*, Series A., Vol. 193, pp. 281-197 (1948).

Hill, R., “Aspects of Invariance in Solid Mechanics,” *Advances in Applied Mechanics*, Vol. 18, pp. 1-75 (1979).

Hill, R., “Constitutive Modeling of Orthotropic Plasticity in Sheet Metals,” *J. Mech. Phys. Solids*, Vol. 38, No. 3, 1989, pp. 405-417.

Hippchen, P., Merklein, M., Lipp, A., Fleischer, M., Grass, H., Craighero, P., “Modelling kinetics of phase transformation for the indirect hot stamping process”, *Key Engineering Materials*, Vol. 549, pages 108-116, (2013).

Hippchen, P., “Simulative Prognose der Geometrie indirekt pressgehärteter Karosseriebauteile für die industrielle Anwendung”, Dissertation, Technische Fakultät der Friedrich-Alexander Universität Erlangen-Nürnberg, Meisenbach Verlag Bamberg, Band 249, (2014).

Hirth, A., P. Du Bois, and K. Weimar, “Improvement of LS-DYNA Material Law 83 (Fu Chang) for the Industrial Simulation of Reversible Energy-Absorbing Foams,” CAD-FEM User's Meeting, Bad Neuenahr - Ahrweiler, Germany, October 7-9, Paper 2-40, (1998).

Hollenstein M., M. Jabareen, and M.B. Rubin, “Modelling a smooth elastic-inelastic transition with a strongly objective numerical integrator needing no iteration”, Comput. Mech., Vol. 52, pp. 649-667, DOI 10.1007/s00466-013-0838-7 (2013).

Hollenstein M., M. Jabareen, and M.B. Rubin, “Erratum to: Modelling a smooth elastic-inelastic transition with a strongly objective numerical integrator needing no iteration”, Comput. Mech., DOI 10.1007/s00466-014-1099-9 (2014).

Holmquist, T.J., G.R. Johnson, and W.H. Cook, “A Computational Constitutive Model for Concrete Subjected to Large Strains, High Strain Rates, and High Pressures”, Proceedings 14th International Symposium on Ballistics, Quebec, Canada, pp. 591-600, (1993).

Hopperstad, O.S. and Remseth, S.,” A return Mapping Algorithm for a Class of Cyclic Plasticity Models”, *International Journal for Numerical Methods in Engineering*, Vol. 38, pp. 549-564, (1995).

Huang, Yuli, private communication, Livermore, (2006).

Hughes, T.J.R. and E. Carnoy, “Nonlinear Finite Element Shell Formulation Accounting for Large Membrane strains,” *AMD*-Vol.48, ASME, 193-208 (1981).

Hughes, T.J.R. and W.K. Liu, "Nonlinear Finite Element Analysis of Shells: Part I. Three-Dimensional Shells." *Comp. Meths. Appl. Mechs.*, 27, 331-362 (1981a).

Hughes, T.J.R. and W.K. Liu, "Nonlinear Finite Element Analysis of Shells: Part II. Two-Dimensional Shells." *Comp. Meths. Appl. Mechs.*, 27, 167-181 (1981b).

Hughes, T.J.R., W.K. Liu, and I. Levit, "Nonlinear Dynamics Finite Element Analysis of Shells." Nonlinear Finite Element Analysis in Struct. Mech., Eds. W. Wunderlich, E. Stein, and K.J. Bathe, Springer-Verlag, Berlin, 151- 168 (1981c).

Huh, H., Chung, K., Han, S.S., and Chung, W.J., "The NUMISHEET 2011 Benchmark Study of the 8th International Conference and Workshop on Numerical Simulation of 3D Sheet Metal Forming Processes, Part C Benchmark Problems and Results", p 171-228, Seoul, Korea, August, 2011.

Huh, H. and Kang, W.J., "Crash-Worthiness Assessment of Thin-Walled Structures with the High-Strength Steel Sheet", *Int. Journal of Vehicle Design*, Vol. 30, Nos. 1/2 (2002).

Ibrahimbegovic, A. and Wilson, E.L. "A unified formulation for triangular and quadrilateral flat shell finite elements with six nodal degrees of freedom", Comm. Applied Num. Meth, 7, 1-9 (1991).

Isenberg, J., Vaughan, D.K., Sandler, I.S., Nonlinear Soil-Structure Interaction, Electric Power Research Institute report EPRI NP-945, Weidlinger Associates (1978).

Ivanov, I., and A. Tabiei, "Loosely Woven Fabric Model With Viscoelastic Crimped Fibers for Ballistic Impact Simulations", IJNME, 57, (2004).

Jabareen, M., "Strongly objective numerical implementation and generalization of a unified large inelastic deformation model with a smooth elastic-inelastic transition", submitted to Int. J. Solids and Struct. (2015).

Jabareen, M., and Rubin, M.B., A Generalized Cosserat Point Element (CPE) for Isotropic Nonlinear Elastic Materials including Irregular 3-D Brick and Thin Structures, J. Mech. Mat. And Struct., Vol 3-8, 1465-1498 (2008).

Jabareen, M., Hanukah, E. and Rubin, M.B., A Ten Node Tetrahedral Cosserat Point Element (CPE) for Nonlinear Isotropic Elastic Materials, J. Comput. Mech. 52, 257-285 (2013).

Johnson, G.C. and D.J. Bammann, "A discussion of stress rates in finite deformation problems," *Int. J. Solids Struct*, 20, 725-737 (1984).

Johnson, G.R. and W.H. Cook, "A Constitutive Model and Data for Metals Subjected to Large Strains, High Strain Rates and High Temperatures." Presented at the

Seventh International Symposium on Ballistics, The Hague, The Netherlands, April 1983.

Johnson, G.R. and T.J. Holmquist, "An Improved Computational Model for Brittle Materials" in High-Pressure Science and Technology - 1993 American Institute of Physics Conference Proceedings 309 (c 1994) pp.981-984 ISBN 1-56396-219-5.

Jones, R.M., Mechanics of Composite Materials, Hemisphere Publishing Corporation, New York, (1975).

Kenchington, G.J., "A Non-Linear Elastic Material Model for DYNA3D," Proceedings of the DYNA3D Users Group Conference, published by Boeing Computer Services (Europe) Limited (1988).

Key, S.W. HONDO – A Finite Element Computer Program for the Large Deformation Dynamic Response of Axisymmetric Solids, Sandia National Laboratories, Albuquerque, N.M., Rept. 74-0039 (1974).

Kolling, S., Haufe, A., Feucht, M., DuBois, P. A. "SAMP-1: A Semi-Analytical Model for the Simulation of Polymers", 4. LS-DYNA Anwenderforum, October 20-21, Bamberg, Germany, (2005).

Kolling, S., Hirth, A., Erhart, and Du Bois P.A., Private Communication, Livermore, California (2006).

Krieg, R.D.,A Simple Constitutive Description for Cellular Concrete, Sandia National Laboratories, Albuquerque, NM, Rept. SC-DR-72-0883 (1972).

Krieg, R.D. and S.W. Key, "Implementation of a Time Dependent Plasticity Theory into Structural Computer Programs," Vol. 20 of Constitutive Equations in Viscoplasticity: Computational and Engineering Aspects (American Society of Mechanical Engineers, New York, N.Y., pp. 125-137 (1976).

Lademo, O.G., Berstad, T., Tryland, T., Furu, T., Hopperstad, O.S. and Langseth, M., "*A model for process-based crash simulation*", 8th International LS-DYNA User's Conference, Detroit, May 3-5, 2004.

Lademo, O.G., Hopperstad, O.S., Berstad, T. and Langseth M., "Prediction of Plastic Instability in Extruded Aluminum Alloys Using Shell Analysis and a Coupled Model of Elasto-plasticity and Damage," Journal of Materials Processing Technology, 2002 (Article in Press).

Lademo, O.G., Hopperstad, O.S., Malo, K.A. and Pedersen, K.O., "Modelling of Plastic Anisotropy in Heat-Treated Aluminum Extrusions", Journal of Materials Processing Technology **125-126**, pp. 84-88 (2002).

Lee, E.L. and C.M. Tarver, "Phenomenological Model of Shock Initiation in Heterogenous Explosives," PHYS. Fluids, Vol. 23, p. 2362 (1980).

Lee, Y. L. and S Balur of Chrysler Group LLC, "Method of predicting spot weld failure" (Attorney Docket No 708494US2), filed on December 22, 2011 and assigned US Serial No. 13/334,701.

Lemaitre, J., A Course on Damage Mechanics, Springer-Verlag, (1992).

Lemaitre, J., and Chaboche, J.L., Mechanics of Solid Materials, Cambridge University Press, (1990).

Lemmen, P. P. M. and Meijer, G. J., "Failure Prediction Tool Theory and User Manual," TNO Report 2000-CMC-R0018, (2001).

Lewis, B.A., "Developing and Implementing a Road Side Safety Soil Model into LS-DYNA," FHWA Research and Development Turner-Fairbank Highway Research Center, (1999).

Li, Y.H. and Sellars, C.M., "Modeling Deformation Behavior of Oxide Scales and their Effects on Interfacial Heat Transfer and Friction during Hot Steel Rolling", Proc. Of the 2nd Int. Conf. Modeling of Metals Rolling Processes, The Insitute of Materials, Londong, UK, 192-201 (1996).

Lindström P.R.M, "DNV Platform of Computational Welding Mechanics", Proc. of Int. Inst. Welding 66th Annual Assembly (2013).

Lindström, P., "Improved CWM platform for modelling welding procedures and their effects on structural behaviour", PhD Thesis, Production Technology, University West, Trollhättan, Sweden (2015).

Lian, W., personal communication: "LS-DYNA Airbag Module Improvement Request", General Motors Corporation (2000).

Y. Luo, "An Efficient 3D Timoshenko Beam Element with Consistent Shape Functions" *Adv. Theor. Appl. Mech.,* 1(3), 95-106, (2008).

MADYMO3D USER'S MANUAL, Version 4.3, TNO Road-Vehicles Research Institute, Department of Injury Prevention, The Hague, The Netherlands, (1990).

Maimí, P., Camanho, P.P., Mayugo, J.A., Dávila, C.G., "A continuum damage model for composite laminates: Part I – Constitutive model", *Mechanics of Materials*, Vol. 39, pp. 897–908, (2007).

Maimí, P., Camanho, P.P., Mayugo, J.A., Dávila, C.G., "A continuum damage model for composite laminates: Part II – Computational implementation and validation", *Mechanics of Materials*, Vol. 39, pp. 909–919, (2007).

Maker, B.N., Private communication Lawrence Livermore National Laboratory, Dr. Maker programmed and implemented the compressible Mooney Rivlin rubber model (1987).

Makris N. and Zhang, J., “Time-domain visco-elastic analysis of earth structures,” *Earthquake Engineering and Structural Dynamics*, vol. 29, pp. 745–768, (2000).

Malvar, L.J., Crawford, J.E., Morrill, K.B., K&C Concrete Material Model Release III — Automated Generation of Material Model Input, K&C Technical Report TR-99-24-B1, 18 August 2000 (*Limited Distribution*).

Malvar, L.J., Crawford, J.E., Wesevich, J.W., Simons, D., “A Plasticity Concrete Material Model for DYNA3D,” *International Journal of Impact Engineering*, Volume 19, Numbers 9/10, pages 847-873, December 1997.

Malvar, L.J., and Ross, C.A., “Review of Static and Dynamic Properties of Concrete in Tension,” *ACI Materials Journal*, Volume 95, Number 6, pages 735-739, November-December 1998.

Malvar, L.J., and Simons,D., “Concrete Material Modeling in Explicit Computations,” Proceedings, Workshop on Recent Advances in Computational Structural Dynamics and High Performance Computing, USAE Waterways Experiment Station, Vicksburg, MS, pages 165-194, April 1996. (LSTC may provide this reference upon request.)

Malvar, H.S., Sullivan, G.S., and Wornell, G.W., “Lapped Orthogonal Vector Quantization”, in Proc. Data Compression Conference, Snowbird, Utah, 1996.

Marin, E.B., unpublished paper, Sandia National Laboratory, CA (2005).

Matzenmiller, A. and Burbulla, F., “ Robustheit und Zuverlässigkeit der Berechnungsmethoden von Klebverbindungen mit hochfesten Stahlblechen unter Crashbedingungen” (2013), http://www.ifm.maschinenbau.uni-kassel.de/~amat/publikationen/p75_p828-modellerweiterung.pdf

Matzenmiller, A., Lubliner, J., and Taylor, R.L., “A Constitutive Model for Anisotropic Damage in Fiber-Composites,” Mechanics of Materials, Vol. 20, pp. 125-152 (1995).

Matzenmiller, A. and J. K. Schweizerhof, “Crashworthiness Considerations of Composite Structures – A First Step with Explicit Time Integration in Nonlinear Computational Mechanics–State-of-the-Art,” Ed. P. Wriggers, W. Wagner, Springer Verlay, (1991).

Mauldin, P.J., R.F. Davidson, and R.J. Henninger, “Implementation and Assessment of the Mechanical-Threshold-Stress Model Using the EPIC2 and PINON Computer Codes,” Report LA-11895-MS, Los Alamos National Laboratory (1990).

Maurer, A., Gebhardt, M., Schweizerhof, K., "Computation of fluid and/or gas filled inflatable dams“, LS-Dyna Forum, Bamberg, Germany, (2010).

McCormick, P.G., "Theory of flow localization due to dynamic strain ageing," *Acta Metallurgica*, 36, 3061-3067 (1988).

Mi Y., Crisfield, M.A., Davies, A.O. Progressive delamination using interface elements. *J Compos Mater*, 32(14)1246-72 (1998).

Moran, B., Ortiz, M. and Shih, C.F., "Formulation of implicit finite element methods for multiplicative finite deformation plasticity". *Int J for Num Methods in Engineering*, 29, 483-514 (1990).

de Moura MFSF, Gonçalves, J.P., Marques, A.T., and de Castro, P.T., Elemento finito isoparamétrico de interface para problemas tridimensionais. *Revista Internacional de Métodos Numéricos Para Cálculo e Diseño en Ingeniería*, 14:447-66 (1996).

Murray, Y.D., Users Manual for Transversely Isotropic Wood Model APTEK, Inc., Technical Report to the FHWA (to be published) (2002).

Murray, Y.D. and Lewis, B.A., Numerical Simulation of Damage in Concrete APTEK, Inc., Technical Report DNA-TR-94-190, Contract DNA 001-91-C-0075, Defense Nuclear Agency, Alexandria VA 22310.

Murray, Y.D., Users Manual for LS-DYNA Concrete Material Model 159, Report No. FHWA-HRT-05-062, Federal Highway Administration, (2007).

Murray, Y.D., A. Abu-Odeh, and R. Bligh, Evaluation of Concrete Material Model 159, Report No. FHWA-HRT-05-063, Federal Highway Administration, (2007).

Muscolini, G., Palmeri, A. and Ricciardelli, F., "Time-domain response of linear hysteretic systems to deterministic and random excitations," *Earthquake Engineering and Stuctrual Dynamics*, vol. 34, pp. 1129–1147, (2005).

Nagararaiah, Reinhorn, & Constantinou, "Nonlinear Dynamic Analysis of 3-D Base-Isolated Structures", *Jounal of Structural Engineering* Vol 117, No 7, (1991).

Nahshon, K. and Hutchinson, J.W., "Modification of the Gurson Model for shear failure", *European Journal of Mechanics A/Solids*, Vol. 27, 1-17, (2008).

Naik D., Sankaran S., Mobasher D., Rajan S.D., Pereira J.M., "Development of reliable modeling methodologies for fan blade out containment analysis – Part I: Experimental studies", International Journal of Impact Engineering, Volume 36, Issue 1, Pages 1-11, (2009)

Neal, M.O., C-H Lin, and J. T. Wang, "Aliasing Effects on Nodal Acceleration Output from Nonlinear Finite Element Simulations," ASME 2000 International

Mechanical Engineering Congress and Exposition, Orlando, Florida, November 5-10, (2000).

Neilsen, M.K., H.S. Morgan, and R.D. Krieg, "A Phenomenological Constitutive Model for Low Density Polyurethane Foams," Rept. SAND86-2927, Sandia National Laboratories, Albuquerque, N.M., (1987).

Nusholtz, G., W. Fong, and J. Wu, "Air Bag Wind Blast Phenomena Evaluation," *Experimental Techniques*, Nov.-Dec. (1991).

Nusholtz, G., D. Wang, and E.B. Wylie, "Air Bag Momentum Force Including Aspiration," Preprint, Chrysler Corporation, (1996).

Nusholz, private communication, (1996).

Nygards, M., M. Just, and J. Tryding, "Experimental and numerical studies of creasing of paperboard," *Int. J. Solids and Struct.*, 46, 2493-2505 (2009).

Ogden, R.W., Non-Linear Elastic Deformations, Ellis Horwood Ltd., Chichester, Great Britian (1984).

Oliver, J., "A Consistent Characteristic Length of Smeared Cracking Models," *International Journal for Numerical Methods in Engineering*, **28**, 461-474 (1989).

Papadrakakis, M., "A Method for the Automatic Evaluation of the Dynamic Relaxation Parameters," *Comp. Meth. Appl. Mech. Eng.*, Vol. 25, pp. 35-48 (1981).

Park, R. and Paulay, T., (1975) Reinforced Concrete Structures, J. Wiley and Sons, New York.

Park, Y.J., Wen, Y.K, and Ang, A.H-S, "Random Vibration of Hysteretic Systems Under Bi-directional Ground Motions", *Earthquake Engineering and Structural Dynamics*, Vol. 14, pp. 543-557 (1986).

Penelis, G.G. and Kappos, A.J., Earthquake-Resistant Concrete Structures, E&FN Spon., (1997).

Pijaudier-Cabot, G., and Bazant, Z.P., "Nonlocal Damage Theory," *Journal of Engineering Mechanics*, ASCE, Vol. 113, No. 10, 1512-1533 (1987).

Pinho, S.T., Iannucci, L., Robinson, R., "Physically-based failure models and criteria for laminated fibre-reinforced composites with emphasis on fibre kinking: Part I: Development", *Composties Part A*, Vol. 37, 63-73 (2006).

Pinho, S.T., Iannucci, L., Robinson, R., "Physically-based failure models and criteria for laminated fibre-reinforced composites with emphasis on fibre kinking: Part II: FE implementation", *Composties Part A*, Vol. 37, 766-777 (2006).

Porcaro, R., A.G. Hanssen, A. Aalberg and M. Langseth, "The behaviour of aself-piercing riveted connection under quasi-static loading conditions," *Int. J. Solids and Structures,* Vol. 43/17, pp. 5110-5131 (2006).

Porcaro, R., A.G. Hanssen, A. Aalberg and M. Langseth, "Self-piercing riveting process, an experimental and numerical investigation," *Journal of Materials processing Technology*, Vol. 171/1, pp. 10-20 (2006).

Porcaro, R., M. Langseth, A.G. Hanssen, H. Zhao, S. Weyer and H. Hooputra, "Crashworthiness of self-piercing riveted connections," *International Journal of Impact Engineering*, In press, Accepted manuscript (2007).

Puck, A., Kopp, J., Knops, M., "Guidelines for the determination of the parameters in Puck's action plane strength criterion", *Composites Science and Technology*, Vol. 62, 371-378 (2002).

Puso, M.A., "A Highly Efficient Enhanced Assumed Strain Physically Stabilized Hexahedral Element", *Int. J. Numer. Meth. Eng.*, Vol. 49, 1029-1064 (2000).

Puso, M.A., and Laursen, T.A., "A Mortar segment-to-segment contact method for large deformation solid mechanics", *Comput. Methods Appl. Mech. Engrg*. 193 (2004) 601-629.

Puso, M.A., and Laursen, T.A., "A Mortar segment-to-segment frictional contact method for large deformations", *Comput. Methods Appl. Mech. Engrg*. 193 (2004) 4891-4913.

Puso, M.A. and Weiss, J.A., "Finite Element Implementation of Anisotropic Quasilinear Viscoelasticity Using a Discrete Spectrum Approximation", *ASME J. Biomech. Engng.*, 120, 62-70 (1998).

Pelessone, D., Private communication, GA Technologies, P.O. Box 85608, San Diego, CA., Telephone No. 619-455-2501 (1986).

Quapp, K.M. and Weiss, J.A., "Material Characterization of Human Medial Collateral Ligament", *ASME J. Biomech Engng.*, 120, 757-763 (1998).

Reyes, A., O.S. Hopperstad, T. Berstad, and M. Langseth, Implementation of a Material Model for Aluminium Foam in LS-DYNA, Report R-01-02, Restricted, Department of Structural Engineering, Norwegian University of Science and Technology, (2002).

Randers-Pehrson, G. and K. A. Bannister, Airblast Loading Model for DYNA2D and DYNA3D, Army Research Laboratory, Rept. ARL-TR-1310, publicly released with unlimited distribution, (1997).

Richards, G.T., Derivation of a Generalized Von Neuman Psuedo-Viscosity with Directional Properties, University of California, Lawrence Livermore National Laboratory, Rept. UCRL-14244 (1965).

Riedel W., Thoma K., Hiermaier S. and Schmolinske E., "Penetration of reinforced concrete by BETA-B-500", in Proc. 9. ISIEMS, Berlin Strausberg, Mai (1999).

Riedel W., "Beton unter Dynamischen Lasten – Meso- und Makromechanische Modelle" In: Ernst-Mach-Institut, editor. Freiburg: Fraunhofer IRB, ISBN 3-8167-6340-5; 2004.

Ritto-Corrêa M. and Camotim D., "On the arc-length and other quadratic control methods: Established, less known and new implementation procedures", *Comput. Struct.*, 86, pp. 1353-1368 (2008).

Roussis, P.C., and Constantinou, M.C., "Uplift-restraining Friction Pendulum seismic isolation system", *Earthquake Engineering and Structural Dynamics*, 35 (5), 577-593, (2006).

Rumpel, T., Schweizerhof, K., "Volume-dependent pressure loading and its influence on the stability of structures", *International Journal for Numerical Methods in Engineering*, Vol. 56, pp. 211-238, (2003).

Rumpel, T., Schweizerhof, K., "Hydrostatic fluid loading in non-linear finite element analysis", *International Journal for Numerical Methods in Engineering*, Vol. 59, pp. 849–870, (2004).

Rupp, A., Grubisic, V., and Buxbaum, O., Ermittlung ertragbarer Beanspruchungen am Schweisspunkt auf Basis der ubertragbaren Schnittgrossen, FAT Schriftenreihe 111, Frankfurt (1994).

Sala, M.O. Neal, and J.T. Wang, Private Communication, General Motors, May, 2004.

Sackett, S.J., "Geological/Concrete Model Development," Private Communication (1987).

Sandler, I.S. and D. Rubin, "An Algorithm and a Modular Subroutine for the Cap Model," *Int. J. Numer. Analy. Meth. Geomech.*, 3, pp. 173-186 (1979).

Schedin, E., Prentzas, L. and Hilding D., "Finite Element Simulation of the TRIP-effect in Austenitic Stainless Steel," presented at SAE 2004, SAE Technical paper 2004-01-0885, (2004).

Schulte-Frankenfeld N., Deiters, T., "General Introduction to FATXML – data format", https://www.vda.de/dam/vda/publications/FATXML-Format%20Version%20V1.1/1305643077_de_2066216985.zip, (2016).

Schweizerhof, K. and E. Ramm, "Displacement Dependent Pressure Loads in Nonlinear Finite Element Analyses," *Comput. Struct.*, 18, pp. 1099-1114 (1984).

Schwer, L.E., "A Viscoplastic Augmentation of the Smooth Cap Model," *Nuclear Engineering and Design*, Vol. 150, pp. 215-223, (1994).

Schwer, L.E., "Demonstration of the Continuous Surface Cap Model with Damage: Concrete Unconfined Compression Test Calibration," LS-DYNA Geomaterial Modeling Short Course Notes, July (2001).

Schwer, L.E., W. Cheva, and J.O. Hallquist, "A Simple Viscoelastic Model for Energy Absorbers Used in Vehicle-Barrier Impact," in Computation Aspects of Contact, Impact, and Penetration, Edited by R.F. Kulak and L.E. Schwer, Elmepress International, Lausanne, Switzerland, pp. 99-117 (1991).

Schwer, L.E. and Y.D. Murray, "A Three-Invariant Smooth Cap Model with Mixed Hardening," *International Journal for Numerical and Analytical Methods in Geomechanics*, Volume 18, pp. 657-688, (1994).

Seeger, F., M. Feucht, T. Frank (DaimlerChrysler AG), and A. Haufe, B. Keding (DYNAmore GmbH), "An Investigation on Spotweld Modeling for Crash Simulation with LS-DYNA", 4th LS-DYNA-Forum, Bamburg, Germany, October (2005), ISBN 3-9809901-1-7.

Shi, M.F., and Gelisse, S., "Issues on the AHSS Forming Limit Determination", International Deep Drawing Research Group (IDDRG), Porto, Portugal, June, 2006.

Shi, M.F., Zhu, X.H., Xia, C., and Stoughton T., "Determination of Nonlinear Isotropic/Kinematic Hardening Constitutive Parameter for AHSS using Tension and Compression Tests", p. 137-142, Proceedings of the 7th International Conference and Workshop on Numerical Simulation of 3D Sheet Metal Forming Processes (NUMISHEET 2008), Interlaken, Switzerland, September, 2008.

Sheppard, S.D., Estimations of Fatigue Propagation Life in Resistance Spot Welds, ASTM STP 1211, pp. 169-185, (1993).

Sheppard, T. and Wright, D.S., "Determination of flow stress: Part 1 constitutive equation for aluminum alloys at elevated temperatures", *Metals Technology*, p. 215, June 1979.

Shvets, I.T. and Dyban, E., P., "Contact Heat Transfer between Plane Metal Surfaces", *Int. Chem. Eng.*, Vol. 4, No. 4, 621 (1964).

Simo, J.C., J.W. Ju, K.S. Pister, and R.L. Taylor, "An Assessment of the Cap Model: Consistent Return Algorithms and Rate-Dependent Extension," *J. Eng. Mech.*, Vol. 114, No. 2, 191-218 (1988a).

Simo, J.C., J.W. Ju, K.S. Pister, and R.L. Taylor, "Softening Response, Completeness Condition, and Numerical Algorithms for the Cap Model," *Int. J. Numer. Analy. Meth. Eng.*, (in press) (1988b).

Simo, J. C., J.W. Ju, K.S. Pister, and R.L. Taylor, "Softening Response, Completeness Condition, and Numerical Algorithms for the Cap Model," *Int. J. Numer. Analy. Meth. Eng.* (1990).

Solberg, J.M., and C.M. Noble, "Contact Algorithm for Small-Scale Surface Features with Application to Finite Element Analysis of Concrete Arch Dams with Beveled Contraction Joints", Lawrence Livermore National Laboratory (2002).

Spanos, P.D. and Tsavachidis, S., "Deterministic and stochastic analyses of a nonlinear system with a Biot visco-elastic element," *Earthquake Engineering and Structural Dynamics*, vol. 30, pp. 595–612, (2001).

Steinberg, D.J. and M.W. Guinan, A High-Strain-Rate Constitutive Model for Metals, University of California, Lawrence Livermore National Laboratory, Rept. UCRL-80465 (1978).

Steinberg, D.J. and C.M. Lund, "A Constitutive Model for Strain Rates form 10^{-4} to 10^{6} S^{-1}," *J. Appl. Phys*., 65, p. 1528 (1989).

Stillman, D.W. and J.O. Hallquist, INGRID: A Three-Dimensional Mesh Generator for Modeling Nonlinear Systems, University of California, Lawrence Livermore National Laboratory, Rept. UCID-20506. (1985).

Stojko, S., privated communication, NNC Limited, Engineering Development Center (1990).

Stahlecker Z., Mobasher B., Rajan S.D., Pereira J.M., "Development of reliable modeling methodologies for engine fan blade out containment analysis. Part II: Finite element analysis", International Journal of Impact Engineering, Volume 36, Issue 3, Pages 447-459, (2009)

Storakers, B., "On material representation and constitutive branching in finite compressible elasticity", J. Mech. Phy. Solids, 34 No. 2, 125-145 (1986).

Stouffer and Dame, Inelastic Deformation of Metals, Wiley, (1996).

Stout, M.G., D.E. Helling, T.L. Martin, and G.R. Canova, *Int. J. Plasticity*, Vol. 1, pp. 163-174, (1985).

Structural Engineers Association of California, Tentative Lateral Force Requirements, Seismology Committee, SEAOC, 1974, 1990, 1996.

Sussman, T. and Bathe, K.J., "A Finite Element Formulation for Nonlinear Incompressible Elastic and Inelastic Analysis," *Computers & Structures*, **26**, Number 1/2, 357-409 (1987).

Tabiei, A. and I. Ivanov, "Computational micro-mechanical Model of Flexible Woven Fabric for Finite Element Impact Simulation," IJNME, 53, (6), 1259-1276, (2002).

Tahoe User Guide, Sandia National Laboratory, can be downloaded from: www.sandia.gov, Input version 3.4.1, (2003).

Taylor, L.M. and D.P. Flanagan, PRONTO3D A Three-Dimensional Transient Solid Dynamics Program, Sandia Report: SAND87-1912, UC-32, (1989).

Taylor, R.L. Finite element analysis of linear shell problems, *in Whiteman, J.R. (ed.),* Proceedings of the Mathematics in Finite Elements and Applications, Academic Press, New York, 191-203, (1987).

Taylor, R.L. and Simo, J.C. Bending and membrane elements for the analysis of thick and thin shells, *Proc. of NUMETA Conference*, Swansea (1985).

Tsai, S.W. and E.M. Wu, "A General Theory of Strength for Anisotropic Materials," J. Composite Materials, 5, pp. 73-96 (1971).

Tuler, F.R. and B.M. Butcher, "A Criterion for the Time Dependence of Dynamic Fracture," *The International Journal of Fracture Mechanics*, Vol. 4, No. 4, (1968).

Tvergaard, V. and J.W. Hutchinson, "The relation between crack growth resistance and fracture process parameters in elastic-plastic solids," *J. of the Mech. And Phy. of Solids*, 40, pp1377-1397, (1992)

Tvergaard, V. and Needleman, A., "Analysis of the cup-cone fracture in a round tensile bar", *Acta Metallurgica*, 32, 157-169 (1984).

Vawter, D., "A Finite Element Model for Macroscopic Deformation of the Lung," published in the *Journal of Biomechanical Engineering*, Vol.102, pp. 1-7 (1980).

VDA Richtlinier (Surface Interfaces), Version 20, Verband der Automobilindustrie e.v., Frankfurt, Main, Germany, (1987).

Vegter, H., Horn, C.H.L.J. ten, An, Y., Atzema, E.H., Pijlman, H.H., Boogaard, A.H. van den, Huetink, H., "Characterisation and Modelling of the Plastic Material Behaviour and its Application in Sheet Metal Forming Simulation", *Proceedings of 7th International Conference on Computational Plasticity COMPLAS VII*, Barcelona (2003).

Vegter, H., and Boogaard, A.H. van den, "A plane stress yield function for anisotropic sheet material by interpolation of biaxial stress states", *International Journal of Plasticity* 22, 557-580 (2006).

Walker, J.C., Ratcliffe M.B., Zhang P., Wallace A.W., Fata, B., Hsu E., Saloner D., and Guccione J.M. "MRI-based finite-element analysis of left ventricular aneurysm", Am J Physiol Heart Circ Physiol 289(2): H692:700 (2005).

Wang, J.T. and O.J. Nefske, "A New CAL3D Airbag Inflation Model," SAE paper 880654, 1988.

Wang, J.T., "An Analytical Model for an Airbag with a Hybrid Inflator", Publication R&D 8332, General Motors Development Center, Warren, Mi. (1995).

Wang, J.T., "An Analytical Model for an Airbag with a Hybrid Inflator", *AMD*-Vol. 210, *BED*-Vol. 30, ASME, pp 467-497, (1995).

Wang, K., Y. J. Chao, Y. J., X. Zhu., and K.W. Miller "Dynamic Separation of Resistance Spot Welded Joints: Part II—Analysis of Test Results and a Model" *Exp Mech* Vol 50, Issue 7, pp 901-913 (2010).

Weiss, J.A., Maker, B.N. and Govindjee, S., "Finite Element Implementation of Incompressible, Transversely Isotropic Hyperelasticity", *Comp. Meth. Appl. Mech. Eng.*, 135, 107-128 (1996).

Wen, T.K. "Method for Random Vibration of Hysteretic Systems", *J. Engrg. Mech., ASCE*, Vol. 102, No. EM2, Proc. Paper 12073, pp.249-263 (1976).

Whirley, R. G., and J. O. Hallquist, DYNA3D, A Nonlinear, Explicit, Three-Dimensional Finite Element Code for Solid and Structural Mechanics-Users Manual, Report No.UCRL-MA-107254 , Lawrence Livermore National Laboratory, (1991).

Whirley, R. G., and G.A. Henshall, "Creep Deformation Structural Analysis Using An Efficient Numerical Algorithm," *IJNME*, Vol. 35, pp. 1427-1442, (1992).

Wilkins, M.L., "Calculations of Elastic Plastic Flow," *Meth. Comp. Phys.*, **3**, (Academic Press), 211-263 (1964).

Wilkins, M.L., Calculation of Elastic-Plastic Flow, University of California, Lawrence Livermore National Laboratory, Rept. UCRL-7322, Rev. I (1969).

Wilkins, M.L., The Use of Artificial Viscosity in Multidimensional Fluid Dynamics Calculations, University of California, Lawrence Livermore National Laboratory, Rept. UCRL-78348 (1976)

Wilkins, M.L., R.E. Blum, E. Cronshagen, and P. Grantham, A Method for Computer Simulation of Problems in Solid Mechanics and Gas Dynamics in Three Dimensions and Time, University of California, Lawrence Livermore National Laboratory, Rept. UCRL-51574 (1974).

Wilkins, M.L., J.E. Reaugh, B. Moran, J.K. Scudder, D.F. Quinones, M.E. Prado, Fundamental Study of Crack Initiation and Propagation Annual Progress Report, Report UCRL-52296, Lawrence Livermore National Laboratory, Livermore, CA. (1977).

Wilkins, M.L, Streit, R.D, and Reaugh, J.E. Cumulative-Strain-Damage Model of Ductile Fracture: Simulation and Prediction of Engineering Fracture Tests, Report UCRL-53058, Lawrence Livermore National Laboritory, Livermore, CA (1980).

Williams K. V., Vaziri R., Poursartip A., "A Physically Based Continuum Damage Mechanics Model for Thin Laminated Composite Structures." *Int J Solids Struct*, Vol 40(9), 2267-2300

Wilson, E.L. *Three Dimensional Static and Dynamic Analysis of Structures,* Computers and Structures, Inc., Berkeley CA, (2000).

Winters, J.M., "Hill-based muscle models: A systems engineering perspective," In Multiple Muscle Systems: Biomechanics and Movement Organization, JM Winters and SL-Y Woo eds, Springer-Verlag (1990).

Winters J.M. and Stark L., "Estimated mechanical properties of synergistic muscles involved in movements of a variety of human joints," *J Biomechanics* 21:1027-1042, (1988).

Woodruff, J.P., KOVEC User's Manual, University of California, Lawrence Livermore National Laboratory, Report UCRL-51079, (1973).

Worswick, M.J., and Xavier Lalbin, Private communication, Livermore, California, (1999).

Wu Y., John E. Crawford, Shengrui Lan, and Joseph M., "Validation studies for concrete constitutive models with blast test data", 13th International LS-DYNA User's Conference, Dearborn MI, (2014).

Xia, Q.S., M.C. Boyce, and D.M. Parks, "A constitutive model for the anisotropic elasticplastic deformation of paper and paperboard," *Int. J. Solids and Struct.*, 39, 4053-4071 (2002).

Yamasaki, H., M. Ogura, R. Nishimura, and K. Nakamura, "Development of Material Model for Crack Propagation of Casting Aluminum", Presented at the 2006 JSAE Annual Congress, Paper Number 20065077, (2006).

Yen, C.F., "Ballistic Impact Modeling of Composite Materials", Proceedings of the 7th International LS-DYNA Users Conference, Dearborn, MI, May 19-21, 2002, 6.15-6.25.

Yoshida, F. and Uemori, T., "A Model of Large-Strain Cyclic Plasticity Describing the Bauschinger Effect and Work Hardening Stagnation", International Journal of Plasticity 18, 661-686 (2002).

Yoshida, F. and Uemori, T., "A Model of Large-Strain Cyclic Plasticity and its Application to Springback Simulation", *International Journal of Mechanical Sciences*, Vol. 45, 1687-1702, (2003).

Zajac F.E., "Muscle and tendon: Properties, models, scaling, and application to biomechanics and motor control", *CRC Critical Reviews in Biomedical Engineering* 17(4):359-411, (1989).

Zayas, V.A., Low, S.S. and Mahin, S.A., "A Simple Pendulum Technique for Achieving Seismic Isolation", *J. Earthquake Spectra*, Vol. 6, No. 2, pp. 317-334 (1990).

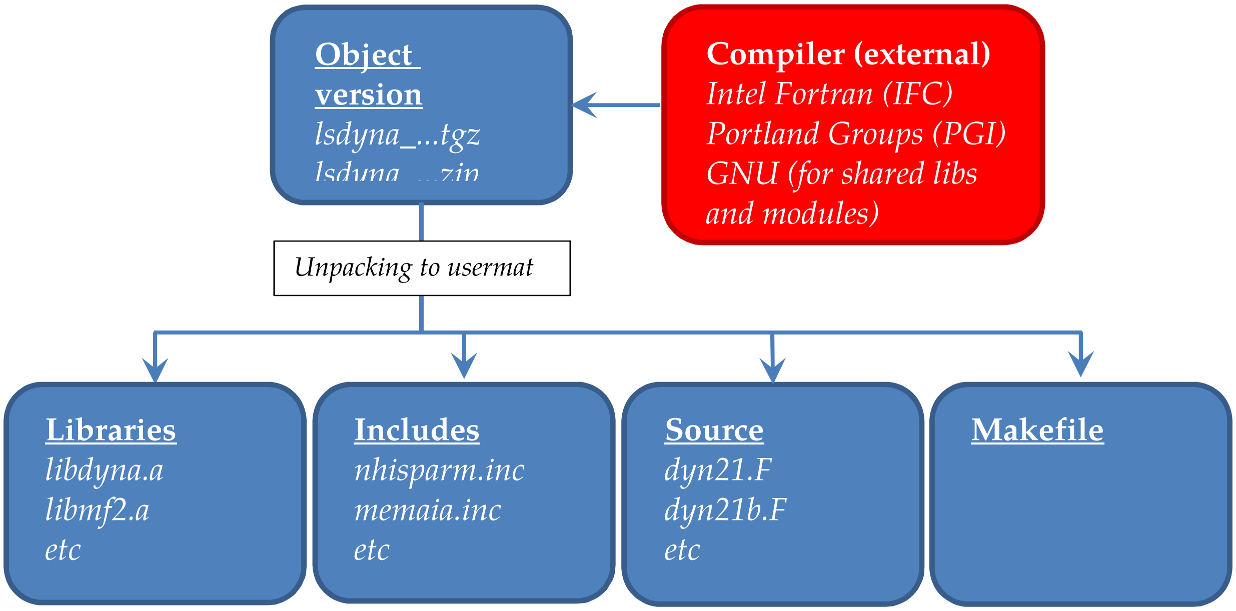
Zhang, S., Approximate Stress Intensity Factors and Notch Stresses for Common Spot-Welded Specimens, Welding Research Supplement, pp. 173s-179s, (1999).

Zhang, S., McCormick, P.G., Estrin, Y., "The morphology of Portevin-Le Chatelier bands: Finite element simulation for Al-Mg-Si", *Acta Materialia* 49, 1087-1094, (2001).

APPENDIX A: User Defined Materials

Getting Started with User Defined Features

As a way of entering the topic, we begin by giving a general introduction to User Defined Features (*UDF*) in general. This section is supposed to be valid for any UDF described in the remaining appendices and serves as a practical guide to get to the point where technical coding may commence. For a comprehensive overview of UDFs and its applications, please refer to Erhart [2010] which can be seen as a complement to the present text.



Download

The first thing to do is to download an *object version* of LS-DYNA for the computer architecture/platform of interest. This is a compressed package (.tgz,.tar.gz,.zip) provided by your local LS-DYNA distributor whose unpacked content is not only a single binary executable but a *usermat* directory possibly including

- precompiled static object libraries (.a, .lib)
- fortran source code files (.f, .F)
- include files (.inc)
- makefile

It is important to know what version to download, i.e., a version that is compatible with your computer environment in terms of architecture, operating system and possible MPI implementation in case of MPP. This may not be obvious to all users and for questions regarding this we refer to your LS-DYNA distributor. An object version does not require a special license but goes under the general LS-DYNA license agreement. The picture above gives a conceptual overview of the usermat package, and at this point we need to mention that presence of libraries and include files are made obsolete with the advent of modules as to be explained below, but we start with the non-modular approach.

Static or Dynamic linking

To get something that is actually runnable it is necessary to *compile* the source code files and link the resulting object files either to a *shared object* (.so, .dll) or with the precompiled libraries to a binary executable. The former option requires a shared object version of the usermat package while the latter assumes a statically linked version, and this is thus a choice that needs to be made before retrieving the package. Working with shared objects is more flexible in the sense that a binary executable can be installed once and for all and the (small) shared object is dynamically linked at runtime as a plugin. Upon execution, LS-DYNA will look for the shared object file in directories specified by a library path that is usually set/edited in the run command script, and when found the shared object content is linked to the execution. On Linux OS's this is for instance governed by the environment variable LD_LIBRARY_PATH. The shared object is easily substituted or the run command script can be edited to redirect the location of the shared object, and this facilitates portability when working on large projects. A statically linked version requires that the entire binary executable is generated at compile time and this option may be preferred if working on small projects or during the development phase.

Compiler and Compiling

The compilation is usually performed in the usermat (working) directory using a terminal window. The object version does not contain a compiler but it is assumed that the appropriate compiler is installed on your system and accessible from your working directory. To render compatibility between the precompiled libraries or binary executable and your compiled object files, the *appropriate* compiler is the (by LSTC) designated compiler with which the precompiled libraries or binary are readily compiled. As examples, on Linux this is typically either the Intel Fortran Compiler (IFC) on Red Hat or CentOS or Portland Groups Compiler (PGF) on SUSE, and on Windows it is the Intel Fortran Compiler (IFC) in combination with Microsoft Visual Studio (MSVS). For MPP you also need a wrapper for whatever MPI implementation you have installed, but again, consult your LS-DYNA distributor for detailed information.

To compile, execute 'make' in the terminal window and in most cases this will generate a shared object or binary executable depending on your type of LS-DYNA object version, and if not it comes down to interpreting error messages. A possible reason for failure is that the makefile that comes with the usermat package does not contain appropriate compiler directives and requires some editing. This could range from trivial tasks like altering the path used to actually find the compiler on your system to more complex endeavors such as adding or changing compiler flags. An even worse scenario is that your operating system is not up-to-date and may require that the library content on your computer is somehow updated. Whatever the reason might be, any case is unique and the situation is preferrably resolved by your computer administrator in collaboration with your LS-DYNA distributor.

Execution

When compiling a virgin instance of an object version, the generated executable *{xyz}dyna* in combination with a possible shared object *lib{xyz}dyna_{t}_{Y}.{Z}.so* makes up an exact replica of a non-object executable counterpart. It is only when the source code is modified that a customized version is generated. In the adopted binary and shared object naming convention above, *xyz* stands for *smp* or *mpp*, *t* is either *s* for single precision or *d* for double precision, Y is the base revision number for the LS-DYNA and Z is the corresponding release revision number. Z is always larger than Y. The binary and shared object may be left in the working directory, moved to some other location on your system or properly installed for execution on clusters by a queuing system. Obviously the way execution is performed is affected accordingly, and we don't claim to cover all these situations but leave this task to you and your computer administrator. While the executable may be renamed, a shared object should in general not be since this dynamic dependence is built into the executable. During the development phase it may be convenient to leave the binary in its place to facilitate debugging, but possibly move the shared object to the execution directory to not having to edit the library path for the executable to find it. Two Linux examples on how to run an input file are given in the following, one assuming an SMP static object version has been downloaded and the other an MPP shared object version. Let */path_to_my_source/usermat* be the complete path to the working directory and *in.k* be the name of the LS-DYNA keyword input file. If located in the *execution* directory, i.e., the directory containing *in.k*, the problem is run as

/path_to_my_source/usermat/lsdyna i = in.k

in the SMP case. For the MPP shared object case, you may copy the shared object file to the execution directory which is a default directory for executables to look for dynamic object files. Assuming the MPI software is platform-MPI, the input file could be run on 2 cores as

/path_to_platform-mpi/bin/mpirun –np 2 /path_to_my_source/usermat/mppdyna i = in.k

These two examples are only intended to indicate how to treat the compiled files and changes in details thereof are to be expected.

Coding

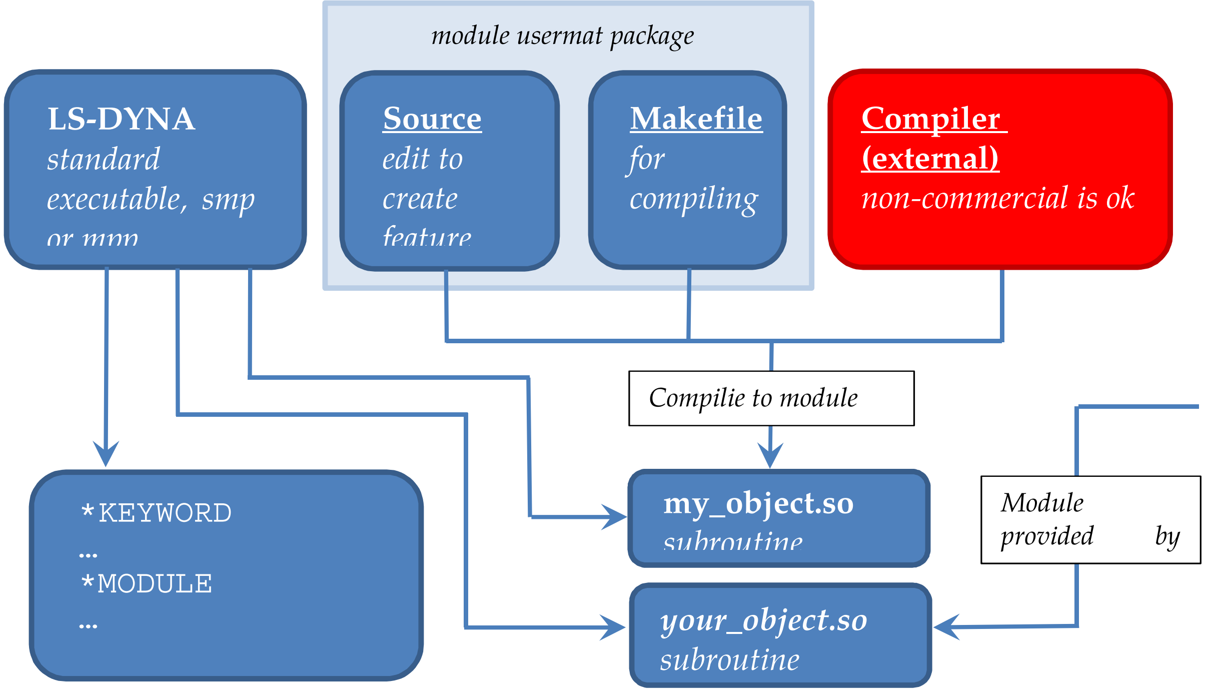
All subroutines for the UDFs are collected in the files *dyn21.f* and *dyn21b.f* and are ready for editing using your favorite text editor. Mechanical user materials, user defined loading, user defined wear and friction are contained in dyn21.f, while user defined elements and thermal user materials are in dyn21b.f, just to give a few examples. The prevailing programming language is Fortran 77, but many compilers support Fortran 90 and it may also be possible to write C code but this requires some manipulation of the makefile and function interfaces. Each individual feature is connected to one or a few keywords to properly take advantage of innovative coding. For user materials, this keyword is *MAT_USER_DEFINED_MATERIAL_MODELS and is described in detail below, and for user defined loads the corresponding keyword is *USER_LOADING. We refer to the individual keyword sections for details on their respective usage.

Module Concept

As of version R9 of LS-DYNA, there is yet another way of approaching user defined features, namely through the concept of *modules*. The purpose is twofold;

1. To facilitate working with UDFs in that the content of the usermat package is significantly reduced and instead replaced by *MODULE keywords
2. To enhance flexibility when incorporating features delivered as shared objects by third parties

By way of 1, the usermat package only consists of a makefile and a few source code files, and in principle all the user needs to do is to (*i*) implement the feature of interest in the source code files, (*ii*) compile to a shared object using the makefile, and (*iii*) use the keyword input file to access the generated object. A shared object to be used in this way is henceforth called a *module*. One of two nice side effects coming out of this approach is that the restriction on the choice of compiler is alleviated, the source code files can be compiled with any valid fortran compiler as long as the generated module is linkable as a plug-in to LS-DYNA. The other is that a *standard* LS-DYNA executable can be used to access the modules, i.e., it is not required to acquire a special “module” version to use with this approach. To explain how LS-DYNA in practice access modulus and specific routines therein, and at the same time address 2 above, the *MODULE keyword requires some attention by means of an example.



In a standard version of LS-DYNA the keywords

```
        *MODULE_PATH
        *MODULE_LOAD
        *MODULE_USE
```

are available, see the Section on `*MODULE` for further explanations than what is provided here. With reference to the picture above, we assume that *my_object.so* and *your_object.so* are two independently generated modules, and both are located in directory */path_to_modules*. We also assume that these two objects contain the same routines names, one of them being the source code for user material 41 (*subroutine umat41*). Without the present approach, it would be at least intricate to execute both of these two source codes in the same LS-DYNA executable and same keyword input. A simple way of dealing with this here is to use the following set of keywords

```
*MODULE_PATH
/path_to_modules
*MODULE_LOAD
$ MDLID                 TITLE
myid                    my library
$ FILENAME
my_object.so
*MODULE_LOAD
$ MDLID                 TITLE
yourid                  your library
$ FILENAME
your_object.so
*MODULE_USE
$ MDLID
```

```
myid
$ TYPE              PARAM1              PARAM2
UMAT                1001                41
*MODULE_USE
$ MDLID
yourid
$ TYPE              PARAM1              PARAM2
UMAT                1002                41
*MAT_USER_DEFINED_MATERIAL_MODELS
$       MID        RO        MT
         1   7.85e-9      1001
…
*MAT_USER_DEFINED_MATERIAL_MODELS
$       MID        RO        MT
         2   7.85e-9      1002
…
*PART
first part
$       PID     SECID       MID
         1         1         1
*PART
second part
$       PID     SECID       MID
         2         2         2
```

The `*MODULE_PATH` lists the path(s) to the modules to be loaded, `*MODULE_LOAD` actually loads the module into LS-DYNA, and `*MODULE_USE` tells LS-DYNA how to access routines in the module. In this particular example, the rules (`TYPE = UMAT`) are that a user material `*MAT_USER_...` with `MT` set to `1001` (because `PARAM1 = 1001`) will execute `subroutine umat41` (because `PARAM2 = 41`) in the module with id `myid` (because `MDLID = myid`) and in the same manner user material with `MT` set to `1002` will also execute `subroutine umat41` but now in the module with id `yourid`. Hence we have made part 1 and part 2, in the same keyword input file, execute the *same* subroutine (by name) but in *different* modules. Obviously this generalizes to any number of modules by analogy, see `*MODULE_USE` for many more rules and yet another example.

General overview

We now turn to the specific documentation of user defined materials. Up to ten user subroutines can currently be implemented simultaneously to update the stresses in solids, shells, beams, discrete beams and truss beams. This text serves as an introductory guide to implement such a model. Note that names of variables and subroutines below may differ from the actual ones depending on platform and current version of LS-DYNA.

When the keyword `*MAT_USER_DEFINED_MATERIAL_MODELS` is defined for a part in the keyword deck, LS-DYNA calls the subroutine `usrmat` with appropriate input data for the constitutive update. This routine in turn calls `urmathn` for 2D and 3D solid elements, `urmats` for 2D plane stress and 3D shell elements, `urmatb` for beam elements, `urmatd` for discrete beam elements and `urmatt` for truss beam elements. In these routines, which may be modified by the user if necessary, the following data structures are initialized for the purpose of being supplied to a specific *scalar* material subroutine.

`sig(6)`	stresses in previous time step
`eps(6)`	strain increments
`epsp`	effective plastic strain in previous time step
`hsv(*)`	history variables in previous time step excluding plastic strain
`dt1`	current time step size
`temper`	current temperature
`failel`	flag indicating failure of element

If the *vectorization* flag is active (`IVECT = 1`) on the material card, variables are in general stored in vector blocks of length `nlq`, with vector indexes ranging from `lft` to `llt` , which allows for a more efficient execution of the material routine. As an example, the data structures mentioned above are for the vectorized case exchanged for

`sigX(nlq)`	stresses in previous time step
`dX(nlq)`	strain increments
`epsps(nlq)`	effective plastic strains in previous time step
`hsvs(nlq,*)`	history variables in previous time step
`dt1siz(nlq)`	current time step sizes
`temps(nlq)`	current temperatures
`failels(nlq)`	flags indicating failure of elements

where *X* ranges from `1` to `6` for the different components. Each entry in a vector block is associated with an element in the finite element mesh for a fix integration point.

The number of entries in the history variables array (indicated by `*` in the above) matches the number of history variables requested on the material card (`NHV`). Hence the number `NHV` should equal to the number of history variables excluding the effective plastic strain since this variable is given a special treatment. All history variables, including the effective plastic strain, are initially zero. Furthermore, all user-defined material models require a bulk modulus and shear modulus for transmitting boundaries, contact interfaces, rigid body constraints, and time step calculations. This generally means that the length of material constants array `LMC` must be increased by 2 for the storage of these parameters. In addition to the variables mentioned above, the following data can be supplied to the user material routines, regardless of whether vectorization is used or not.

`cm(*)`	material constants array
`capa`	transverse shear correction factor for shell elements
`tt`	current time

crv(lq1,2,*) – array representation of curves defined in the keyword deck

A specific material routine, umat*XX* in the scalar case or umat*XX*v in the vector case, is now called with any necessary parameters of the ones above, and possibly others as well. The letters *XX* stands for a number between 41 and 50 and matches the number MT on the material card. This subroutine is written by the user, and should update the stresses and history variables to the current time. For shells and beams it is also necessary to determine the strain increments in the directions of constrained zero stress. To be able to write different stress updates for different elements, the following character string is passed to the user-defined subroutine

etype – character string that equals solid, shell, beam, dbeam or tbeam

A sample user subroutine of a hypo-elastic material in the scalar case is provided below. This sample and the others below are from the dyn21.F file that is distributed with version R6.1.

Sample user subroutine 41

```
      subroutine umat41 (cm,eps,sig,epsp,hsv,dt1,capa,etype,tt,
     1 temper,failel,crv,nnpcrv,cma,qmat,elsiz,idele,reject)
c
c******************************************************************
c|  Livermore Software Technology Corporation  (LSTC)             |
c|  ------------------------------------------------------------  |
c|  Copyright 1987-2008 Livermore Software Tech.  Corp              |
c|  All rights reserved                                           |
c******************************************************************
c
c     isotropic elastic material (sample user subroutine)
c
c     Variables
c
c     cm(1)=first material constant, here young's modulus
c     cm(2)=second material constant, here poisson's ratio
c        .
c        .
c        .
c     cm(n)=nth material constant
c
c     eps(1)=local x  strain increment
c     eps(2)=local y  strain increment
c     eps(3)=local z  strain increment
c     eps(4)=local xy strain increment
c     eps(5)=local yz strain increment
c     eps(6)=local zx strain increment
c
c     sig(1)=local x  stress
c     sig(2)=local y  stress
c     sig(3)=local z  stress
c     sig(4)=local xy stress
c     sig(5)=local yz stress
c     sig(6)=local zx stress
c
c     hsv(1)=1st history variable
```

```
c     hsv(2)=2nd history variable
c        .
c        .
c        .
c        .
c     hsv(n)=nth history variable
c
c     dt1=current time step size
c     capa=reduction factor for transverse shear
c     etype:
c        eq."solid" for solid elements
c        eq."sld2d" for shell forms 13, 14, and 15 (2D solids)
c        eq."shl_t" for shell forms 25, 26, and 27 (shells with thickness
c         stretch)
c        eq."shell" for all other shell elements plus thick shell forms 1
c         and 2
c        eq."tshel" for thick shell forms 3 and 5
c        eq."hbeam" for beam element forms 1 and 11
c        eq."tbeam" for beam element form 3 (truss)
c        eq."dbeam" for beam element form 6 (discrete)
c        eq."beam " for all other beam elements
c
c     tt=current problem time.
c
c     temper=current temperature
c
c     failel=flag for failure, set to .true.  to fail an integration point,
c            if .true.  on input the integration point has failed earlier
c
c     crv=array representation of curves in keyword deck
c
c     nnpcrv=# of discretization points per crv()
c
c     cma=additional memory for material data defined by LMCA at
c       6th field of 2nd crad of *DATA_USER_DEFINED
c
c     elsiz=characteristic element size
c
c     idele=element id
c
c     reject (implicit only) = set to .true.  if this implicit iterate is
c                              to be rejected for some reason
c
c     All transformations into the element local system are
c     performed prior to entering this subroutine.  Transformations
c     back to the global system are performed after exiting this
c     routine.
c
c     All history variables are initialized to zero in the input
c     phase.  Initialization of history variables to nonzero values
c     may be done during the first call to this subroutine for each
c     element.
c
c     Energy calculations for the dyna3d energy balance are done
c     outside this subroutine.
c
      include 'nlqparm'
      include 'bk06.inc'
      include 'iounits.inc'
      dimension cm(*),eps(*),sig(*),hsv(*),crv(lq1,2,*),cma(*)
      integer nnpcrv(*)
      logical failel,reject
      character*5 etype
c
      if (ncycle.eq.1) then
        if (cm(16).ne.1234567) then
```

```
          call usermsg('mat41')
        endif
      endif
c
c     compute shear modulus, g
c
      g2 =abs(cm(1))/(1.+cm(2))
      g  =.5*g2
c
      if (etype.eq.'solid'.or.etype.eq.'shl_t'.or.
     1     etype.eq.'sld2d'.or.etype.eq.'tshel') then
        if (cm(16).eq.1234567) then
          call mitfail3d(cm,eps,sig,epsp,hsv,dt1,capa,failel,tt,crv)
        else
          if (.not.failel) then
          davg=(-eps(1)-eps(2)-eps(3))/3.
          p=-davg*abs(cm(1))/(1.-2.*cm(2))
          sig(1)=sig(1)+p+g2*(eps(1)+davg)
          sig(2)=sig(2)+p+g2*(eps(2)+davg)
          sig(3)=sig(3)+p+g2*(eps(3)+davg)
          sig(4)=sig(4)+g*eps(4)
          sig(5)=sig(5)+g*eps(5)
          sig(6)=sig(6)+g*eps(6)
          if (cm(1).lt.0.) then
            if (sig(1).gt.cm(5)) failel=.true.
          endif
          endif
        end if
c
      else if (etype.eq.'shell') then
        if (cm(16).eq.1234567) then
          call mitfailure(cm,eps,sig,epsp,hsv,dt1,capa,failel,tt,crv)
        else
          if (.not.failel) then
          gc    =capa*g
          q1    =abs(cm(1))*cm(2)/((1.0+cm(2))*(1.0-2.0*cm(2)))
          q3    =1./(q1+g2)
          eps(3)=-q1*(eps(1)+eps(2))*q3
          davg  =(-eps(1)-eps(2)-eps(3))/3.
          p     =-davg*abs(cm(1))/(1.-2.*cm(2))
          sig(1)=sig(1)+p+g2*(eps(1)+davg)
          sig(2)=sig(2)+p+g2*(eps(2)+davg)
          sig(3)=0.0
          sig(4)=sig(4)+g *eps(4)
          sig(5)=sig(5)+gc*eps(5)
          sig(6)=sig(6)+gc*eps(6)
          if (cm(1).lt.0.) then
            if (sig(1).gt.cm(5)) failel=.true.
          endif
          endif
        end if
      elseif (etype.eq.'beam ' ) then
          q1    =cm(1)*cm(2)/((1.0+cm(2))*(1.0-2.0*cm(2)))
          q3    =q1+2.0*g
          gc    =capa*g
          deti  =1./(q3*q3-q1*q1)
          c22i  = q3*deti
          c23i  =-q1*deti
          fac   =(c22i+c23i)*q1
          eps(2)=-eps(1)*fac-sig(2)*c22i-sig(3)*c23i
          eps(3)=-eps(1)*fac-sig(2)*c23i-sig(3)*c22i
          davg  =(-eps(1)-eps(2)-eps(3))/3.
          p     =-davg*cm(1)/(1.-2.*cm(2))
          sig(1)=sig(1)+p+g2*(eps(1)+davg)
          sig(2)=0.0
          sig(3)=0.0
          sig(4)=sig(4)+gc*eps(4)
          sig(5)=0.0
          sig(6)=sig(6)+gc*eps(6)

c
      elseif (etype.eq.'tbeam') then
        q1    =cm(1)*cm(2)/((1.0+cm(2))*(1.0-2.0*cm(2)))
        q3    =q1+2.0*g
        deti  =1./(q3*q3-q1*q1)
        c22i  = q3*deti
        c23i  =-q1*deti
        fac   =(c22i+c23i)*q1
        eps(2)=-eps(1)*fac
        eps(3)=-eps(1)*fac
        davg  =(-eps(1)-eps(2)-eps(3))/3.
        p     =-davg*cm(1)/(1.-2.*cm(2))
        sig(1)=sig(1)+p+g2*(eps(1)+davg)
        sig(2)=0.0
        sig(3)=0.0
c
      else
c       write(iotty,10) etype
c       write(iohsp,10) etype
c       write(iomsg,10) etype
c       call adios(TC_ERROR)
        cerdat(1)=etype
        call lsmsg(3,MSG_SOL+1150,ioall,ierdat,rerdat,cerdat,0)
      endif
c
c10   format(/
c    1 ' *** Error element type ',a,' can not be',
c    2 '           run with the current material model.')
      return
      end
```

Based on the subroutine umat41 shown above, the following material input…

```
*MAT_USER_DEFINED_MATERIAL_MODELS
$#     mid        ro        mt       lmc       nhv    iortho     ibulk        ig
         1 7.8300E-6        41         4         0         0         3         4
$#   ivect     ifail    itherm    ihyper      ieos
         0         0         0         0         0
$        E        PR      BULK         G
$#      p1        p2        p3        p4        p5        p6        p7        p8
  2.000000  0.300000  1.667000  0.769200     0.000     0.000     0.000     0.000
```

… is functionally equivalent to …

```
*MAT_ELASTIC
$#     mid        ro         e        pr        da        db  not used
         1 7.8300E-6  2.000000  0.300000     0.000     0.000         0
```

ADDITIONAL FEATURES

Load curves and tables

If the material of interest should require load curves, for instance a curve defining yield stress as a function of effective plastic strain, curve and table lookup are easily obtained by predefined routines.
The routines to be called are

```
subroutine crvval(crv,nnpcrv,eid,xval,yval,slope)
```

and

```
subroutine crvval_v(crv,nnpcrv,eid,xval,yval,slope,lft,llt)
```

where the former routine is used in the scalar context and the latter for vectorized umat. The arguments are the following

crv - the load curve array (available in material routine, just pass on)

nnpcrv - curve data pointer (available in material routine, just pass on)

eid - external load curve ID, i.e., the load curve ID taken from the keyword deck

.GT.0: Use approximate representation of curve

.LT.0: Use exact representation of curve (with id −eid)

xval - abscissa value

yval - ordinate value (output from routine)

slope - slope of curve (output from routine)

lft - first index of vector

llt - final index of vector

where xval, yval and slope are scalars in the scalar routine and vectors of length nlq in the vectorized routine. Note that eid should be passed as float. Using a positive number for eid will use the approximative representation of the curve, whereas if eid is a negative number the extraction will be made on the curve as it is defined in the keyword input deck.

For tables, two subroutines are available for extracting values. A scalar version is

```
subroutine tabval(crv,nnpcrv,eid,dxval,yval,dslope,xval,slope)
```

and a vector version is

```
  subroutine
1 tabval_v(crv,nnpcrv,eid,dxval,yval,dslope,lft,llt,xval,slope)
```

where

crv - curve array (available in material routine, just pass on)

nnpcrv - curve pointer (available in material routine, just pass on)

`eid`	-	external table id (data type real), i.e., table id taken from keyword deck GT.0: Use approximative representation of curve LT.0: Use exact representation of curve (with id –eid)
`dxval`	-	abscissa value (x2-axis)
`yval`	-	ordinate value (y-axis, output from routine)
`dslope`	-	slope of curve (dy/dx2, output from routine)
`xval`	-	abscissa value (x1-axis)
`slope`	-	slope of curve (dy/dx1, output from routine)
`lft`	-	vector index
`llt`	-	vector index

In the scalar routine, dxval, yval, dslope, xval and slope are all scalars whereas in the vector routine they are vectors of length nlq. Also here, using a positive number for `eid` will use the approximative representation of the table, whereas if `eid` is a negative number the extraction will be made on the table as it is defined in the keyword input deck.

Local coordinate system

If the material model has directional properties, such as composites and anisotropic plasticity models, the local coordinate system option can be invoked. This is done by putting `IORTHO` equal to `1` on the material card. This also requires two additional cards with values for how the coordinate system is formed and updated. When this option is used, all data passed to the constitutive routine `umat`*`XX`* or `umat`*`XX`*`v` is in the local system and the transformation back to the global system is done outside this user-defined routine. There is one exception however, see the section on the deformation gradient.

Temperature

For a material with thermal properties, temperatures are made available by putting the flag `ITHERMAL` equal to `1` on the material card. The temperatures in the elements are then available in the `temper` variable for a scalar and `temps` array for the vectorized implementation. For a coupled thermal structural analysis, the thermal problem is solved first and temperatures at the current time are available in the user-defined subroutine. Calculation of dissipated heat in the presence of plastic deformation is taken care of by LS-DYNA and needs not be considered by the user. If the time derivative of the temperature is needed for the stress update, a history variable that contains the temperature in the previous time step should be requested. The time derivative can then be obtained by a backward finite difference estimate.

Failure

It is possible to include failure in the material model, resulting in the deletion of elements that fulfill a certain failure criterion. To accomplish this, the flag `IFAIL` must be set to `1` or a negative number on the material card. For a scalar implementation, the variable `failel` is set to `.true.` when a failure criterion is met. For a vectorized implementation, the corresponding entry in the `failels` array is set to `.true`.

Deformation gradient

For some materials, the stresses are not obtained from incremental strains, but are expressed in terms of the deformation gradient **F**. This is the case for hyper-elastic(-plastic) materials. To make the deformation gradient available for bricks and shells in the user-defined material subroutines, the variable `IHYPER` on the material card should be set to `1`. The deformation gradient components F_{11}, F_{21}, F_{31}, F_{12}, F_{22}, F_{32}, F_{13}, F_{23} and F_{33}can then be found in the history variables array in positions `NHV+1` to `NHV+9`, i.e., the positions coming right after the requested number of history variables.

For shell elements, the components of the deformation gradient are with respect to the co-rotational system for the element currently used. In this case the third row of the deformation gradient, i.e., the components F_{31}, F_{32} and F_{33}, will not be properly updated when entering the user-defined material routine. These components depend on the thickness strain increment which in turn must be determined so that the normal stress in the shell vanishes. For a given thickness strain increment `d3`, these three components, `f31`, `f32` and `f33`, can be determined by calling the subroutine

```
subroutine compute_f3s(f31,f32,f33,d3)
```

for a scalar implementation and

```
subroutine compute_f3(f31,f32,f33,d3,lft,llt)
```

for a vector implementation. The first four arguments are arrays of length `nlq` for the vector routine and scalars for the scalar routine.

For hyper-elastic materials there are push forward operations that can be called from within the user defined subroutines. These are

```
subroutine push_forward_2(sig1,sig2,sig3,sig4,sig5,sig6,
      f11,f21,f31,f12,f22,f32,f13,f23,f33,lft,llt)
```

which performs a push forward operation on the stress tensor, and the corresponding scalar routine

```
subroutine push_forward_2s(sig1,sig2,sig3,sig4,sig5,sig6,
      f11,f21,f31,f12,f22,f32,f13,f23,f33)
```

In the latter subroutine all arguments are scalars whereas the corresponding entries in the vectorized routine are vectors of length `nlq`. The `sig1` to `sig6` are components of the stress tensor and `f11` to `f33` are components of the deformation gradient.

If the local coordinate system option is invoked (`IORTHO = 1`), then the deformation gradient is transformed to this local system prior to entering the user-defined material routine according to

$$\bar{F}_{ij} = Q^s_{ki} F_{kj}$$

where Q^s_{ij} refers to a transformation between the current global and material frames. For `IORTHO` equal to `1` one can choose to put `IHYPER` equal to `-1` which results in that the deformation gradient is transformed according to

$$\bar{F}_{ij} = F_{ik} Q^r_{kj}$$

where Q^r_{ij} is the transformation between the reference global and material and frames. For this latter option the spatial frame remains the global one so the stresses should be expressed in this frame of reference upon exiting the user defined routines. The suitable choice of `IHYPER` depends on the formulation of the material model.

For shells, there is also the special option of setting `IHYPER = 3` which will make the deformation gradient computed from the nodal coordinates and in the global coordinate system. With this option the user must compute the stress in the local system of interest, whence a transformation matrix between the global and this local system is passed to the user material routines (`qmat`). The columns in this matrix correspond to local basis vectors expressed in global coordinates, and this is the system that stress needs to be computed in. The user must be aware that since the deformation gradient is calculated directly from the element deformation it may not be consistent with the theory of the element that is used for the material. To account for thickness changes due to membrane straining, there are routines

```
subroutine usrshl_updatfs(f,t,s,e)
real f(3,3),t(4),s(4),e
```

that recompute the deformation gradient based on the thickness strain increment `e`, and the nodal thicknesses `t`. The current nodal thicknesses are stored in the history variables array immediately following the storage of the deformation gradient. This routine must be called with these four values as `t`. This subroutine is expected to produce a deformation `f` and the new thicknesses `s`. This routine is used to find the strain increment `e` giving zero thickness stress. Once zero thickness stress is obtained, the user needs to store the new thicknesses `s` in the history variables array, which is achieved by copying the new thicknesses `s` to the location for the nodal thicknesses. There is also a vectorized version of this routine called `usrshl_updatfv`. Sample code is provided in the object library.

In the following, a Neo-Hookean material is used as an example of the usage of the deformation gradient in user-defined materials. With λ and μ being the Lame parameters in the linearized theory, the strain energy density for this material is given by

$$\psi = \frac{1}{2}\lambda(\ln(\det\mathbf{F}))^2 - \mu\ln(\det\mathbf{F}) + \frac{1}{2}\mu(\mathrm{tr}(\mathbf{F}^T\mathbf{F}) - 3)$$

meaning that the Cauchy stress can be expressed as

$$\sigma = \frac{1}{\det\mathbf{F}}(\lambda\ln(\det\mathbf{F})\mathbf{I} + \mu(\mathbf{F}\mathbf{F}^T - \mathbf{I})).$$

Sample user subroutine 45

```
      subroutine umat45 (cm,eps,sig,epsp,hsv,dt1,capa,
     .  etype,time,temp,failel,crv,nnpcrv,cma,qmat,elsiz,idele,reject)
c
c******************************************************************
c|  Livermore Software Technology Corporation  (LSTC)             |
c|  ------------------------------------------------------------  |
c|  Copyright 1987-2008 Livermore Software Tech.  Corp             |
c|  All rights reserved                                           |
c******************************************************************
c
c     Neo-Hookean material (sample user subroutine)
c
c     Variables
c
c     cm(1)=first material constant, here young's modulus
c     cm(2)=second material constant, here poisson's ratio
c        .
c        .
c        .
c     cm(n)=nth material constant
c
c     eps(1)=local x  strain increment
c     eps(2)=local y  strain increment
c     eps(3)=local z  strain increment
c     eps(4)=local xy strain increment
c     eps(5)=local yz strain increment
c     eps(6)=local zx strain increment
c
c     sig(1)=local x  stress
c     sig(2)=local y  stress
c     sig(3)=local z  stress
c     sig(4)=local xy stress
c     sig(5)=local yz stress
c     sig(6)=local zx stress
c
c     hsv(1)=1st history variable
c     hsv(2)=2nd history variable
c        .
c        .
c        .
c        .
c     hsv(n)=nth history variable
c
c     dt1=current time step size
c     capa=reduction factor for transverse shear
c     etype:
c       eq."solid" for solid elements
c       eq."sld2d" for shell forms 13, 14, and 15 (2D solids)
c       eq."shl_t" for shell forms 25, 26, and 27 (shells with thickness
c        stretch)
c       eq."shell" for all other shell elements plus thick shell forms 1
c        and 2
```

```
c         eq."tshel" for thick shell forms 3 and 5
c         eq."hbeam" for beam element forms 1 and 11
c         eq."tbeam" for beam element form 3 (truss)
c         eq."dbeam" for beam element form 6 (discrete)
c         eq."beam " for all other beam elements
c
c       time=current problem time.
c
c       temp=current temperature
c
c       failel=flag for failure, set to .true.  to fail an integration point,
c              if .true.  on input the integration point has failed earlier
c
c       crv=array representation of curves in keyword deck
c
c       nnpcrv=# of discretization points per crv()
c
c       cma=additional memory for material data defined by LMCA at
c         6th field of 2nd crad of *DATA_USER_DEFINED
c
c       elsiz=characteristic element size
c
c       idele=element id
c
c       reject (implicit only) = set to .true.  if this implicit iterate is
c                                to be rejected for some reason
c
c       All transformations into the element local system are
c       performed prior to entering this subroutine.  Transformations
c       back to the global system are performed after exiting this
c       routine.
c
c       All history variables are initialized to zero in the input
c       phase.   Initialization of history variables to nonzero values
c       may be done during the first call to this subroutine for each
c       element.
c
c       Energy calculations for the dyna3d energy balance are done
c       outside this subroutine.
c
        include 'nlqparm'
        include 'iounits.inc'
        include 'bk06.inc'
        character*5 etype
        dimension cm(*),eps(*),sig(*),hsv(*),crv(lq1,2,*),cma(*)
        logical failel
c
        if (ncycle.eq.1) then
          call usermsg('mat45')
        endif
c
c       compute lame parameters
c
        xlambda=cm(1)*cm(2)/((1.+cm(2))*(1.-2.*cm(2)))
        xmu=.5*cm(1)/(1.+cm(2))
c
        if (etype.eq.'solid'.or.etype.eq.'shl_t'.or.
       1     etype.eq.'sld2d'.or.etype.eq.'tshel') then
c
c         deformation gradient stored in hsv(1),...,hsv(9)
c
c         compute jacobian
c
          detf=hsv(1)*(hsv(5)*hsv(9)-hsv(6)*hsv(8))
       1      -hsv(2)*(hsv(4)*hsv(9)-hsv(6)*hsv(7))
       2      +hsv(3)*(hsv(4)*hsv(8)-hsv(5)*hsv(7))
```

```
c
c       compute left cauchy-green tensor
c
        b1=hsv(1)*hsv(1)+hsv(4)*hsv(4)+hsv(7)*hsv(7)
        b2=hsv(2)*hsv(2)+hsv(5)*hsv(5)+hsv(8)*hsv(8)
        b3=hsv(3)*hsv(3)+hsv(6)*hsv(6)+hsv(9)*hsv(9)
        b4=hsv(1)*hsv(2)+hsv(4)*hsv(5)+hsv(7)*hsv(8)
        b5=hsv(2)*hsv(3)+hsv(5)*hsv(6)+hsv(8)*hsv(9)
        b6=hsv(1)*hsv(3)+hsv(4)*hsv(6)+hsv(7)*hsv(9)
c
c       compute cauchy stress
c
        detfinv=1./detf
        dmu=xmu-xlambda*log(detf)
        sig(1)=detfinv*(xmu*b1-dmu)
        sig(2)=detfinv*(xmu*b2-dmu)
        sig(3)=detfinv*(xmu*b3-dmu)
        sig(4)=detfinv*xmu*b4
        sig(5)=detfinv*xmu*b5
        sig(6)=detfinv*xmu*b6
c
      else if (etype.eq.'shell') then
c
c       deformation gradient stored in hsv(1),...,hsv(9)
c
c       compute part of left cauchy-green tensor
c       independent of thickness strain increment
c
        b1=hsv(1)*hsv(1)+hsv(4)*hsv(4)+hsv(7)*hsv(7)
        b2=hsv(2)*hsv(2)+hsv(5)*hsv(5)+hsv(8)*hsv(8)
        b4=hsv(1)*hsv(2)+hsv(4)*hsv(5)+hsv(7)*hsv(8)
c
c       secant iterations for zero normal stress
c
        do iter=1,5
c
c         first thickness strain increment initial guess
c         assuming Poisson's ratio different from zero
c
          if (iter.eq.1) then
            eps(3)=-xlambda*(eps(1)+eps(2))/(xlambda+2.*xmu)
c
c         second thickness strain increment initial guess
c
          else if (iter.eq.2) then
            sigold=sig(3)
            epsold=eps(3)
            eps(3)=0.
c
c         secant update of thickness strain increment
c
          else if (abs(sig(3)-sigold).gt.0.0) then
            deps=-(eps(3)-epsold)/(sig(3)-sigold)*sig(3)
            sigold=sig(3)
            epsold=eps(3)
            eps(3)=eps(3)+deps
          endif
c
c         compute last row of deformation gradient
c
          call compute_f3s(hsv(3),hsv(6),hsv(9),eps(3))
c
c         compute jacobian
c
          detf=hsv(1)*(hsv(5)*hsv(9)-hsv(6)*hsv(8))
     1        -hsv(2)*(hsv(4)*hsv(9)-hsv(6)*hsv(7))
     2        +hsv(3)*(hsv(4)*hsv(8)-hsv(5)*hsv(7))
c
c         compute normal component of left cauchy-green tensor
c
          b3=hsv(3)*hsv(3)+hsv(6)*hsv(6)+hsv(9)*hsv(9)
c
c         compute normal stress
c
          detfinv=1./detf
          dmu=xmu-xlambda*log(detf)
          sig(1)=detfinv*(xmu*b1-dmu)
          sig(2)=detfinv*(xmu*b2-dmu)
          sig(3)=detfinv*(xmu*b3-dmu)
          sig(4)=detfinv*xmu*b4
c
c         exit if normal stress is sufficiently small
c
          if (abs(sig(3)).le.1.e-5*
     1     (abs(sig(1))+abs(sig(2))+abs(sig(4)))) goto 10
         enddo
c
c       compute remaining components of left cauchy-green tensor
c
 10     b5=hsv(2)*hsv(3)+hsv(5)*hsv(6)+hsv(8)*hsv(9)
        b6=hsv(1)*hsv(3)+hsv(4)*hsv(6)+hsv(7)*hsv(9)
c
c       compute remaining stress components
c
        sig(5)=detfinv*xmu*b5
        sig(6)=detfinv*xmu*b6
c
c     material model only available for solids and shells
c
      else
        cerdat(1)=etype
        call lsmsg(3,MSG_SOL+1151,ioall,ierdat,rerdat,cerdat,0)
      endif
      return
      end
```

Implicit analysis

For brick, and shell, thick shell and Hughes-Liu beam elements, a user-defined material model can also be run with implicit analysis. When an implicit analysis is requested in the input keyword deck, LS-DYNA calls the subroutine urtanh for bricks, urtans for shells and urtanb for beams with appropriate input data for the calculation of the material tangent modulus. For a scalar implementation, this routine in turn calls utan*XX* with all necessary input parameters including

es(6,6) – material tangent modulus

Again, *XX* is the number that matches MT on the material card. For a vectorized implementation, the routine utan*XX*v is called, this time with the corresponding vector block

dsave(nlq,6,6) – material tangent modulus

This subroutine builds the tangent modulus to be used for assembling the tangent stiffness matrix and must be provided by the user. This matrix is equal to the zero matrix when entering the user-defined routine, it must be symmetric and if the local coordinate system option is invoked for bricks, then it should be expressed in this local system. For shell elements, it should be expressed in the co-rotational system defined for the current shell element. All transformations back to the global system are made after exiting the user-defined routine.

A feature that can be made useful for improving convergence characteristics is the parameter reject, which can be set to .true. in the user material routine. The purpose of this parameter is to indicate *something* that renders the iteration unacceptable. An example of this something may be too much increase in plastic strain in one step, another is a criterion on the total strain increment. What LS-DYNA will do in this situation is to print a warning message 'Material model rejected current iterate' and retry the step with a smaller time step. If chosen carefully (by way of experimenting), this may result in a good trade-off between the number of implicit iterations per step and the step size for overall speed.

If the material is hyper-elastic, there are push forward operations of tangent modulus tensor available in

```
subroutine push_forward_4(dsave,
      f11,f21,f31,f12,f22,f32,f13,f23,f33,lft,llt)
```

which performs a push forward operation on the tangent modulus tensor, and the corresponding scalar routine

```
subroutine push_forward_4s(es,
      f11,f21,f31,f12,f22,f32,f13,f23,f33)
```

In the latter subroutine all arguments are scalars whereas the corresponding entries in the vectorized routine are vectors of length `nlq`. The `f11` to `f33` are components of the deformation gradient.

The following sample user subroutine illustrates how to implement the tangent stiffness modulus for the Neo-Hookean material above. The material tangent modulus is for this material given by

$$\mathbf{C} = \frac{1}{\det\mathbf{F}}(\lambda\mathbf{I} \otimes \mathbf{I} + 2(\mu - \lambda\ln(\det\mathbf{F}))\mathbf{I}).$$

Sample user subroutine 42, tangent modulus

```
      subroutine utan42(cm,eps,sig,epsp,hsv,dt1,capa,
     .     etype,tt,temper,es,crv)
c*****************************************************************
c|  livermore software technology corporation  (lstc)           |
c|  ------------------------------------------------------------ |
c|  copyright 1987-1999                                          |
c|  all rights reserved                                          |
c*****************************************************************
c
c     Neo-Hookean material tangent modulus (sample user subroutine)
c
```

```
c     Variables
c
c     cm(1)=first material constant, here young's modulus
c     cm(2)=second material constant, here poisson's ratio
c         .
c         .
c         .
c      cm(n)=nth material constant
c
c     eps(1)=local x  strain increment
c     eps(2)=local y  strain increment
c     eps(3)=local z  strain increment
c     eps(4)=local xy strain increment
c     eps(5)=local yz strain increment
c     eps(6)=local zx strain increment
c
c     sig(1)=local x  stress
c     sig(2)=local y  stress
c     sig(3)=local z  stress
c     sig(4)=local xy stress
c     sig(5)=local yz stress
c     sig(6)=local zx stress
c
c      epsp=effective plastic strain
c
c     hsv(1)=1st history variable
c     hsv(2)=2nd history variable
c        .
c        .
c        .
c        .
c     hsv(n)=nth history variable
c
c     dt1=current time step size
c     capa=reduction factor for transverse shear
c     etype:
c        eq."brick" for solid elements
c        eq."shell" for all shell elements
c        eq."beam"  for all beam elements
c        eq."dbeam"  for all discrete beam elements
c
c     tt=current problem time.
c
c     temper=current temperature
c
c     es=material tangent modulus
c
c      crv=array representation of curves in keyword deck
c
c      The material tangent modulus is set to 0 prior to entering
c      this routine.  It should be expressed in the local system
c      upon exiting this routine.  All transformations back to the
c      global system is made outside this routine.
      include 'nlqparm'
      character*(*) etype
      dimension cm(*),eps(*),sig(*),hsv(*),crv(lq1,2,*)
      dimension es(6,*)
c
c     no history variables, NHV=0
c     deformation gradient stored in hsv(1),...,hsv(9)
c
c     compute jacobian
c
      detf=hsv(1)*(hsv(5)*hsv(9)-hsv(6)*hsv(8))
     1     -hsv(2)*(hsv(4)*hsv(9)-hsv(6)*hsv(7))
     2     +hsv(3)*(hsv(4)*hsv(8)-hsv(5)*hsv(7))
```

```
c
c     compute lame parameters
c
      xlambda=cm(1)*cm(2)/((1.+cm(2))*(1.-2.*cm(2)))
      xmu=.5*cm(1)/(1.+cm(2))
c
c     compute tangent stiffness
c     same for both shells and bricks
c
      detfinv=1./detf
      dmu=xmu-xlambda*log(detf)
      es(1,1)=detfinv*(xlambda+2.*dmu)
      es(2,2)=detfinv*(xlambda+2.*dmu)
      es(3,3)=detfinv*(xlambda+2.*dmu)
      es(4,4)=detfinv*dmu
      es(5,5)=detfinv*dmu
      es(6,6)=detfinv*dmu
      es(2,1)=detfinv*xlambda
      es(3,2)=detfinv*xlambda
      es(3,1)=detfinv*xlambda
      es(1,2)=es(2,1)
      es(2,3)=es(3,2)
      es(1,3)=es(3,1)
c
      return
      end
```

User-Defined Materials with Equations of State

The following example umat44v is set up to be used with an equation of state (EOS). Unlike standard models, it updates only the deviatoric stress and it assigns a value to *PC*, the pressure cut-off. The pressure cut-off limits the amount of hydrostatic pressure that can be carried in tension (i.e., when the pressure is negative). The default value is zero, and a large negative number will allow the material to carry an unlimited pressure load in tension. It is calculated within the material model because it is typically a function of the current state of the material and varies with time. In this example, however, it is a constant value for simplicity. The pressure cut-off array is passed through the named common block `eosdloc`. Depending on the computing environment, compiler directives may be required (e.g., the task common directive in the example) for correct SMP execution.

In addition, the number of history variables, NHV, must be increased by 4 in the input file to allocate the extra storage required for the EOS. The storage is the first 4 variables in `hsvs`, and it must not be altered by the user-defined material model.

```
      subroutine umat44v(cm,d1,d2,d3,d4,d5,d6,sig1,sig2,
     .  sig3,sig4,sig5,sig6,eps,hsvs,lft,llt,dt1siz,capa,
     .  etype,tt,temps,failels,nlqa,crv)
      parameter (third=1.0/3.0)
      include 'nlqparm'
c
c***  isotropic plasticity with linear hardening
c
c***  updates only the deviatoric stress so that it can be used with
c     an equation of state
c
      character*5 etype
      logical failels
c
C_TASKCOMMON (eosdloc)
      common/eosdloc/pc(nlq)
c
      dimension cm(*),d1(*),d2(*),d3(*),d4(*),d5(*),d6(*),
     & sig1(*),sig2(*),sig3(*),sig4(*),sig5(*),sig6(*),
     & eps(*),hsvs(nlqa,*),dt1siz(*),temps(*),crv(lq1,2,*),
     & failels(*)
c
c***  shear modulus, initial yield stress, hardening, and pressure cut-off
      g   =cm(1)
      sy0 =cm(2)
      h   =cm(3)
      pcut=cm(4)
c
      ofac=1.0/(3.0*g+h)
      twog=2.0*g
c
      do i=lft,llt
c
c***    trial elastic deviatoric stress
        davg=third*(d1(i)+d2(i)+d3(i))
        savg=third*(sig1(i)+sig2(i)+sig3(i))
        sig1(i)=sig1(i)-savg+twog*(d1(i)-davg)
        sig2(i)=sig2(i)-savg+twog*(d2(i)-davg)
        sig3(i)=sig3(i)-savg+twog*(d3(i)-davg)
        sig4(i)=sig4(i)+g*d4(i)
        sig5(i)=sig5(i)+g*d5(i)
        sig6(i)=sig6(i)+g*d6(i)
c
c***    radial return
        aj2=sqrt(1.5*(sig1(i)**2+sig2(i)**2+sig3(i)**2)+
     &          3.0*(sig4(i)**2+sig5(i)**2+sig6(i)**2))
        sy=sy0+h*eps(i)
        eps(i)=eps(i)+ofac*max(0.0,aj2-sy)
        synew=sy0+h*eps(i)
        scale=synew/max(synew,aj2)
c
c***    scaling for radial return.  note that the stress is now deviatoric.
        sig1(i)=scale*sig1(i)
        sig2(i)=scale*sig2(i)
        sig3(i)=scale*sig3(i)
        sig4(i)=scale*sig4(i)
        sig5(i)=scale*sig5(i)
        sig6(i)=scale*sig6(i)
c
c***    set pressure cut-off
        pc(i)=pcut
c
      enddo
c
      return
      end
```

Post-processing a user-defined material

Post-processing a user-defined material is very similar to post-processing a regular LS-DYNA material. There are however some things that are worth being stressed, all dealing with how to post-process history variables.

First, the effective plastic strain is always written to the d3plot database and thus need not be requested by the user. It is in LS-PRE/POST treated just as it is for any other LS-DYNA material.

The number of additional history variables written to the d3plot database must be requested as the parameter NEIPH (for bricks) or NEIPS (for shells) on *DATABASE_EXTENT_BINARY. For instance, if NEIPH (NEIPS) equals 2 the first two history variables in the history variables array are obtained as history var#1 and history var#2 in the d3plot database. By putting NEIPH (NEIPS) equal to NHV, all history variables are written to the d3plot database. Furthermore, if the material uses the deformation gradient (IHYPER = 1) an additional 9 variables must be requested to make this available for post-processing, i.e., put NEIPH (NEIPS) equal to NHV+9. This makes the deformation gradient available in the d3plot database as history variables NHV+1 to NHV+9, note however that for shells it is expressed in the co-rotational system. If the local coordinate system option (IORTHO = 1) is used, then the deformation gradient is expressed in this local system. To make the deformation gradient in the global system for bricks and co-rotational system for shells available and stored as history variables NHV+10 to NHV+18, NEIPH (NEIPS) is put equal to NHV+9+9(=NHV+18).

APPENDIX B: User Defined Equation of State

The user can supply his/her own subroutines defining equation of state (EOS) models in LS-DYNA. To invoke a user-defined EOS, one must

1. Write a user EOS subroutine that is called by the LS-DYNA user EOS interface.

2. Create a custom executable which includes the EOS subroutine.

3. Invoke that subroutine by defining a part in the keyword input deck that uses `*EOS_USER_DEFINED` with the appropriate input parameters.

Subroutine ueoslib and sample subroutines ueos21s and ueos21v are provided in the file dyn21b.f. This text serves as an introductory guide to implementing such a model. Note that names of variables and subroutines below may differ from the actual ones depending on platform and current version of LS-DYNA.

General overview

When the keyword `*EOS_USER_DEFINED` is defined for a part in the keyword deck, LS-DYNA calls the subroutine `ueoslib` with the appropriate input data for the EOS update. This subroutine is called twice for each integration point in each element. The first call requires the EOS to calculate the bulk modulus, and the second updates the pressure and internal energy. In these routines, which may be modified by the user if necessary, the following data structures are initialized for the purpose of being supplied to a specific *scalar* material subroutine.

`iflag` – mode flag
EQ.-1: for initializing EOS constants
EQ.0: for calculating the bulk modulus
EQ.1: for the pressure and energy update
`cb` – bulk modulus
`pnew` – the new pressure
`rho0` – reference density
`hist` – array of user-defined history variables NHV in length
`specen` – internal energy per unit reference volume
`df` – volume ratio, V/V0
`v0` – the initial volume.
`dvol` – volume increment
`pc` – pressure cut-off

If the *vectorization* flag is active (`IVECT = 1`) on the EOS card, variables are, in general, stored in vector blocks of length `nlq`, with vector indices ranging from `lft` to `llt` , which allows for a more efficient execution of the EOS routine. As an example, the data structures mentioned above for the vectorized case are

`cb(nlq)`	–	bulk modulus
`pnew(nlq)`	–	the new pressure
`hist(nlq,*)`	–	array of user-defined history variables with NHV columns
`specen(nlq)`	–	internal energy per unit reference volume
`df(nlq)`	–	volume ratio, V/V0
`v0(nlq)`	–	the initial volume
`dvol(nlq)`	–	volume increment
`pc(nlq)`	–	pressure cut-off

The value of `nlq` is set as a parameter in the include file `nlqparm`, `included` at the top of the subroutine, and varies between machines and operating systems. Each entry in a vector block is associated with an element in the finite element mesh for a fix integration point. The number of entries in the history variables array (indicated by `*` in the above) matches the number of history variables requested on the material card (`NHV`). All history variables are initially zero and are initialized within the EOS on the first time step, when the logical variable `first`, passed through the argument list, is `.TRUE.` Furthermore, all user-defined EOS models require a bulk modulus, `cb`, for transmitting boundaries, contact interfaces, rigid body constraints, and time step calculations. In addition to the variables mentioned above, the following data can be supplied to the user material routines, regardless of whether vectorization is used or not.

`eosp(*)`	–	array of material constants from the input file
`tt`	–	current time
`crv(lq1,2,*)`	–	array representation of curves defined in the keyword deck.

A user defined EOS subroutine, ueos*XX*s in the scalar case or ueos*XX*v in the vector case, will be called for parts that point to *EOS_USER_DEFINED in the input deck. The letters *XX* stand for a number between 21 and 30 that matches the input variable EOST in the *EOS_USER_DEFINED keyword. During the initialization phase, the EOS is called with iflag = -1 to permit the initialization of constants in the user EOS. Although fewer than 48 constants may be read into the array `eosp` during the input, the user may use all 48 within the EOS subroutines. The user defined subroutine should calculate the bulk modulus when iflag = 0, and update the pressure, internal energy and history variables when iflag = 1.

The use of curves (*DEFINE_CURVE) is discussed in Appendix A.

A sample scalar user subroutine for a Gruneisen EOS is provided below and it is immediately followed by its vector counterpart.

Sample user subroutine 21

```
      subroutine ueos21s(iflag,cb,pnew,hist,rho0,eosp,specen,
     &                   df,dvol,v0,pc,dt,tt,crv,nnpcrv,first)
      include 'nlqparm'
c
c***  example scalar user implementation of the Gruneisen EOS
c
c***  variables
c          iflag ----- =0 calculate bulk modulus
c                      =1 update pressure and energy
c          cb -------- bulk modulus
c          pnew ------ new pressure
c          hist ------ history variables
c          rho0 ------ reference density
c          eosp ------ EOS constants
c          specen ---- energy/reference volume
c          df -------- volume ratio, v/v0 = rho0/rho
c          dvol ------ change in volume over time step
c          v0 -------- reference volume
c          pc -------- pressure cut-off
c          dt -------- time step size
c          tt -------- current time
c          crv ------- curve array
c          nnpcrv ---- number of points in each curve
c          first ----- logical .true.  for tt,crv,first time step
c                      (for initialization of the history variables)
c
      logical first
c
      dimension hist(*),eosp(*),crv(lq1,2,*)
      integer nnpcrv(*)
c
      c  =eosp(1)
      s1 =eosp(2)
      s2 =eosp(3)
      s3 =eosp(4)
      g0 =eosp(5)
      sa =eosp(6)
      s11=s1-1.
      s22=2.*s2
      s33=3.*s3
      s32=2.*s3
      sad2=.5*sa
      g0d2=1.-.5*g0
      roc2=rho0*c**2
c
c***  calculate the bulk modulus for the EOS contribution to the sound speed
      if (iflag.eq.0) then
        xmu=1.0/df-1.
        dfmu=df*xmu
        facp=.5*(1.+sign(1.,xmu))
        facn=1.-facp
        xnum=1.+xmu*(+g0d2-sad2*xmu)
        xdem=1.-xmu*(s11+dfmu*(s2+s3*dfmu))
        tmp=facp/(xdem*xdem)
        a=roc2*xmu*(facn+tmp*xnum)
        b=g0+sa*xmu
        pnum=roc2*(facn+facp*(xnum+xmu*(g0d2-sa*xmu)))
        pden=2.*xdem*(-s11 +dfmu*(-s22+dfmu*(s2-s33+s32*dfmu)))
        cb=pnum*(facn+tmp)-tmp*a*pden+sa*specen+
     &        b*df**2*max(pc,(a+b*specen))
c
c***  update the pressure and internal energy
      else
        xmu=1.0/df-1.
        dfmu=df*xmu
        facp=.5*(1.+sign(1.,xmu))
        facn=1.-facp
        xnum=1.+xmu*(+g0d2-sad2*xmu)
        xdem=1.-xmu*(s11+dfmu*(s2+s3*dfmu))
        tmp=facp/(xdem*xdem)
        a=roc2*xmu*(facn+tmp*xnum)
        b=g0+sa*xmu
        dvov0=0.5*dvol/v0
        denom=1.+ b*dvov0
        pnew=(a+specen*b)/max(1.e-6,denom)
        pnew=max(pnew,pc)
        specen=specen-pnew*dvov0
      endif
c
      return
      end
      subroutine ueos21v(lft,llt,iflag,cb,pnew,hist,rho0,eosp,specen,
     &                   df,dvol,v0,pc,dt,tt,crv,nnpcrv,first)
      include 'nlqparm'
c
c***  example vectorized user implementation of the Gruneisen EOS
c
c***  variables
c          lft,llt --- tt,crv,first and last indices into arrays
c          iflag ----- =0 calculate bulk modulus
c                      =1 update pressure and energy
c          cb -------- bulk modulus
c          pnew ------ new pressure
c          hist ------ history variables
c          rho0 ------ reference density
c          eosp ------ EOS constants
c          specen ---- energy/reference volume
c          df -------- volume ratio, v/v0 = rho0/rho
c          dvol ------ change in volume over time step
c          v0 -------- reference volume
c          pc -------- pressure cut-off
c          dt -------- time step size
c          tt -------- current time
c          crv ------- curve array
c          nnpcrv ---- number of points in each curve
c          first ----- logical .true.  for tt,crv,first time step
c                      (for initialization of the history variables)
c
      logical first
c
      dimension cb(*),pnew(*),hist(nlq,*),eosp(*),
     &          specen(*),df(*),dvol(*),pc(*),v0(*)
      dimension crv(lq1,2,*)
      integer nnpcrv(*)
c
      c  =eosp(1)
      s1 =eosp(2)
      s2 =eosp(3)
      s3 =eosp(4)
      g0 =eosp(5)
      sa =eosp(6)
      s11=s1-1.
      s22=2.*s2
      s33=3.*s3
      s32=2.*s3
      sad2=.5*sa
      g0d2=1.-.5*g0
      roc2=rho0*c**2
c
c***  calculate the bulk modulus for the EOS contribution to the sound speed
```

```
      if (iflag.eq.0) then
        do i=lft,llt
          xmu=1.0/df(i)-1.
          dfmu=df(i)*xmu
          facp=.5*(1.+sign(1.,xmu))
          facn=1.-facp
          xnum=1.+xmu*(+g0d2-sad2*xmu)
          xdem=1.-xmu*(s11+dfmu*(s2+s3*dfmu))
          tmp=facp/(xdem*xdem)
          a=roc2*xmu*(facn+tmp*xnum)
          b=g0+sa*xmu
          pnum=roc2*(facn+facp*(xnum+xmu*(g0d2-sa*xmu)))
          pden=2.*xdem*(-s11 +dfmu*(-s22+dfmu*(s2-s33+s32*dfmu)))
          cb(i)=pnum*(facn+tmp)-tmp*a*pden+sa*specen(i)+
     &          b*df(i)**2*max(pc(i),(a+b*specen(i)))
        enddo
c
c***  update the pressure and internal energy
      else
        do i=lft,llt
          xmu=1.0/df(i)-1.
          dfmu=df(i)*xmu
          facp=.5*(1.+sign(1.,xmu))
          facn=1.-facp
          xnum=1.+xmu*(+g0d2-sad2*xmu)
          xdem=1.-xmu*(s11+dfmu*(s2+s3*dfmu))
          tmp=facp/(xdem*xdem)
          a=roc2*xmu*(facn+tmp*xnum)
          b=g0+sa*xmu
          dvov0=0.5*dvol(i)/v0(i)
          denom=1.+b*dvov0
          pnew(i)=(a+specen(i)*b)/max(1.e-6,denom)
          pnew(i)=max(pnew(i),pc(i))
          specen(i)=specen(i)-pnew(i)*dvov0
        enddo
      endif
c
      return
      end
```

The Gruneisen EOS implemented in the example subroutines has the same form as *EOS_GRUNEISEN, EOS Form 4. Its update of the pressure and the internal energy are typical for an EOS that is linear in the internal energy,

$$P = A(\rho) + B(\rho)E$$

where A and B correspond to the variables `a` and `b` in the example subroutines, and E is `specen`. Integrating the energy equation with the trapezoidal rule gives

$$E^{n+1} = E^n + \frac{1}{2}(\sigma'^{n} + \sigma'^{n+1})\Delta\varepsilon - \frac{1}{2}(P^n + q^n + P^{n+1} + q^{n+1})\frac{\Delta V}{V_0}$$

where the superscripts refer to the time step, ΔV is the change in the volume associated with the Gauss point and V_0 is the reference volume. Collecting all the energy contributions on the right hand side except for the contribution from the new pressure gives a simple linear relationship between the new internal energy and pressure,

$$E^{n+1} = \tilde{E} - \frac{P^{n+1}\Delta V}{2V_0}.$$

The value of specen passed to ueosXX for the pressure and energy update corresponds to$\tilde{E}$. Substituting this relation into the EOS and solving for the new pressure gives

$$P^{n+1} = \frac{A\rho^{n+1} + B\rho^{n+1}\tilde{E}}{1 + \frac{B\Delta V}{2V_0}}.$$

The final update of the new energy is calculated using the new pressure. For a more general EOS, the nonlinear equation in the new pressure,

$$P^{n+1} = P\left(\rho^{n+1}, \tilde{E} - \frac{P^{n+1}\Delta V}{2V_0}\right)$$

is solved iteratively using Newton iteration or successive substitution.

The pressure cut-off, pc, is used to limited the amount of pressure that can be generated by tensile loading, pnew=max(pnew,pc). Its value is usually specified in the *MAT input, e.g., *MAT_JOHNSON_COOK. It is not enforced outside of the EOS subroutines, and it is up to the user to determine whether or not to enforce the pressure cut-off in ueosXX. If the user does enforce it, the pressure cut-off should be applied before the final update to the internal energy otherwise the energy will be incorrect.

Many of the calculations performed to calculate the bulk modulus are the same as those for updating the pressure and energy. Since the bulk modulus calculation always precedes the pressure update, the values may be saved in a common block during the bulk modulus calculation to reduce the cost of the pressure update. The arrays used to store the values in the vectorized subroutines should be dimensioned by nlq.

One of the most common errors in implementing an EOS from a paper or book is the use of the wrong internal energy. There are three internal energies in common use: the energy per unit mass,e_M, the energy per unit current volume,e_V, and the energy per unit reference volume, E. LS-DYNA always uses the energy per unit reference volume. Some useful relations for converting between EOS in the literature and the variables in LS-DYNA are

$$e_V = E\frac{V_0}{V} = \frac{\text{specen}}{\text{df}}$$

$$e_M = E\frac{V_0}{M} = \frac{\text{specen}}{\text{rho0}}$$

$$\rho = \rho_0\frac{V_0}{V} = \frac{\text{rho0}}{\text{df}}$$

APPENDIX C: User Defined Element Interface for Solids and Shells

In this appendix the user-defined element interface for solids and shells is described. The interface can accommodate either an integrated or a resultant element. For the integrated element, the user needs to supply two matrices defining the kinematical properties of the element, and choose between using standard LS-DYNA hourglass stabilization, a user-defined stabilization, or no stabilization when zero energy modes are not present. The number and location of the integration points is arbitrary, i.e., user-defined. For the resultant/discrete element formulations, the force and stiffness assembly must also be implemented. History variables can be associated with the user defined elements. If desired, the element may utilize more than the conventional 3 (for bricks) and 6 (for shells) degrees-of-freedom per node.

The user element is implemented according to how standard elements are implemented in LS-DYNA with the exception that two user routines are called for setting up the matrices of interest. In the end, the gradient-displacement matrix B_{ijkK} is constructed with the property that

$$B_{ijkK}u_{kK} = \frac{\partial v_i}{\partial x_j}$$

where u_{kK} is the vector of velocity nodal degrees of freedom and the right hand side is the velocity gradient. Moreover, the determinant J of the jacobian matrix determining the mapping from the isoparametric to physical domain is needed for numerical integration. From these expressions, the strains are determined as the symmetric part of the velocity gradient and the spin as the corresponding antisymmetric part. The stresses are evaluated using the constitutive models in LS-DYNA and the internal forces are obtained from

$$f_{kK} = \int \sigma_{ij} B_{ijkK} dV$$

where σ_{ij} are the stresses. Furthermore, the geometric and material tangent stiffnesses are obtained through

$$K_{iIjJ}^{\mathrm{mat}} = \int C_{klmn} B_{kliI} B_{mnjJ} dV$$

and

$$K_{iIjJ}^{\mathrm{geo}} = \int \sigma_{mn} B_{kmiI} B_{knjJ} dV$$

where C_{klmn} is the tangent modulus for the material. The integrals are evaluated using user-defined quadrature using the determinant J.

For user-defined hourglass control, the user must provide the corresponding internal force and stiffness contribution in a separate user routine. There is also the option to provide the force and stiffness matrix directly for the entire element.

To invoke a user-defined element one must do the following:

1. Write user element subroutine that defines the kinematics or kinetics of the element.

2. Create a custom executable which includes these subroutines.

3. Invoke the element by specifying this on the corresponding *SECTION card.

The dummy subroutines for the user defined elements are provided to the user in a FORTRAN source file for you to modify along with the necessary object files to compile a new executable. Contact LSTC or your local distributor for information about how to obtain these files as well as what compiler/version to use for your specific platform. Up to five user elements can simultaneously be used for bricks and shells (i.e. a total of ten). This text serves as an introductory guide on how to implement such an element.

General overview

To activate a user-defined element, it is necessary to set ELFORM to a number between 101 and 105 on the *SECTION definition. By doing so, the kinematics of the elements in the corresponding part will be determined from calling the subroutine

```
subroutine uXXX_bYYY(bmtrx,gmtrx,gjac,...
⋮
dimension bmtrx(nlq,3,3,*),gmtrx(nlq,3,3),gjac(*)
```

where XXX is substituted for shl for a shell-section and sld for a solid-section and YYY is the number specified in position ELFORM. Depending on the choice of ITAJ in the input, the user should set the matrices as follows.

If ITAJ = 0, then set the isoparametric gradient-displacement matrix, represented by the array bmtrx , and jacobian matrix, represented by the array gmtrx. Here, the first index corresponds to the LS-DYNA block loop index where nlq is the block size. For a more convenient notation in the following, we assign a correspondence between the arrays gmtrx and bmtrx in the subroutines to matrices/tensors as follows

gmtrx(*,i,j) - g_{ij}
bmtrx(*,i,j,k) - b_{ijk}

These matrices should be determined so that at the current integration point:

$$g_{ij} = \frac{\partial x_i}{\partial \xi_j}$$

$$b_{ijk} u_k = \frac{\partial v_i}{\partial \xi_j} \Delta t$$

In the above, summation over repeated indices is assumed. We use the following notation:

$x_i(\xi_1, \xi_2, \xi_3, t) =$ ith component of the current position vector at the isoparametric coordinate (ξ_1, ξ_2, ξ_3) and time t.

$v_i(\xi_1, \xi_2, \xi_3, t) =$ ith component of the velocity vector at the isoparemetric coordinate (ξ_1, ξ_2, ξ_3) and time t.

$\Delta t =$ current time step

$u_k =$ kth component of generalized local displacements

$\xi_i =$ ith component of the isoparametric coordinate ranging from -1.0 to 1.0

For shells, there is an option to get all variables in either the LS-DYNA local coordinate system (`ILOC=0`) or in the global coordinate system (`ILOC=1`). The matrix for the coordinate system transformation is also passed to the user routines where the columns represent the local unit base vectors. The resulting strains must always be in the local coordinate system for the constitutive evaluations. For no extra degrees of freedom (see below), the index k in the displacement expression is determined from the formula

$$k = n(m - 1) + d$$

where $n = 3$ if only translational degrees of freedom are present (typical for solids) and $n = 6$ if rotational degrees of freedom are present (typical for shells), m is the local node number ($m = 1,2, \ldots$) and d is the degree of freedom. The translational degrees of freedom correspond to $d \leq 3$ and the rotational degrees of freedom to $4 \leq d \leq 6$.

If `ITAJ=1`, the user should set up the physical gradient-displacement matrix, represented by the array `bmtrx`, and jacobian determinant, represented by the array `gjac`. Again, we assign a correspondence between the arrays `gjac` and `bmtrx` in the subroutines to matrices/tensors as follows

`gjac(*)` - J

`bmtrx(*,i,j,k)` - b_{ijk}

These matrices should be determined so that at the current integration point:

$$J = \det \frac{\partial x_i}{\partial \xi_j}$$

$$b_{ijk}u_k = \frac{\partial v_i}{\partial x_j}\Delta t$$

To be able to set up these matrices, a set of additional auxiliary variables are passed to the user element subroutines. These include the isoparametric coordinate, the element thickness, and the shape function values, and derivatives. Again, for shells these are expressed in either the local or global coordinate system depending on the user's choice. For more information on these variables, the user is referred to the comments in the subroutines.

The integrated elements can use up to a total of 100 integration points (in the plane for shells) at arbitrary locations. These must be specified in terms of isoparametric coordinates and weights following the first of the user-defined cards in the `*SECTION_` ... input. The isoparametric coordinates should range from –1 to 1 and the weights should sum up to 4 for shells and 8 for solids.

It may be necessary to incorporate hourglass stabilization to suppress zero energy modes, this is done by putting `IHGF.GT.0` in the input. For `IHGF.EQ.1`, the LS-DYNA hourglass routines are used automatically and for `IHGF.EQ.2` or `IHGF.EQ.3` the user must provide hourglass force and stiffness in a specific user-defined routine. If `IHGF.EQ.3`, physical stabilization becomes available since the resultant material tangent moduli are passed to the hourglass routine to provide the current membrane, bending and coupled membrane-bending stiffness of the material. With C_{ij} denoting the material tangent modulus in matrix form, the resultant tangent moduli are expressed as

$$\begin{aligned}
\bar{C}_{ij}^0 &= \int C_{ij}\,dV && (\text{membrane})\\
\bar{C}_{ij}^1 &= \int z^1 C_{ij}\,dV && (\text{membrane} - \text{bending})\\
\bar{C}_{ij}^2 &= \int z^2 C_{ij}\,dV && (\text{bending})
\end{aligned}$$

where z is the thickness coordinate for shells. For solids, only the first resultant modulus is passed. In this case the array has 21 entries that correspond to the subdiagonal terms of the 6 by 6 resultant matrix. For the matrix index (i,j) in the material tangent modulus matrix, where $i \geq j$, the index I of the array passed to the routine is given by

$$I = i(i-1)/2 + j$$

i.e., the subdiagonal terms are stored row-wise in the array. For shells, all three moduli are passed in the local coordinate system where each array has 15 entries corresponding to the subdiagonal terms of the 5 by 5 resultant matrices. The through thickness direction is here eliminated from the plane stress assumption. The formula for the array indices transformation above holds. This subroutine is called

```
      subroutine uXXX_eYYY(force,stiff,ndtot,...
      ⋮
```

```
     dimension force(nlq,*),stiff(nlq,ndtot,*)
```

where again XXX and YYY should be substituted as described for the other subroutines in the above. The variables in the subroutine corresponds to the force and stiffness as

```
       force(*,i) - f_i
     stiff(*,i,j) - K_ij
```

where the indices corresponds to node and degree of freedom numbers exactly as for the displacements. For shells the force and stiffness is set up in the local element system (ILOC=0) or global system (ILOC=1). The variable ndtot is the total number of degrees of freedom for the element. Passed to this subroutine are also the property parameters and history variables associated with the element. The values of the property parameters are defined in the input of a user-defined element. No more than 40 property parameters and 100 history variables can be used for each user-defined element. The history variables must be updated in this routine by the user.

Resultant/discrete elements

By putting NIP(P) equal to 0 in the input, a resultant/discrete element is assumed. For this option (which is incompatible with IHGF.GT.0) the user must provide force and stiffness in the same user-defined routine as for the user-defined hourglass control. This means that no material routine is called to update stresses and history variables, rather stresses and history variables are to be updated from within the user element routine.

Nevertheless, the user should define *MAT_ELASTIC as the material for the corresponding part with suitable values of the Young's modulus and Poisson's ratio. These material properties are used to calculate the time step and for determining contact stiffnesses. Again, property parameters and history variables are passed to the routine, and for shells also the thicknesses of the elements. For the shell thickness update option (ISTUPD.GT.0 on *CONTROL_SHELL) it is up to the user to update the thicknesses in this routine. For this option, and this option only, the stiffness matrix assembled in the element routines can be input as nonsymmetric *if* LCPACK=3 on *CONTROL_IMPLICIT_SOLVER, i.e., if the nonsymmetric solver is used to update the Newton iterates.

In what follows, a short description of the additional features associated with the user elements is given.

Nodal fiber vectors

If a user-defined shell element formulation uses the nodal fiber vectors, this must be specified by putting IUNF=1 on the *SECTION_SHELL card. With this option the nodal fiber vectors are processed in the element routines and can be used as input for determining the b_{ijk}, g_{ij}/J, f_i and K_{ij} tensors/matrices in the user routines. If not, it is assumed that the fiber direction is normal to the plane of the shell at all times. These are expressed in either the local or global system depending on the user's choice. See comments in the subroutines for more information.

Extra degrees of freedom

Exotic element formulations may require extra degrees-of-freedom per node besides the translational (and rotational) degrees-of-freedom. Currently, up to 3 extra degrees of freedom per node can be used for user-defined elements. To use extra degrees of freedom, a scalar node must be defined for each node that makes up the connectivity of the user element. A scalar node is defined using the keyword *NODE_SCALAR_VALUE, in which the user also prescribe initial and boundary conditions associated with the extra variables. The connectivity of the user elements must then be specified with the option *ELEMENT_SOLID_DOF or *ELEMENT_SHELL_DOF, where an extra line is used to connect the scalar nodes to the element. As an example:

```
*NODE_SCALAR_VALUE
$    NID              V1              V2              V3     NDF
      11             1.0                                       1
      12             1.0                                       1
      13             1.0                                       1
      14             1.0                                       1
*ELEMENT_SHELL_DOF
$    EID     PID      N1      N2      N3      N4
       1       1       1       2       3       4
$                    NS1     NS2     NS3     NS4
                      11      12      13      14
```

defines an element with one extra degree of freedom. The initial value of the corresponding variable is 1.0 and it is unconstrained. Finally, the user sets the parameter NXDOF on the *SECTION_… card to 1, 2 or 3 depending on how many extra degrees of freedom that should be used in the user-defined element. An array xdof containing the current values of these extra variables are passed to the user routines for setting up the correct kinematical properties, see comments in the routines for more information. The formula for the displacement index changes to

$$k = (n + n_{xdof})(m - 1) + d$$

where n_{xdof} is the number of extra degrees of freedom. The extra degrees of freedom for each node corresponds to $n + 1 \leq d \leq n + n_{xdof}$. For dynamic simulations, the mass corresponding to these extra nodes are defined using *ELEMENT_INERTIA or *ELEMENT_MASS.

Related keywords:

The following is a list of keywords that apply to the user defined elements

The *SECTION_SHELL card

A third card with accompanying optional cards of the *SECTION_SHELL keyword must be added if the user defined element option is invoked

Additional Card for ELFORM = 101,102,103,104 or 105

Card 3	1	2	3	4	5	6	7	8
Variable	NIPP	NXDOF	IUNF	IHGF	ITAJ	LMC	NHSV	ILOC
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

Include NIPP cards according to the following format.

Card 4	1	2	3	4	5	6	7	8
Variable	XI	ETA	WGT					
Type	F	F	F					

Define LMC property parameters using 8 parameters per card.

Card 5	1	2	3	4	5	6	7	8
Variable	P1	P2	P3	P4	P5	P6	P7	P8
Type	F	F	F	F	F	F	F	F

VARIABLE	DESCRIPTION
ELFORM	GT.100.AND.LT.106: User-defined shell
NIPP	Number of in-plane integration points for user-defined shell (0 if resultant element)

VARIABLE	DESCRIPTION
NXDOF	Number of extra degrees of freedom per node for user-defined shell
IUNF	Flag for using nodal fiber vectors in user-defined shell EQ.0: Nodal fiber vectors are not used. EQ.1: Nodal fiber vectors are used
IHGF	Flag for using hourglass stabilization (NIPP.GT.0) EQ.0: Hourglass stabilization is not used EQ.1: LS-DYNA hourglass stabilization is used EQ.2: User-defined hourglass stabilization is used EQ.3: Same as 2, but the resultant material tangent moduli are passed
ITAJ	Flag for setting up finite element matrices (NIPP.GT.0) EQ.0: Set up matrices wrt isoparametric domain EQ.1: Set up matrices wrt physical domain
LMC	Number of property parameters
NHSV	Number of history variables
ILOC	Local coordinate system option EQ.0: All variables are passed in the local element system EQ.1: All variables are passed in the global system
XI	First isoparametric coordinate
ETA	Second isoparametric coordinate
WGT	Isoparametric weight
PI	Ith property parameter

For more information on the variables the user may consult the previous sections in this appendix.

The *SECTION_SOLID card

A second card with accompanying optional cards of the *SECTION_SOLID keyword must be added if the user defined elements option is invoked.

Additional card for ELFORM = 101,102,103,104 or 105

Card 3	1	2	3	4	5	6	7	8
Variable	NIP	NXDOF	IHGF	ITAJ	LMC	NHSV		
Type	I	I	I	I	I	I		
Default	0	0	0	0	0	0		

Include NIP cards according to the following format.

Card 4	1	2	3	4	5	6	7	8
Variable	XI	ETA	ZETA	WGT				
Type	F	F	F	F				

Define LMC property parameters using 8 parameters per card.

Card 5	1	2	3	4	5	6	7	8
Variable	P1	P2	P3	P4	P5	P6	P7	P8
Type	F	F	F	F	F	F	F	F

VARIABLE	DESCRIPTION
ELFORM	GT.100.AND.LT.106: User-defined solid
NIP	Number of integration points for user-defined solid (0 if resultant element)
NXDOF	Number of extra degrees of freedom per node for user-defined solid

VARIABLE	DESCRIPTION
IHGF	Flag for using hourglass stabilization (NIP.GT.0) EQ.0: Hourglass stabilization is not used EQ.1: LS-DYNA hourglass stabilization is used EQ.2: User-defined hourglass stabilization is used EQ.3: Same as 2, but the resultant material tangent moduli are passed
ITAJ	Flag for setting up finite element matrices (NIP.GT.0) EQ.0: Set up matrices wrt isoparametric domain EQ.1: Set up matrices wrt physical domain
LMC	Number of property parameters
NHSV	Number of history variables
XI	First isoparametric coordinate
ETA	Second isoparametric coordinate
ZETA	Third isoparametric coordinate
WGT	Isoparametric weight
PI	Ith property parameter

For more information on the variables the user may consult the previous sections in this appendix.

Sample User Shell Element 101 (Belytschko-Tsay shell)

The geometry of the Belytschko-Tsay element in local coordinates can be written

$$x_i = (x_{iI} + \frac{t}{2} \xi_3 \delta_{i3}) N_I(\xi_1, \xi_2)$$

$$v_i = (v_{iI} + \frac{t}{2} \xi_3 e_{ij3} \omega_{jI}) N_I(\xi_1, \xi_2)$$

Where,

$x_{iI} = i$th component of coordinate of node$I$

$v_{iI} = i$th component of translational velocity of node I

$\omega_{jI} = j$th component of rotational velocity of node I

t = thickness of element

e_{ijk} = permutation tensor

N_I = shape function localized at node I

δ_{i3} = Kronecker delta

Taking the derivative of these expressions with respect to the isoparametric coordinate yields

$$\frac{\partial x_i}{\partial \xi_1} = (x_{iI} + \frac{t}{2}\xi_3\delta_{i3})\frac{\partial N_I}{\partial \xi_1}$$

$$\frac{\partial x_i}{\partial \xi_2} = (x_{iI} + \frac{t}{2}\xi_3\delta_{i3})\frac{\partial N_I}{\partial \xi_2}$$

$$\frac{\partial x_i}{\partial \xi_3} = \frac{t}{2}\delta_{i3}$$

and

$$\frac{\partial v_i}{\partial \xi_1} = (v_{iI} + \frac{t}{2}\xi_3 e_{ij3}\omega_{jI})\frac{\partial N_I}{\partial \xi_1}$$

$$\frac{\partial v_i}{\partial \xi_2} = (v_{iI} + \frac{t}{2}\xi_3 e_{ij3}\omega_{jI})\frac{\partial N_I}{\partial \xi_2}$$

$$\frac{\partial v_i}{\partial \xi_3} = \frac{t}{2}e_{ij3}\omega_{jI}N_I$$

respectively. Using these expressions the element is implemented as a user-defined shell as follows.

```
      subroutine ushl_b101(bmtrx,gmtrx,gjac,
     1     xi,eta,zeta,
     2     n1,n2,n3,n4,
     3     dn1dxi,dn2dxi,dn3dxi,dn4dxi,
     4     dn1deta,dn2deta,dn3deta,dn4deta,
     5     x1,x2,x3,x4,y1,y2,y3,y4,z1,z2,z3,z4,
     6     xdof,
     7     thick,thck1,thck2,thck3,thck4,
     8     fx1,fx2,fx3,fx4,
     9     fy1,fy2,fy3,fy4,
     .     fz1,fz2,fz3,fz4,
.     gl11,gl21,gl31,gl12,gl22,gl32,gl13,gl23,gl33,
     .     lft,llt)
      include 'nlqparm'
c
c     Compute b and g matrix for user-defined shell 101
c
      dimension bmtrx(nlq,3,3,*),gmtrx(nlq,3,3),gjac(nlq)
      REAL n1,n2,n3,n4
      dimension x1(nlq),x2(nlq),x3(nlq),x4(nlq)
      dimension y1(nlq),y2(nlq),y3(nlq),y4(nlq)
      dimension z1(nlq),z2(nlq),z3(nlq),z4(nlq)
      dimension thick(nlq)
      dimension thck1(nlq),thck2(nlq),thck3(nlq),thck4(nlq)
      dimension xdof(nlq,8,3)
      dimension fx1(nlq),fx2(nlq),fx3(nlq),fx4(nlq)
      dimension fy1(nlq),fy2(nlq),fy3(nlq),fy4(nlq)
      dimension fz1(nlq),fz2(nlq),fz3(nlq),fz4(nlq)
 dimension gl11(nlq),gl21(nlq),gl31(nlq),
.      gl12(nlq),gl22(nlq),gl32(nlq),
.      gl13(nlq),gl23(nlq),gl33(nlq)
c
      do i=lft,llt
c
         gmtrx(i,1,1)=
     1        x1(i)*dn1dxi+x2(i)*dn2dxi+
     2        x3(i)*dn3dxi+x4(i)*dn4dxi
         gmtrx(i,2,1)=
     1        y1(i)*dn1dxi+y2(i)*dn2dxi+
     2        y3(i)*dn3dxi+y4(i)*dn4dxi
         gmtrx(i,3,1)=
     1        0.
         gmtrx(i,1,2)=
     1        x1(i)*dn1deta+x2(i)*dn2deta+
     2        x3(i)*dn3deta+x4(i)*dn4deta
         gmtrx(i,2,2)=
     1        y1(i)*dn1deta+y2(i)*dn2deta+
     2        y3(i)*dn3deta+y4(i)*dn4deta
         gmtrx(i,3,2)=
     1        0.
         gmtrx(i,1,3)=
     1        0.
         gmtrx(i,2,3)=
     1        0.
         gmtrx(i,3,3)=
     1        .5*thick(i)
c
         coef=.5*thick(i)*zeta
c
         bmtrx(i,1,1,1) =dn1dxi
         bmtrx(i,1,1,7) =dn2dxi
         bmtrx(i,1,1,13)=dn3dxi
         bmtrx(i,1,1,19)=dn4dxi
c
         bmtrx(i,1,1,5) =coef*dn1dxi
         bmtrx(i,1,1,11)=coef*dn2dxi
         bmtrx(i,1,1,17)=coef*dn3dxi
         bmtrx(i,1,1,23)=coef*dn4dxi
c
         bmtrx(i,1,2,1) =dn1deta
         bmtrx(i,1,2,7) =dn2deta
         bmtrx(i,1,2,13)=dn3deta
         bmtrx(i,1,2,19)=dn4deta
c
         bmtrx(i,1,2,5) =coef*dn1deta
         bmtrx(i,1,2,11)=coef*dn2deta
         bmtrx(i,1,2,17)=coef*dn3deta
         bmtrx(i,1,2,23)=coef*dn4deta
c
         bmtrx(i,2,1,2) =dn1dxi
         bmtrx(i,2,1,8) =dn2dxi
         bmtrx(i,2,1,14)=dn3dxi
         bmtrx(i,2,1,20)=dn4dxi
c
         bmtrx(i,2,1,4) =-coef*dn1dxi
         bmtrx(i,2,1,10)=-coef*dn2dxi
         bmtrx(i,2,1,16)=-coef*dn3dxi
         bmtrx(i,2,1,22)=-coef*dn4dxi
```

```
c
         bmtrx(i,1,3,5) =.5*thick(i)*n1
         bmtrx(i,1,3,11)=.5*thick(i)*n2
         bmtrx(i,1,3,17)=.5*thick(i)*n3
         bmtrx(i,1,3,23)=.5*thick(i)*n4
c
         bmtrx(i,3,1,3) =dn1dxi
         bmtrx(i,3,1,9) =dn2dxi
         bmtrx(i,3,1,15)=dn3dxi
         bmtrx(i,3,1,21)=dn4dxi
c
         bmtrx(i,2,2,2) =dn1deta
         bmtrx(i,2,2,8) =dn2deta
         bmtrx(i,2,2,14)=dn3deta
         bmtrx(i,2,2,20)=dn4deta
c
         bmtrx(i,2,2,4) =-coef*dn1deta
         bmtrx(i,2,2,10)=-coef*dn2deta
         bmtrx(i,2,2,16)=-coef*dn3deta
         bmtrx(i,2,2,22)=-coef*dn4deta
c
         bmtrx(i,2,3,4) =-.5*thick(i)*n1
         bmtrx(i,2,3,10)=-.5*thick(i)*n2
         bmtrx(i,2,3,16)=-.5*thick(i)*n3
         bmtrx(i,2,3,22)=-.5*thick(i)*n4
c
         bmtrx(i,3,2,3) =dn1deta
         bmtrx(i,3,2,9) =dn2deta
         bmtrx(i,3,2,15)=dn3deta
         bmtrx(i,3,2,21)=dn4deta
c
      enddo
c
      return
      end
```

```
To use the element for a part the section card can be written as
*SECTION_SHELL
$    SECID    ELFORM
         1       101
$       T1        T2        T3        T4

$     NIPP     NXDOF      IUNF      IHGF
         1         0         0         1
$       XI       ETA       WGT
        0.        0.        4.
```

Sample User Solid Element 101 (constant stress solid)

The geometry for the constant stress solid is given as

$$x_i = x_{iI} N_I(\xi_1, \xi_2)$$

$$v_i = v_{iI} N_I(\xi_1, \xi_2)$$

where,

x_{iI} = ith component of coordinate of node I

v_{iI} = ith component of translational velocity of node I

N_I = shape function localized at node I

The matrices necessary for implementing this element as a user-defined solid are derived from the expressions given by

$$\frac{\partial x_i}{\partial \xi_1} = x_{iI}\frac{\partial N_I}{\partial \xi_1}$$

$$\frac{\partial x_i}{\partial \xi_2} = x_{iI}\frac{\partial N_I}{\partial \xi_2}$$

$$\frac{\partial x_i}{\partial \xi_3} = x_{iI}\frac{\partial N_I}{\partial \xi_3}$$

and,

$$\frac{\partial v_i}{\partial \xi_1} = v_{iI}\frac{\partial N_I}{\partial \xi_1}$$

$$\frac{\partial v_i}{\partial \xi_2} = v_{iI}\frac{\partial N_I}{\partial \xi_2}$$

$$\frac{\partial v_i}{\partial \xi_3} = v_{iI}\frac{\partial N_I}{\partial \xi_3}$$

The user element implementation is given by

```
      subroutine usld_b101(bmtrx,gmtrx,gjac,
     1     xi,eta,zeta,
     2     n1,n2,n3,n4,n5,n6,n7,n8,
     3     dn1dxi,dn2dxi,dn3dxi,dn4dxi,
     4     dn5dxi,dn6dxi,dn7dxi,dn8dxi,
     5     dn1deta,dn2deta,dn3deta,dn4deta,
     6     dn5deta,dn6deta,dn7deta,dn8deta,
     7     dn1dzeta,dn2dzeta,dn3dzeta,dn4dzeta,
     8     dn5dzeta,dn6dzeta,dn7dzeta,dn8dzeta,
     9     x1,x2,x3,x4,x5,x6,x7,x8,
     .     y1,y2,y3,y4,y5,y6,y7,y8,
     .     z1,z2,z3,z4,z5,z6,z7,z8,
     .     xdof,
     .     lft,llt)
      include 'nlqparm'
c
c     Compute b and g matrix for user-defined solid 101
c
      dimension bmtrx(nlq,3,3,*),gmtrx(nlq,3,3),gjac(nlq)
      REAL n1,n2,n3,n4,n5,n6,n7,n8
      dimension x1(nlq),x2(nlq),x3(nlq),x4(nlq)
      dimension x5(nlq),x6(nlq),x7(nlq),x8(nlq)
      dimension y1(nlq),y2(nlq),y3(nlq),y4(nlq)
      dimension y5(nlq),y6(nlq),y7(nlq),y8(nlq)
      dimension z1(nlq),z2(nlq),z3(nlq),z4(nlq)
      dimension z5(nlq),z6(nlq),z7(nlq),z8(nlq)
      dimension xdof(nlq,8,3)
c
      do i=lft,llt
c
         gmtrx(i,1,1)=x1(i)*dn1dxi+x2(i)*dn2dxi+
     1        x3(i)*dn3dxi+x4(i)*dn4dxi+
     2        x5(i)*dn5dxi+x6(i)*dn6dxi+
     3        x7(i)*dn7dxi+x8(i)*dn8dxi
         gmtrx(i,2,1)=y1(i)*dn1dxi+y2(i)*dn2dxi+
     1        y3(i)*dn3dxi+y4(i)*dn4dxi+
     2        y5(i)*dn5dxi+y6(i)*dn6dxi+
     3        y7(i)*dn7dxi+y8(i)*dn8dxi
         gmtrx(i,3,1)=z1(i)*dn1dxi+z2(i)*dn2dxi+
     1        z3(i)*dn3dxi+z4(i)*dn4dxi+
     2        z5(i)*dn5dxi+z6(i)*dn6dxi+
     3        z7(i)*dn7dxi+z8(i)*dn8dxi
         gmtrx(i,1,2)=x1(i)*dn1deta+x2(i)*dn2deta+
     1        x3(i)*dn3deta+x4(i)*dn4deta+
     2        x5(i)*dn5deta+x6(i)*dn6deta+
     3        x7(i)*dn7deta+x8(i)*dn8deta
         gmtrx(i,2,2)=y1(i)*dn1deta+y2(i)*dn2deta+
     1        y3(i)*dn3deta+y4(i)*dn4deta+
     2        y5(i)*dn5deta+y6(i)*dn6deta+
     3        y7(i)*dn7deta+y8(i)*dn8deta
         gmtrx(i,3,2)=z1(i)*dn1deta+z2(i)*dn2deta+
     1        z3(i)*dn3deta+z4(i)*dn4deta+
     2        z5(i)*dn5deta+z6(i)*dn6deta+
     3        z7(i)*dn7deta+z8(i)*dn8deta
         gmtrx(i,1,3)=x1(i)*dn1dzeta+x2(i)*dn2dzeta+
     1        x3(i)*dn3dzeta+x4(i)*dn4dzeta+
     2        x5(i)*dn5dzeta+x6(i)*dn6dzeta+
     3        x7(i)*dn7dzeta+x8(i)*dn8dzeta
         gmtrx(i,2,3)=y1(i)*dn1dzeta+y2(i)*dn2dzeta+
     1        y3(i)*dn3dzeta+y4(i)*dn4dzeta+
     2        y5(i)*dn5dzeta+y6(i)*dn6dzeta+
     3        y7(i)*dn7dzeta+y8(i)*dn8dzeta
         gmtrx(i,3,3)=z1(i)*dn1dzeta+z2(i)*dn2dzeta+
     1        z3(i)*dn3dzeta+z4(i)*dn4dzeta+
     2        z5(i)*dn5dzeta+z6(i)*dn6dzeta+
     3        z7(i)*dn7dzeta+z8(i)*dn8dzeta
c
         bmtrx(i,1,1,1) =dn1dxi
         bmtrx(i,1,1,4) =dn2dxi
         bmtrx(i,1,1,7) =dn3dxi
         bmtrx(i,1,1,10)=dn4dxi
         bmtrx(i,1,1,13)=dn5dxi
         bmtrx(i,1,1,16)=dn6dxi
         bmtrx(i,1,1,19)=dn7dxi
         bmtrx(i,1,1,22)=dn8dxi
c
         bmtrx(i,2,1,2) =dn1dxi
         bmtrx(i,2,1,5) =dn2dxi
         bmtrx(i,2,1,8) =dn3dxi
         bmtrx(i,2,1,11)=dn4dxi
         bmtrx(i,2,1,14)=dn5dxi
         bmtrx(i,2,1,17)=dn6dxi
         bmtrx(i,2,1,20)=dn7dxi
         bmtrx(i,2,1,23)=dn8dxi
c
         bmtrx(i,3,1,3) =dn1dxi
         bmtrx(i,3,1,6) =dn2dxi
         bmtrx(i,3,1,9) =dn3dxi
         bmtrx(i,3,1,12)=dn4dxi
         bmtrx(i,3,1,15)=dn5dxi
         bmtrx(i,3,1,18)=dn6dxi
         bmtrx(i,3,1,21)=dn7dxi
         bmtrx(i,3,1,24)=dn8dxi
c
         bmtrx(i,1,2,1) =dn1deta
         bmtrx(i,1,2,4) =dn2deta
         bmtrx(i,1,2,7) =dn3deta
         bmtrx(i,1,2,10)=dn4deta
         bmtrx(i,1,2,13)=dn5deta
         bmtrx(i,1,2,16)=dn6deta
         bmtrx(i,1,2,19)=dn7deta
         bmtrx(i,1,2,22)=dn8deta
c
         bmtrx(i,2,2,2) =dn1deta
         bmtrx(i,2,2,5) =dn2deta
         bmtrx(i,2,2,8) =dn3deta
         bmtrx(i,2,2,11)=dn4deta
         bmtrx(i,2,2,14)=dn5deta
         bmtrx(i,2,2,17)=dn6deta
         bmtrx(i,2,2,20)=dn7deta
         bmtrx(i,2,2,23)=dn8deta
c
         bmtrx(i,3,2,3) =dn1deta
         bmtrx(i,3,2,6) =dn2deta
         bmtrx(i,3,2,9) =dn3deta
         bmtrx(i,3,2,12)=dn4deta
         bmtrx(i,3,2,15)=dn5deta
         bmtrx(i,3,2,18)=dn6deta
         bmtrx(i,3,2,21)=dn7deta
         bmtrx(i,3,2,24)=dn8deta
c
         bmtrx(i,1,3,1) =dn1dzeta
         bmtrx(i,1,3,4) =dn2dzeta
         bmtrx(i,1,3,7) =dn3dzeta
         bmtrx(i,1,3,10)=dn4dzeta
         bmtrx(i,1,3,13)=dn5dzeta
         bmtrx(i,1,3,16)=dn6dzeta
         bmtrx(i,1,3,19)=dn7dzeta
         bmtrx(i,1,3,22)=dn8dzeta
c
         bmtrx(i,2,3,2) =dn1dzeta
         bmtrx(i,2,3,5) =dn2dzeta
         bmtrx(i,2,3,8) =dn3dzeta
         bmtrx(i,2,3,11)=dn4dzeta
         bmtrx(i,2,3,14)=dn5dzeta
         bmtrx(i,2,3,17)=dn6dzeta
         bmtrx(i,2,3,20)=dn7dzeta
         bmtrx(i,2,3,23)=dn8dzeta
c
         bmtrx(i,3,3,3) =dn1dzeta
         bmtrx(i,3,3,6) =dn2dzeta
         bmtrx(i,3,3,9) =dn3dzeta
         bmtrx(i,3,3,12)=dn4dzeta
         bmtrx(i,3,3,15)=dn5dzeta
         bmtrx(i,3,3,18)=dn6dzeta
         bmtrx(i,3,3,21)=dn7dzeta
         bmtrx(i,3,3,24)=dn8dzeta
c
      enddo
c
      return
      end
```

To use the element for a part the section card can be written as

```
*SECTION_SOLID
$    SECID    ELFORM
         1       101
$      NIP     NXDOF      IHGF
         1         0         1
$       XI       ETA      ZETA       WGT
        0.        0.        0.       8.0
```

Examples

We present three test examples.

One example was a simple tension-compression test of a solid cylinder. The geometry is shown in Figure 46-1. The problem is using the sample implementations of user elements and compared the results and performance with standard LS-DYNA elements. As for the computational efficiency, we note that the performance is worse but this is expected since there is little room for optimization of the code while retaining a user friendly interface. The implicit performance compares well with the other elements in LS-DYNA.

The second example was a combined bending and stretching example with the geometry shown in Figure 46-2. Again we ran the problem with the user element implementations and compared the results and performance with standard LS-DYNA elements. We could see the same tendencies as for the solid elements.

The third and final example is an impact between a solid bar and shell beam. Both parts are modeled with user-defined elements. The results were very similar to the ones obtained by substituting the sections for standard LS-DYNA sections, but the simulation time was about 3-4 times longer.

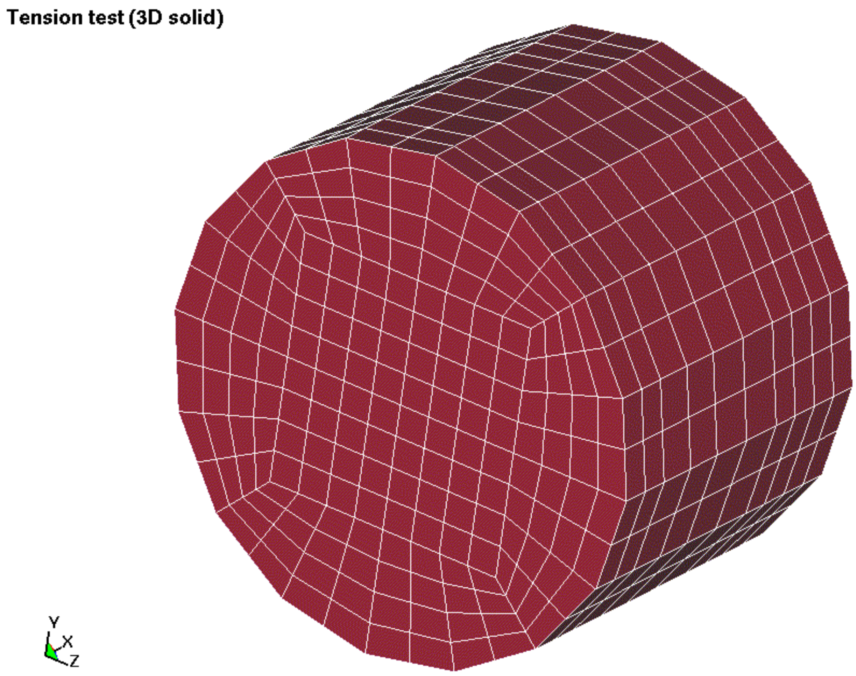


Figure 46-1. Solid mesh for user element test.

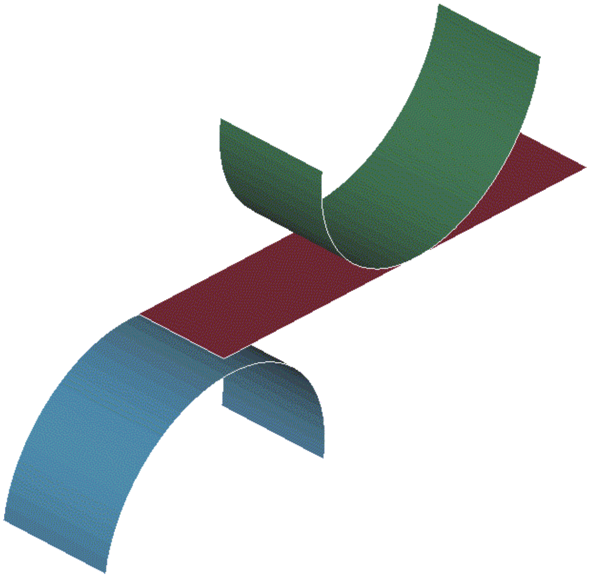


Figure 46-2. Shell mesh for the user element test.

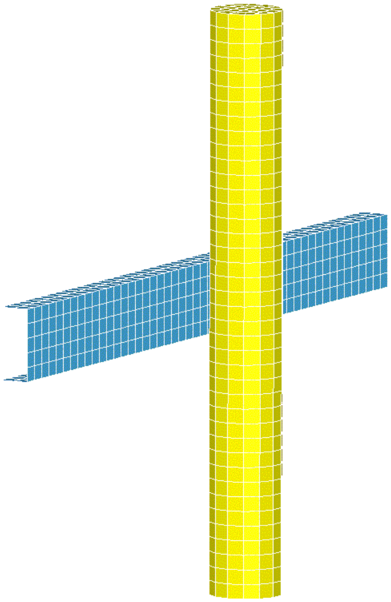


Figure 46-3. Impact between a user-defined shell and user-defined solid part.

APPENDIX D: User Defined Airbag Sensor

The addition of a user sensor subroutine into LS-DYNA is relatively simple. The sensor is mounted on a rigid body which is attached to the structure. The motion of the sensor is provided in the local coordinate system defined for the rigid body in the definition of material model 20–the rigid material. When the user defined criterion is met for the deployment of the airbag, a flag is set and the deployment begins. All load curves relating to the mass flow rate versus time are then shifted by the initiation time. The user subroutine is given below with all the necessary information contained in the comment cards.

```
      subroutine airusr (rbu,rbv,rba,time,dt1,dt2,param,hist,itrnon,
     .  rbug,rbvg,rbag,icnv)
c
c******************************************************************
c|  Livermore Software Technology Corporation  (LSTC)             |
c|  ------------------------------------------------------------  |
c|  Copyright 1987-2008 Livermore Software Tech.  Corp             |
c|  All rights reserved                                           |
c******************************************************************
c
c     user subroutine to initiate the inflation of the airbag
c
c     variables
c
c     displacements are defined at time n+1 in local system
c     velocites are defined at time n+1/2 in local system
c     accelerations are defined at time n in local system
c
c         rbu(1-3) total displacements in the local xyz directions
c         rbu(3-6) total rotations about the local xyz axes
c         rbv(1-3) velocities in the local xyz directions
c         rbv(3-6) rotational velocities about the local xyz axes
c         rba(1-3) accelerations in the local xyz directions
c         rba(3-6) rotational accelerations about the local xyz axes
c         time is the current time
c         dt1 is time step size at n-1/2
c         dt2 is time step size at n+1/2
c         param is user defined input parameters
c         hist is user defined history variables
c         itrnon is a flag to turn on the airbag inflation
c         rbug,rbvg,rbag, are similar to rbu,rbv,rba but are defined
c                         globally.
c         icnv is the airbag ID
c
c     the user subroutine sets the variable itrnon to:
c
c           itrnon=0 bag is not inflated
c           itrnon=1 bag inflation begins and this subroutine is
c                    not called again
c
      include 'iounits.inc'
      dimension rbu(6),rbv(6),rba(6),param(25),hist(25),
     .  rbug(6),rbvg(6),rbag(6)
c
      itrnon=0
      ra=sqrt(rba(1)**2+rba(2)**2+rba(3)**2)
      if (ra.gt.param(1)) then
```

```
        itrnon=1
        write(iotty,100) time
        write(iohsp,100) time
        write(iomsg,100) time
      endif
 100  format (' Airbag activated at time ',1pe10.3)
c
      return
      end
```

APPENDIX E: User Defined Solution Control

This subroutine may be provided by the user to control the I/O, monitor the energies and other solution norms of interest, and to shut down the problem whenever he pleases. The arguments are defined in the listing provided below. This subroutine is called each time step and does not need any control card to operate.

```
      subroutine uctrl1 (numnp,ndof,time,dt1,dt2,prtc,pltc,frci,prto,
     .  plto,frco,vt,vr,at,ar,ut,ur,xmst,xmsr,irbody,rbdyn,usrhv,
     .  messag,totalm,cycle,idrint,mtype,mxrb,nrba,rbcor,x,rbv,nrbn,
     .  nrb,xrb,yrb,zrb,axrb,ayrb,azrb,dtx,nmmat,rba,fvalnew,fvalold,
     .  fvalmid,fvalnxt)
c
c*************************************************************
c|  Livermore Software Technology Corporation  (LSTC)          |
c|  ------------------------------------------------------     |
c|  Copyright 1987-2008 Livermore Software Tech.  Corp          |
c|  All rights reserved                                       |
c*************************************************************
c
c     user subroutine for solution control and is called at the
c     beginning of time step n+1.   The time at n+
c
c
c     note:  ls-dyna3d uses an internal numbering system to
c            accomodate arbitrary node numbering.   to access
c            information for user node n, address array location m,
c            m = lqf8(n,1).  to obtain user node number, n,
c            corresponding to array address m, set n = lqfinv(m,1)
c
c     arguments:
c          numnp = number of nodal points
c          ndof = number of degrees of freedom per node
c          time = current solution time at n+1
c          dt1  = time step size between time n-1 and n
c          dt2  = time step size between time n and n+1
c          prtc = output interval for taurus time history data
c          pltc = output interval for taurus state data
c          frci = output interval for taurus interface force data
c          prto = output time for time history file
c          plto = output time for state data
c          frco = output time for force data
c          vt(3,numnp) = nodal translational velocity vector
c          vr(3,numnp) = nodal rotational velocity vector.  this
c                        array is defined if and only if ndof = 6
c          at(3,numnp) = nodal translational acceleration vector
c          ar(3,numnp) = nodal rotational acceleration vector.  this
c                        array is defined if and only if ndof = 6
c          ut(3,numnp) = nodal translational displacement vector
c          ur(3,numnp) = nodal rotational displacement vector.  this
c                        array is defined if and only if ndof = 6
c          xmst(numnp) = reciprocal of nodal translational masses
c          xmsr(numnp) = reciprocal of nodal rotational masses.  this
c                        array is defined if and only if ndof = 6
```

```
c          fvalold     = array for storing load curve values at time n
c          fvalnew     = array for setting load curve values at time n+1
c                        only load curves with 0 input points may be user
c                        defined.  When the load curve is user set, the
c                        value at time n must be stored in array fvalold.
c          fvalmid     = array for predicting load curve values at time n+3/2
c          fvalnxt     = array for predicting load curve values at time n+2
c                        for some applications it is necessary to predict the
c                           load curve values at time n+2, a time that is not
known,
c                        this for instance for boundary prescribed motion.  In
c                        this case the load curve values at time n+3/2 need to
c                        be predicted in fvalmid, and fvalold should be set to
c                        fvalnew and fvalnew should be set to fvalnxt.  See
c                        coding below.
c
c          irbody      = 0 if no rigid bodies
c          rbdyn(numnp)=flag for rigid body nodal points
c                        if deformable node then set to 1.0
c                        if rigid body node then set to 0.0
c                        defined if and only if rigid bodies are present
c                        i.e., irbody.ne.0 if no rigid bodies are
c                        present
c          usrhv(lenhv)=user defined history variables that are stored
c                        in the restart file.  lenhv = 100+7*nummat where
c                        nummat is the # of materials in the problem.
c                        array usrhv is updated only in this subroutine.
c          messag      = flag for dyna3d which may be set to:
c                       'sw1.' ls-dyna3d terminates with restart file
c                       'sw3.' ls-dyna3d writes a restart file
c                      'sw4.' ls-dyna3d writes a plot state
c           totalm     = total mass in problem
c           cycle      = cycle number
c           idrint     = flag for dynamic relaxation phase
c                        .ne.0: dynamic relaxation in progress
c                        .eq.0: solution phase
c
      include 'ptimes.inc'
c
c     prtims(1-37)=output intervals for ascii files
c
c      ascii files:
c           ( 1)-cross section forces
c           ( 2)-rigid wall forces
c           ( 3)-nodal data
c           ( 4)-element data
c           ( 5)-global data
c           ( 6)-discrete elements
c           ( 7)-material energies
c           ( 8)-noda interface forces
c           ( 9)-resultant interface forces
c           (10)-smug animator
c           (11)-spc reaction forces
c           (12)-nodal constraint resultant forces
c           (13)-airbag statistics
c           (14)-avs database
c           (15)-nodal force groups
c           (16)-output intervals for nodal boundary conditions
c           (17)-(32) unused at this time
c           (37)-auto tiebreak damage output
```

```
c
c     prtlst(32)=output times for ascii files above.  when solution time
c                exceeds the output time a print state is dumped.
c
      common/rbkeng/enrbdy,rbdyx,rbdyy,rbdyz
c
c      total rigid body energies and momentums:
c           enrbdy = rigid body kinetic energy
c           rbdyx = rigid body x-momentum
c           rbdyy = rigid body y-momentum
c           rbdyz = rigid body z-momentum
c
      common/swmke/swxmom,swymom,swzmom,swkeng
c
c      total stonewall energies and momentums:
c           swxmom = stonewall x-momentum
c           swymom = stonewall y-momentum
c           swzmom = stonewall z-momentum
c           swkeng = stonewall kinetic energy
c
      common/deengs/deeng
c
c      deeng = total discrete element energy
c
      common/bk28/summss,xke,xpe,tt,xte0,erodeke,erodeie,selie,selke,
     .  erodehg
c
c      xpe  = total internal energy in the finite elements
c
      common/sprengs/spreng
c
c      spreng = total spr energy
c
      character*(*) messag
      integer cycle
      real*8 x
      dimension vt(3,*),vr(3,*),at(3,*),ar(3,*),
     .  xmst(*),xmsr(*),rbdyn(*),usrhv(*),mtype(*),mxrb(*),nrba(*),
     .  rbcor(3,1),x(*),rbv(6,*),nrbn(*),nrb(*),xrb(*),yrb(*),
     .  zrb(*),axrb(*),ayrb(*),azrb(*),rba(6,*),fvalnew(*),
     .  fvalold(*),fvalmid(*),fvalnxt(*)
      real*8 ut(3,*),ur(3,*)
c
c     sample momentum and kinetic energy calculations
c
c     remove all comments in column 1 below to activate
      i = 1
      if (i.eq.1) return
c     return
cc
cc
cc     initialize kinetic energy, xke, and x,y,z momentums.
cc
c      xke = 2.*swkeng+2.*enrbdy
c      xm = swxmom+rbdyx
c      ym = swymom+rbdyy
c      zm = swzmom+rbdyz
cc
c      numnp2 = numnp
c      if (ndof.eq.6) then
```

```
c        numnp2 = numnp+numnp
c      endif
c      write(iotty,*)ndof
cc
cc
cc     no rigid bodies present
cc
c      if (irbody.eq.0) then
cc       note in blank comment vr follows vt.  this fact is used below.
c        do 10 n = 1,numnp2
c        xmsn = 1./xmst(n)
c        vn1 = vt(1,n)
c        vn2 = vt(2,n)
c        vn3 = vt(3,n)
c        xm = xm+xmsn*vn1
c        ym = ym+xmsn*vn2
c        zm = zm+xmsn*vn3
c        xke = xke+xmsn*(vn1*vn1+vn2*vn2+vn3*vn3)
c   10   continue
cc
cc
cc     rigid bodies present
cc
c      else
cc       nodal accerations for rigid bodies
cc
c        do 12 n = 1,nmmat
c        if (mtype(n).ne.20.or.mxrb(n).ne.n) go to 12
c        lrbn = nrba(n)
c        call stvlut(rbcor(1,n),x,vt,at,ar,vr,rbv(1,n),dt2,
c    .   nrbn(n),nrb(lrbn),xrb,yrb,zrb,axrb,ayrb,azrb,dtx)
c
c        rigid body nodal accelerations
c
c        if (ndof.eq.6) then
c          call rbnacc(nrbn(n),nrb(lrbn),rba(4,n),ar)
c        endif
c
c  12    continue
cc
c        do 20 n = 1,numnp
c        xmsn = 1./xmst(n)
c        vn1 = rbdyn(n)*vt(1,n)
c        vn2 = rbdyn(n)*vt(2,n)
c        vn3 = rbdyn(n)*vt(3,n)
c        xm = xm+xmsn*vn1
c        ym = ym+xmsn*vn2
c        zm = zm+xmsn*vn3
c        xke = xke+xmsn*(vn1*vn1+vn2*vn2+vn3*vn3)
c   20   continue
c        if (ndof.eq.6) then
c          do 30 n = 1,numnp
c          xmsn = 1./xmsr(n)
c          vn1 = rbdyn(n)*vr(1,n)
c          vn2 = rbdyn(n)*vr(2,n)
c          vn3 = rbdyn(n)*vr(3,n)
c          xm = xm+xmsn*vn1
c          ym = ym+xmsn*vn2
c          zm = zm+xmsn*vn3
c          xke = xke+xmsn*(vn1*vn1+vn2*vn2+vn3*vn3)
```

```
c   30      continue
c         endif
c       endif
cc
cc      total kinetic energy
c       xke=.5*xke
cc      total internal energy
c       xie = xpe+deeng+spreng
cc      total energy
c       xte = xke+xpe+deeng+spreng
cc      total x-rigid body velocity
c       xrbv = xm/totalm
cc      total y-rigid body velocity
c       yrbv = ym/totalm
cc      total z-rigid body velocity
c       zrbv = zm/totalm
        return
        end
```

APPENDIX F: User Defined Interface Control

This subroutine may be provided by the user to turn the interfaces on and off. This option is activated by the *USER_INTERFACE_CONTROL keyword. The arguments are defined in the listing provided below.

```
      subroutine uctrl2(nsi,nty,time,cycle,msr,nmn,nsv,nsn,
     1 thmr,thsv,vt,xi,ut,iskip,idrint,numnp,dt2,ninput,ua,
     2 irectm,nrtm,irects,nrts)
c
c******************************************************************
c|  Livermore Software Technology Corporation  (LSTC)             |
c|  ------------------------------------------------------------  |
c|  Copyright 1987-2008 Livermore Software Tech.  Corp             |
c|  All rights reserved                                           |
c******************************************************************
c
c     user subroutine for interface control
c
c     note:  ls-dyna3d uses an internal numbering system to
c            accomodate arbitrary node numbering.   to access
c            information for user node n, address array location m,
c            m = lqf8(n,1).  to obtain user node number, n,
c            corresponding to array address m, set n = lqfinv(m,1)
c
c     arguments:
c          nsi        = number of sliding interface
c          nty        = interface type.
c                        .eq.4:single surface
c                        .ne.4:surface to surface
c          time       = current solution time
c          cycle      = cycle number
c          msr(nmn)   = list of master nodes numbers in internal
c                       numbering scheme
c          nmn        = number of master nodes
c          nsv(nsn)   = list of slave nodes numbers in internal
c                       numbering scheme
c          nsn        = number of slave nodes
c          thmr(nmn)  = master node thickness
c          thsv(nsn)  = slave node thickness
c          vt(3,numnp)=nodal translational velocity vector
c          xi(3,numnp)=initial coordinates at time = 0
c          ut(3,numnp)=nodal translational displacement vector
c          idrint     = flag for dynamic relaxation phase
c                        .ne.0: dynamic relaxation in progress
c                        .eq.0: solution phase
c          numnp      = number of nodal points
c          dt2        = time step size at n+1/2
c          ninput     = number of variables input into ua
c          ua(*)      = users' array, first ninput locations
c                       defined by user.  the length of this
c                       array is defined on control card 10.
c                       this array is unique to interface nsi.
c          irectm(4,*)=list of master segments in internal
```

```
c                      numbering scheme
c          nrtm       = number of master segments
c          irects(4,*)=list of slave segments in internal
c                      numbering scheme
c          nrts       = number of master segments
c
c     set flag for active contact
c       iskip = 0 active
c       iskip = 1 inactive
c
c*******************************************************************
c
      integer cycle
      real*8 ut
      real*8 xi
      dimension msr(*),nsv(*),thmr(*),thsv(*),vt(3,*),xi(3,*),
     .          ut(3,*),ua(*),irectm(4,*),irects(4,*)
c
c     the following sample of codeing is provided to illustrate how
c     this subroutine might be used.   here we check to see if the
c     surfaces in the surface to surface contact are separated.   if
c     so the iskip = 1 and the contact treatment is skipped.
c
c     if (nty.eq.4) return
c     dt2hlf = dt2/2.
c     xmins = 1.e+16
c     xmaxs = -xmins
c     ymins = 1.e+16
c     ymaxs = -ymins
c     zmins = 1.e+16
c     zmaxs = -zmins
c     xminm = 1.e+16
c     xmaxm = -xminm
c     yminm = 1.e+16
c     ymaxm = -yminm
c     zminm = 1.e+16
c     zmaxm = -zminm
c     thks = 0.0
c     thkm = 0.0
c     do 10 i = 1,nsn
c     dsp1 = ut(1,nsv(i))+dt2hlf*vt(1,nsv(i))
c     dsp2 = ut(2,nsv(i))+dt2hlf*vt(2,nsv(i))
c     dsp3 = ut(3,nsv(i))+dt2hlf*vt(3,nsv(i))
c     x1 = xi(1,nsv(i))+dsp1
c     x2 = xi(2,nsv(i))+dsp2
c     x3 = xi(3,nsv(i))+dsp3
c     thks = max(thsv(i),thks)
c     xmins = min(xmins,x1)
c     xmaxs = max(xmaxs,x1)
c     ymins = min(ymins,x2)
c     ymaxs = max(ymaxs,x2)
c     zmins = min(zmins,x3)
c     zmaxs = max(zmaxs,x3)
c  10 continue
c     do 20 i = 1,nmn
c     dsp1 = ut(1,msr(i))+dt2hlf*vt(1,msr(i))
c     dsp2 = ut(2,msr(i))+dt2hlf*vt(2,msr(i))
c     dsp3 = ut(3,msr(i))+dt2hlf*vt(3,msr(i))
c     x1 = xi(1,msr(i))+dsp1
c     x2 = xi(2,msr(i))+dsp2
```

```
c     x3 = xi(3,msr(i))+dsp3
c     thkm = max(thmr(i),thks)
c     xminm = min(xminm,x1)
c     xmaxm = max(xmaxm,x1)
c     yminm = min(yminm,x2)
c     ymaxm = max(ymaxm,x2)
c     zminm = min(zminm,x3)
c     zmaxm = max(zmaxm,x3)
c  20 continue
c
c     if thks or thkm equal zero set them to some reasonable value
c
c     if (thks.eq.0.0) then
c       e1=(xi(1,irects(1,1))-xi(1,irects(3,1)))**2
c    .    +(xi(2,irects(1,1))-xi(2,irects(3,1)))**2
c    .    +(xi(3,irects(1,1))-xi(3,irects(3,1)))**2
c       e2=(xi(1,irects(2,1))-xi(1,irects(4,1)))**2
c    .    +(xi(2,irects(2,1))-xi(2,irects(4,1)))**2
c    .    +(xi(3,irects(2,1))-xi(3,irects(4,1)))**2
c       thks=.3*sqrt(max(e1,e2))
c     endif
c     if (thkm.eq.0.0) then
c       e1=(xi(1,irectm(1,1))-xi(1,irectm(3,1)))**2
c    .    +(xi(2,irectm(1,1))-xi(2,irectm(3,1)))**2
c    .    +(xi(3,irectm(1,1))-xi(3,irectm(3,1)))**2
c       e2=(xi(1,irectm(2,1))-xi(1,irectm(4,1)))**2
c    .    +(xi(2,irectm(2,1))-xi(2,irectm(4,1)))**2
c    .    +(xi(3,irectm(2,1))-xi(3,irectm(4,1)))**2
c       thkm=.3*sqrt(max(e1,e2))
c     endif
c
c     if (xmaxs+thks.lt.xminm-thkm) go to 40
c     if (ymaxs+thks.lt.yminm-thkm) go to 40
c     if (zmaxs+thks.lt.zminm-thkm) go to 40
c     if (xmaxm+thkm.lt.xmins-thks) go to 40
c     if (ymaxm+thkm.lt.ymins-thks) go to 40
c     if (zmaxm+thkm.lt.zmins-thks) go to 40
c     iskip = 0
c
c     return
c  40 iskip = 1
c
      return
      end
```

APPENDIX G: User Defined Interface Friction and Conductivity

An easy-to-use user contact interface is provided in LS-DYNA where the user has the possibility to define the frictional coefficients (static and dynamic) as well as contact heat transfer conductance as functions of contact pressure, relative sliding velocity, separation and temperature. To be able to use this feature, an object version of the LS-DYNA code is required and the user must write his/her own Fortran (or C) code to define the contact parameters of interest.

In the text file dyn21.f that comes with the object version of LS-DYNA, the subroutines of interest are

```
      subroutine usrfrc(fstt,fdyn,...)
```

for defining the frictional coefficients fstt (static) and fdyn (dynamic) and

```
      subroutine usrhcon(h,...)
```

for defining the heat transfer contact conductance h.

We emphasize at this point that the user friction interface differs between LS-DYNA (SMP) and MPP-DYNA (MPP), for reasons that have to do with how the contacts are implemented in general. In LS-DYNA (SMP) the user is required not only to define the frictional coefficients but also to assemble and store contact forces and history, whereas in MPP-DYNA (MPP) only the frictional coefficients have to be defined.

For the friction interface (SMP and MPP) the user may associate history variables with each contact node. Unfortunately, the user friction interface is currently not supported by all available contacts in LS-DYNA and MPP-DYNA, but it should cover the most interesting ones among others, *CONTACT_(FORMING_)NODES_TO_SURFACE, *CONTACT_(FORMING_)SURFACE_TO_SURFACE, *CONTACT_(FORMING_)ONE_WAY_SURFACE_TO_SURFACE. Upon request by customers additional contact types can be supported.

One of the arguments to the user contact routines is the curve array `crv`, also available in the user material interface. Note that when using this array, the curve identity must be converted to an internal number or the subroutine `crvval` may be utilized. For more information, see the appendix A on user materials.

For definition of user contact parameters the user must define the keywords

*USER_INTERFACE_FRICTION

or

*USER_INTERFACE_CONDUCTIVITY

The card format for these two keywords are identical and can be found in other sections in this manual.

There is an alternate route to defining the conductivity parameters for a user defined thermal contact. On the *CONTACT_..._THERMAL_FRICTION optional card the parameter FORMULA may be set to a negative number. This will automatically create a user defined conductivity interface and invoke reading of –FORMULA contact parameters immediately following the card including the FORMULA parameter. Note that FORMULA is related to NOC and NOCI in the *USER_INTERFACE_CONDUCTIVITY keyword as

$$-\text{FORMULA} = \text{NOC} = \text{NOCI}.$$

Note that the pressure is automatically computed for each user conductivity interface, i.e., the keyword *LOAD_SURFACE_STRESS is not necessary.

A sample friction subroutine is provided below for SMP.

```
      subroutine usrfrc(nosl,time,ncycle,dt2,insv,areas,xs,ys,zs,
     .  lsv,ix1,ix2,ix3,ix4,aream,xx1,xx2,xx3,stfn,stf,fni,
     .  dx,dy,dz,fdt2,ninput,ua,side,iisv5,niisv5,n1,n2,n3,fric1,
     .  fric2,fric3,fric4,bignum,fdat,iseg,fxis,fyis,fzis,ss,tt,
     .  ilbsv,stfk,frc,numnp,npc,pld,lcfst,lcfdt,temp,temp_bot,
     .  temp_top,isurface)
c
c******************************************************************
c|  LIVERMORE SOFTWARE TECHNOLOGY CORPORATION  (LSTC)             |
c|  ------------------------------------------------------------  |
c|  COPYRIGHT © 1987-2007 JOHN O.  HALLQUIST, LSTC                 |
c|  ALL RIGHTS RESERVED                                            |
c******************************************************************
c
c     user subroutine for interface friction control
c
c     note:  LS-DYNA uses an internal numbering system to
c            accomodate arbitrary node numbering.   to access
c            information for user node n, address array location m,
c            m=lqf(n,1).  to obtain user node number, n,
c            corresponding to array address m, set n=lqfinv(m,1)
c
c     arguments:
c
c          nosl       =number of sliding interface
c          time       =current solution time
c          ncycle     =ncycle number
c          dt2        =time step size at n+1/2
c          insv       =slave node array where the nodes are stored
c                      in ls-dyna3d internal numbering.  User numbers
c                      are given by function: lqfinv(insv(ii),1)
c                      for slave node ii.
c          areas(ii)  =slave node area (interface types 5&10 only) for
c                      slave node ii
c          xs(ii)     =x-coordinate slave node ii (projected)
c          ys(ii)     =y-coordinate slave node ii (projected)
c          zs(ii)     =z-coordinate slave node ii (projected)
c          lsv(ii)    =master segment number for slave node ii
c          ix1(ii), ix2(ii), ix3(ii), ix4(ii)
c                     =master segment nodes in ls-dyna3d internal
c                      numbering for slave node ii
```

```
c          aream(ii)  =master segment area for slave node ii.
c          xx1(ii,4)  =x-coordinates master surface (projected) for
c                      slave node ii
c          xx2(ii,4)  =y-coordinates master surface (projected) for
c                      slave node ii
c          xx3(ii,4)  =z-coordinates master surface (projected) for
c                      slave node ii
c          stfn       =slave node penalty stiffness
c          stf        =master segment penalty stiffness
c          fni        =normal force
c          dx,dy,dz   =relative x,y,z-displacement between slave node and
c                      master surface.  Multipling by fdt2 defines the
c                      relative velocity.
c          n1,n2,n3   =x,y, and z components of master segments normal
c                      vector
c
c***********************************************************************
c       frictional coefficients defined for the contact interface
c
c          fric1      =static friction coefficient
c          fric2      =dynamic friction coefficient
c          fric3      =decay constant
c          fric4      =viscous friction coefficient (setting fric4=0
c                      turns this option off)
c
c***********************************************************************
c
c          bignum     =0.0 for one way surface to surface and
c                      for surface to surface, and 1.e+10 for nodes
c                      to surface contact
c          ninput     =number of variables input into ua
c          ua(*)      =users' array, first ninput locations
c                      defined by user.  the length of this
c                      array is defined on control card 10.
c                      this array is unique to interface nosl.
c
c          side       ='master' for first pass.  the master
c                       surface is the surface designated in the
c                       input
c                     ='slave' for second pass after slave and
c                      master surfaces have be switched for
c                      the type 3 symmetric interface treatment.
c
c          iisv5      =an array giving the pointers to the active nodes
c                      in the arrays.
c
c          niisv5     =number of active nodes
c
c          fdat       =contact history data array
c          iseg       =contact master segment from previous step.
c          fxis       =slave node force component in global x dir.
c                      to be updated to include friction
c          fyis       =slave node force component in global y dir.
c                      to be updated to include friction
c          fzis       =slave node force component in global z dir.
c                      to be updated to include friction
c          ss(ii)     =s contact point (-1 to 1) in parametric coordinates
c                      for slave node ii.
c          tt(ii)     =t contact point (-1 to 1) in parametric coordinates
c                      for slave node ii.
c          ilbsv(ii)  =pointer for node ii into global arrays.
c          stfk(ii)   =penalty stiffness for slave node ii which was used
c                      to compute normal interface force.
c          frc(1,lsv(ii))
c                    =Coulomb friction scale factor for segment lsv(ii)
c          frc(2,lsv(ii))
```

```
c                    =viscous friction scale factor for segment lsv(ii)
c
c***********************************************************************
c       parameters for a coupled thermal-mechanical contact
c
c          numnp      = number of nodal points in the model
c          npc        = load curve pointer
c          pld        = load curve (x,y) data
c          lcfst(nosl)= load curve number for static coefficient of
c                       friction versus temperture for contact
c                       surface nosl
c          lcfdt(nosl)= load curve number for dynamic coefficient of
c                       friction versus temperture for contact
c                       surface nosl
c          temp(j)    = temperature for node point j
c          temp_bot(j)= temparature for thick thermal shell bottom
c                       surface
c          temp_top(j)= temparature for thick thermal shell top
c                       surface
c          numsh12    = number of thick thermal shells
c          itopaz(1)  = 999 ==> thermal-mechanical analysis
c          isurface   = thick thermal shell surface pointer
c
c*******************************************************************************
```

APPENDIX H: User Defined Thermal Material Model

The addition of a thermal user material routine into LS-DYNA is fairly straightforward. The thermal user material is controlled using the keyword *MAT_THERMAL_USER_DEFINED, which is described at the appropriate place in the manual.

The thermal user material can be used alone or in conjunction with any given mechanical material model in a coupled thermal-mechanical solution. A heat-source can be included and the specific heat updated so that it possible to model e.g. phase transformations including melt energy.

If for the same part (shell or solid elements) both a thermal and mechanical user material model is defined then the two user material models have (optionally) read access to each other's history variables. If the integration points of the thermal and mechanical elements not are coincident then interpolation or extrapolation is used when reading history variables. Linear interpolation or extrapolation using history data from the two closest integration points is used in all cases except when reading history variables from the thick thermal shell (THSHEL = 1 on *CONTROL_SHELL). For the latter thermal shell, the shape functions of the element are used for the interpolation or extrapolation.

The thermal user materials are thermal material types 11-15. These thermal user material subroutines are defined in file dyn21.f as subroutines thumat11, … , thumat15. The latter subroutines are called from the subroutine thusrmat. The source code of subroutine thusrmat is also in file dyn21.f. Additional useful information is available in the comments of subroutines thusrmat, thumat12, and umat46 that all reside in the source file dyn21.f

Thermal history variables

Thermal history variables can be used by setting NVH greater that 0. Thermal history variables are output to the tprint file, see *DATABASE_TPRINT.

Interchange of history variables with mechanical user material

In a coupled thermo-mechanical solution there is for each mechanical shell, thick shell, or solid element a corresponding thermal element. A pair consisting of a mechanical and a corresponding thermal element both have integration points and possibly history variables. The mechanical and thermal elements do not necessarily have the same number of integration points.

By setting IHVE to 1, a thermal user material model can read, but not write, the history variables from a mechanical user material model and vice versa.

If the locations of the points where the history variables are located differ between the mechanical and thermal element differ interpolation or extrapolation is used to calculate the history value. More information is available in the comments to the subroutines thusrmat and thumat11.

Limitations:

Currently there are a few limitations of the thermal user material implementation. LS-DYNA will in most cases give an appropriate warning or error message when such a limit is violated. The limitations include:

1. Option IHVE.EQ.1 is only supported for a limited range of mechanical elements:

    a) Solid elements: ELFORM = 1, 2, 10, 13.

    b) Shell elements: ELFORM = 2, 3, 4, 16. Note that user-defined integration rules are not supported.

2. Thermal history variables limitations:

    a) Thermal history variables are not output to d3plot.

3. The thermal solver includes not only the plastically dissipated energy as a heat source but also wrongly the elastic energy. The latter however is in most cases not of practical importance.

Example source code:

Example source code for thermal user material models is available in thumat11 and thumat12 as well as in umat46. Note that there is space for up to 64 material parameters in r_matp (material parameter array) but only 32 can be read in from the *MAT_THERMAL_USER_DEFINED card. The material parameters in r_matp(i), i = 41-64, which are initially set to 0.0, may be used by the user to store additional data.

Subroutine crvval evaluates load curves. Note that when using crvval the load curves are first re-interpolated to 100 equidistant points. See Appendix A for more information on subroutine crvval.

Following is a short thermal user material model. The card format is in this case, if enabling orthotropic conduction, and with sample input in SI-units:

*MAT_THERMAL_USER_DEFINED

Card 1	1	2	3	4	5	6	7	8
Variable	MID	RO	MT	LMC	NVH	AOPT	IORTHO	IHVE
Type	21	7800.0	12.0	6.0	3.0	0.0	1.0	0.0

Card 2	1	2	3	4	5	6	7	8
Variable	XP	YP	ZP	A1	A2	A3		
Type	0.0	0.0	0.0	0.0	0.0	0.0		

Card 3	1	2	3	4	5	6	7	8
Variable	D1	D2	D3					
Type	0.0	0.0	0.0					

Card 4	1	2	3	4	5	6	7	8
Variable	C1	C2	C3	HC	HSRC	HCFAC		
Type	25.0	25.0	20.0	470.0	11.0	12.0		

VARIABLE	DESCRIPTION
C1-C3	Heat conduction in 11, 22, and 33 direction of material coordinate system.
HC	Heat capacity
HSRC	Load curve ID of load curve giving a heat source output (W/m3) as a function of time.
HCFAC	Load curve ID of load curve giving a scaling of the heat capacity as function of time.

The source code is:

```
      subroutine thumat12(c1,c2,c3,cvl,dcvdtl,hsrcl,dhsrcdtl,
     1     hsv,hsvm,nmecon,r_matp,crv,
     2     nel,nep,iep,eltype,dt,atime,ihsrcl)
      character*(*) eltype
      dimension hsv(*),hsvm(*),r_matp(*),crv(101,2,*)
      include 'iounits.inc'
c
c     Thermal user-material number 12.
c
c     See comments at the beginning of subroutine thusrmat
c     for instructions.
c
c     Example: isotropic/orthotropic material with k1=P1 and
c     cvl=P2 for solid and shell elements including optional
c     change of heat capacity and a heat source, both functions
c     of time input as load curves.
c
c     Print out some info on start-up, use material parameter 64
c     as a flag.
      if(nint(r_matp(64)).eq.0) then
         r_matp(64)=1.
         write( *,1200)  (r_matp(8+i),i=1,6)
         write(iohsp,1200) (r_matp(8+i),i=1,6)
         write(59,1200) (r_matp(8+i),i=1,6)
      endif
c
c     Calculate response
      c1=r_matp(8+1)
      c2=r_matp(8+2)
      c3=r_matp(8+3)
      cvl=r_matp(8+4)
      dcvdtl=0.0
      eid=nint(r_matp(8+6))
      if(nint(eid).gt.0) then
         call crvval(crv,eid,atime,cvlfac,tmp1)
         cvl=cvl*cvlfac
         dcvdtl=0.0
      endif
c
c     If flux or time step calculation then we are done.
      if(eltype.eq.'soliddt'.or.eltype.eq.'flux'.or.
     .     eltype.eq.'shelldt') return
      eid=nint(r_matp(8+5))
      if(nint(eid).gt.0) then
         ihsrcl=1
         call crvval(crv,eid,atime,hsrcl,tmp1)
         dhsrcdtl=0.0
      endif
c
c     Update history variables
      hsv(1)=cvl
      hsv(2)=atime
      hsv(3)=hsv(3)+1.0
c
c     Done
      return
 1200 format(/'This is thermal user defined material #12.  '/
     1        /' Material parameter c1-c3       : ',3E10.3
     2        /' Material parameter hc          : ',E10.3
     3        /' Heat source load curve         : ',F10.0
     4        /' hc scale factor load curve     : ',F10.0
     5        /' Thermal History variable 1     : cv'
     6        /' Thermal History variable 2-3   : Dummy'/)
```

```
      return
      end
```

APPENDIX I: Occupant Simulation Including the Coupling to the CAL3D and MADYMO programs

INTRODUCTION

LS-DYNA is coupled to occupant simulation codes to generate solutions in automotive crashworthiness that include occupants interacting with the automotive structure. In such applications LS-DYNA provides the simulation of the structural and deformable aspects of the model and the OSP (Occupant Simulation Program) simulates the motion of the occupant. There is some overlap between the two programs which provides flexibility in the modeling approach. For example, both the OSP and LS-DYNA have the capability of modeling seat belts and other deformable restraints. The advantage of using the OSP is related to the considerable databases and expertise that have been developed in the past for simulating dummy behavior using these programs.

The development of the interface provided LSTC a number of possible approaches. The approach selected is consistent with the LSTC philosophy of providing the most flexible and useful interface possible. This is important because the field of non-linear mechanics is evolving rapidly and techniques which are used today are frequently rendered obsolete by improved methodologies and lower cost computing which allows more rigorous techniques to be used. This does make the learning somewhat more difficult as there is not any single procedure for performing a coupling.

One characteristic of LS-DYNA is the large number of capabilities, particularly those associated with rigid bodies. This creates both an opportunity and a difficulty: LS-DYNA3D has many ways approximating different aspects of problems, but they are frequently not obvious to users without considerable experience. Therefore, in this Appendix we emphasize modeling methods rather than simply listing capabilities.

THE LS-DYNA/OCCUPANT SIMULATION PROGRAM LINK

Coupling between the OSP and LS-DYNA is performed by combining the programs into a single executable. In the case of CAL3D, LS-DYNA calls CAL3D as a subroutine, but in the case of MADYMO, LS-DYNA is called as a subroutine. The two programs are then integrated in parallel with the results being passed between the two until a user defined termination time is reached.

The OSP and LS-DYNA have different approaches to the time integration schemes. The OSP time integrators are based on accurate implicit integrators which are valid for large time steps which are on the order of a millisecond for the particular applications of interest here. An iterative solution is used to insure that the problem remains in equilibrium. The implicit integrators are extremely good for smoothly varying loads, however, sharp nonlinear pulses can introduce considerable error. An automatic time step size control which decreases the time step size quickly restores the accuracy for such events. The LS-DYNA time integrator is based on an explicit central difference scheme. Stability requires that the time step size be less than the highest frequency in the system. For a coarse airbag mesh, this number is on the order of 100 microseconds while an actual car crash simulation is on the order of 1 microsecond. The smallest LS-DYNA models have at least 1,000 elements. Experience indicates that the cost of a single LS-DYNA time step for a small model is at least as great as the cost of a time step in the OSP. Therefore, in the coupling, the LS-DYNA time step is used to control the entire simulation including the OSP part. This approach has negligible cost penalties and avoids questions of stability and accuracy that would result by using a subcycling scheme between the two programs.

LS-DYNA has a highly developed rigid body capability which is used in different parts of automobile crash simulation. In particular, components such as the engine are routinely modeled with rigid bodies. These rigid bodies have been modified so that they form the basis of the coupling procedure in LS-DYNA to the OSP.

Please contact the LSTC technical support team (support@lstc.com) for instructions to download and run LS-DYNA executables coupled with Madymo.

AIRBAG MODELING

Modeling of airbags is accomplished by use of shell or membrane elements in conjunction with a control volume (see *AIRBAG_*OPTION*) and possibly a single surface contact algorithm to eliminate interpenetrations during the inflation phase (see *CONTACT_*OPTION*). The contact types showing an “a” in front are most suited for airbag analysis. Current recommended material types for the airbags are:

1. *MAT_ELASTIC (Type 1, Elastic)

2. *MAT_COMPOSITE_DAMAGE (Type 22, layered orthotropic elastic for composites)

3. *MAT_FABRIC (Type 34, fabric model for folded airbags)

Model 34 is a “fabric” model which can be used for flat bags. As a user option this model may or may not support compression.

The elements which can be used are as follows:

1. Belytschko-Tsay quadrilateral with 1 point quadrature. This element behaves rather well for folded and unfolded cases with only a small tendency to hour-

glass. The element tends to be a little stiff. Stiffness form hourglass control is recommended.

2. Belytschko-Tsay membrane. This model is softer than the normal Belytschko-Tsay element and can hourglass quite badly. Stiffness form hourglass is recommended. As a better option, the fully integrated Belytschko-Tsay membrane element can be chosen.

3. C0 Triangular element. The C0 triangle is very good for flat bag inflation and has no tendency to hourglass.

4. The best choice is a specially developed airbag membrane element with quadrilateral shape. This is an automatic choice when the fabric material is used.

As an airbag inflates, a considerable amount of energy is transferred to the surrounding air. This energy transfer decreases the kinetic energy of the bag as it inflates. In the control volume logic, this is simulated either by using either a mass weighted damping option or a back pressure on the bag based on a stagnation pressure. In both cases, the energy that is absorbed is a function of the fabric velocity relative to a rigid body velocity for the bag. For the mass weighted case, the damping force on a node is proportional to the mass times the damping factor times the velocity vector. This is quite effective in maintaining a stable system, but has little physical justification. The latter approach using the stagnation pressure method estimates the pressure needed to accelerate the surrounding air to the speed of the fabric. The formula for this is:

$$P = \text{Area} \times \alpha \times \left[\left(\vec{V}_i - \vec{V}_{cg}\right) \cdot \hat{n}\right]^2$$

This formula accomplishes a similar function and has a physical justification. Values of the damping factor, α, are limited to the range of 0 to 1, but a value of 0.1 or less is more likely to be a good value.

COMMON ERRORS

1. **Improper airbag inflation or no inflation.**

   The most common problem is inconsistency in the units used for the input constants. An inflation load curve must also be specified. The normals for the airbag segments must all be consistent and facing outwards. If a negative volume results, this can sometimes be quickly cured by using the “flip” flag on the control volume definition to force inward facing normals to face outwards.

2. **Excessive airbag distortions.**

   Check the material constants. Triangular elements should have less distortion problems than quadrilaterals. Overlapped elements at time zero can cause locking to occur in the contact leading to excessive distortions. The considerable energy input to the bag will create numerical noise and some damping is recommended to avoid problems.

3. **The dummy passes through the airbag.**

   A most likely problem is that the contacts are improperly defined. Another possibility is that the models were developed in an incompatible unit system. The extra check for penetration flag if set to 1 on the contact control cards variable PENCHK in the *CONTACT_... definitions may sometimes cause nodes to be prematurely released due to the softness of the penalties. In this case the flag should be turned off.

4. **The OSP fails to converge.**

   This may occur when excessively large forces are passed to the OSP. First, check that unit systems are consistent and then look for improperly defined contacts in the LS-DYNA input.

5. **Time step approaches zero.**

   This is almost always in the airbag. If elastic or orthotropic (*MAT_ELASTIC or *MAT_COMPOSITE material 1 or 22) is being used, then switch to fabric material *MAT_FABRIC which is less time step size sensitive and use the fully integrated membrane element. Increasing the damping in the control volume usually helps considerably. Also, check for “cuts” in the airbag where nodes are not merged. These can allow elements to deform freely and cut the time step to zero.

APPENDIX J: Interactive Graphics Commands

Only the first four or less characters of command are significant. These commands are available in the interactive phase of LS-DYNA. The interactive graphics are available by using the “SW5.” command after invoking the Ctrl-C interrupt. The MENU command brings up a push button menu. Only available in Unix and Linux.

ANIMATE Animate saved sequence, stop with switch 1.

BACK Return to previous display size after zoom, then list display attributes.

BGC Change display background color RGB proportions BGC <red> <green> <blue>.

BIP Select beam integration point for contour; BIP <#>.

CENTER Center model, center on node, or center with mouse, i.e., center cent <value> or cent gin.

CL Classification labels on display; class commercial_in_ confidence.

CMA Color materials on limited color displays.

COLOR Set or unset shaded coloring of materials.

CONTOUR View with colored contour lines; contour < component #> < list mat #>; see TAURUS manual.

COOR Get node information with mouse.

COP Hardcopy of display on the PC copy < laserj paintj tekcol coljet or epson>.

CR Restores cutting plane to default position.

CUT Cut away model outside of zoom window; use mouse to set zoom window size.

CX Rotate slice plane at zmin about x axis.

CY Rotate slice plane at zmin about y axis.

CZ Rotate slice plane at zmin about z axis.

DIF	Change diffused light level for material; DIF < mat #, -1 for all > <value>.
DISTANCE	Set distance of model from viewer; DIST < value in normalized model dimensions>.
DMATERIALS	Delete display of material in subsequent views; DMAT < ALL or list of numbers>.
DRAW	Display outside edges of model.
DSCALE	Scale current displacement from initial shape.
DYN	After using TAURUS command will reset display to read current DYNA3D state data.
ELPLT	Set or unset element numbering in subsequent views.
END	Delete display and return to execution.
ESCAPE	Escapes from menu pad mode.
EXECUTE	Return to execution and keep display active.
FCL	Fix or unfix current contour levels.
FOV	Set display field of view angle; FOV < value in degrees>.
FRINGE	View with colored contour fringes; fringe < component #> < list mat #>; see TAURUS manual.
GETFRAME	Display a saved frame; GETF < frame #>.
HARDWARE	Hardware mode; workstation hardware calls are used to draw, move and color model; repeat command to reset to normal mode.
HELP	
HZB	Switch on or off hardware zbuffer for a subsequent view, draw or contour command; rotations and translations will be in hardware.
LIMIT	Set range of node numbers subsequent views; limit < first node #> < last node #>.
MAT	Re-enable display of deleted materials mat < all or list of numbers>.

MENU Button menu pad mode.

MOTION Motion of model through mouse movement or use of a dial box. The left button down enables translation in the plane, middle button rotation about axes in the plane; and with right button down in the out of plane axis; left and middle button down quit this mode.

MOV Drag picked part to new position set with mouse.

NDPLT Set or unset node numbering in subsequent views.

NOFRAME Set and unset drawing of a frame around the picture.

PAUSE Animation display pause in seconds

PHS2 or THISTORY Time history plotting phase. Similar to LS-TAURUS.

PICK Get element information with mouse.

POST Enable or disable postscript mode on the PC and eps file is written as picture is drawn; remove eofs and init-graphics for eps use.

QUIT Same as execute.

RANGE Set fix range for contour levels; range <minvalue> <maxvalue>.

RAX Reflect model about xy plane; restore command will switch-off reflections.

RAY Reflect model about yz plane; restore command will switch-off reflections.

RAZ Reflect model about zx plane, restore command will switch-off reflections.

RESTORE Restores model to original position, also switches off element and node numbers, slice capper, reflections and cut model.

RETURN Exit.

RGB Change color red green blue element < mat #> <red> <green> <blue>.

RX Rotate model about x axis.

RY	Rotate model about y axis.
RZ	Rotate model about z axis.
SAVE	Set or unset saving of display for animation.
SEQUENCE	Periodic plot during execution; SEQ <# of cycles > <commands> EXE.
SHR	Shrink element facets towards centroids in subsequent views, shrink <value>.
SIP	Select shell integration point for contour; SIP <#>.
SLICE	Slice model a z-minimum plane; slice < value in normalized model dimension > this feature is removed after using restore. Slice enables internal details for brick elements to be used to generate new polygons on the slice plane.
SNORMAL	Set or unset display of shell direction normals to indicate topology order.
SPOT	Draw node numbers on model spot < first #> < last # for range>.
TAURUS	LS-DYNA database, TAU < state #>, or state < state #>, reads LS-TAURUS file to extract previous state data.
TRIAD	Set or unset display of axis triad.
TSHELL	Set or unset shell element thickness simulation in subsequent views.
TV	Change display type.
TX	Translates model along x axis.
TY	Translates model along y axis.
TZ	Translates model along z axis.
V	Display model using painters algorithm.
VECTOR v or d	View with vector arrows of velocity or displacement; <v> or <d>.
ZB	Switch on or off zbuffer algorithm for subsequent view; or draw commands.
ZIN	Zoom in using mouse to set display size and position.

ZMA Set position of zmax plane; ZMAX < value in normalized model dimensions>.

ZMI Set position of zmin plane; ZMIN < value in normalized model dimensions>.

ZOUT Zoom out using mouse to set displays size expansion and position.

APPENDIX K: Interactive Material Model Driver

INTRODUCTION

The interactive material model driver in LS-DYNA allows calculation of the material constitutive response to a specified strain path. Since the constitutive model subroutines in LS-DYNA are directly called by this driver, the behavior of the constitutive model is precisely that which can be expected in actual applications. In the current implementation the constitutive subroutines for both shell elements and solid elements can be examined.

INPUT DEFINITION

The material model driver is invoked when no *NODE or *ELEMENT commands are present in a standard LS-DYNA input file. The number of material model definitions should be set to one, the number of load curves should be nine, and the termination time to the desired length of the driver run. The complete state dump interval as given in *DATABASE_BINARY_D3PLOT serves as the time step to be used in the material model driver run. Plotting information is saved in core for the interactive plotting phase.

The input deck typically consists only of *KEYWORD, *DATABASE_BINARY_D3PLOT, *CONTROL_TERMINATION, one each of *PART/*MAT/*SECTION, and nine load curves (*DEFINE_CURVE) describing the strain path. These nine curves define the time history of the displacement gradient components shown in Table 54-1.

The velocity gradient matrix, L_{ij}, is approximated by taking the time derivative of the components in Table 54-1. If these components are considered to form a tensor S_{ij} , then

$$L_{ij}(t) = \frac{S_{ij}(t) - S_{ij}(t_{k-1})}{(t - t_k)}$$

and the strain rate tensor is defined as

$$d_{ij} = \frac{L_{ij} + L_{ij}^t}{2}$$

and the spin tensor as

$$\omega_{ij} = \frac{L_{ij} - L_{ij}^t}{2}$$

Load Curve Number	Component Definition
1	$\frac{\partial u}{\partial x}$
2	$\frac{\partial v}{\partial y}$
3	$\frac{\partial w}{\partial z}$
4	$\frac{\partial u}{\partial y}$
5	$\frac{\partial v}{\partial x}$
6	$\frac{\partial u}{\partial z}$
7	$\frac{\partial w}{\partial x}$
8	$\frac{\partial v}{\partial z}$
9	$\frac{\partial w}{\partial y}$

Table 54-1. Load Curve Definitions versus Time

INTERACTIVE DRIVER COMMANDS

After reading the input file and completing the calculations, LS-DYNA gives a command prompt to the terminal. A summary of the available interactive commands is given below. An on-line help package is available by typing HELP. Only available in Unix and Linux.

ACCL — Scale all abscissa data by f. Default is f = 1.

ASET amin omax — Set min and max values on abscissa to amin and amax, respectively. If amin = amax = 0, scaling is automatic.

CHGL n — Change label for component n. LS-DYNA prompts for new label.

CONTINUE Re-analyze material model.

CROSS c_1 c_2 Plot component c_1 versus c_2.

ECOMP Display component numbers on the graphics display:

EQ.1: x-stress,
EQ.2: y-stress,
EQ.3: z-stress,
EQ.4: xy-stress,
EQ.5: yz-stress,
EQ.6: zx-stress,
EQ.7: effective plastic strain,
EQ.8: pressure,
EQ.9: von Mises (effective) stress,
EQ.10: 1st principal deviatoric stress,
EQ.11: 2nd principal deviatoric stress,
EQ.12: 3rd principal deviatoric stress,
EQ.13: maximum shear stress,
EQ.14: 1st principal stress,
EQ.15: 2nd principal stress,
EQ.16: 3rd principal stress,
EQ.17: ln (v/v0),
EQ.18: relative volume,
EQ.19: v0/v - 1.0,
EQ.20: 1st history variable,
EQ.21: 2nd history variable.

Adding 100 or 400 to component numbers 1-16 yields strains and strain rates, respectively.

FILE name Change pampers filename to name for printing.

GRID Graphics displays will be overlaid by a grid of orthogonal lines.

NOGRID Graphics displays will not be overlaid by a grid of orthogonal lines.

OSCL Scale all ordinate data by f. Default is f = 1.

OSET omin omax Set min and max values on ordinate to omin and omax, respectively. If omin = omax = 0, scaling is automatic.

PRINT — Print plotted time history data into file “pampers.” Only data plotted after this command is printed. File name can be changed with the “file” command.

QUIT, END, T — Exit the material model driver program.

RDLC m n r_1 z_1 ... r_n z_n — Redefine load curve m using n coordinate pairs (r_1,z_1) (r_2,z_2),...(r_n,z_n).

TIME c — Plot component c versus time.

TV n — Use terminal output device type n. LS-DYNA provides a list of available devices.

Presently, the material model drive is implemented for solid and shell element material models. The driver does not yet support material models for beam elements.

Use the keyword *CONTROL_MPP_MATERIAL_MODEL_DRIVER and run the input deck only on one processor if a distributed memory executable (MPP) is used.

APPENDIX L: VDA Database

VDA surfaces describe the surface of geometric entities and are useful for the simulation of sheet forming problems. The German automobile and automotive supplier industry (VDA) has defined the VDA guidelines [VDA 1987] for a proper surface definition used for the exchange of surface data information. In LS-DYNA, this format can be read and used directly. Some files have to be provided for proper linkage to the motion of the correlation parts/materials in LS-DYNA.

Linking is performed via names. To these names surfaces are attached, which in turn can be linked together from many files externally to LS-DYNA. Thus, arbitrary surfaces can be provided by a preprocessor and then can be written to various files. The so-called VDA file given on the LS-DYNA execution line via V = **vda** contains references to all other files. It also contains several other parameters affecting the treatment in the contact subroutines; see below.

The procedure is as follows. If VDA surfaces are to be used, the file specified by **vda** must have the following form. The file is free formatted with blanks as delimiters. Note that the characters "}" and "{" must be separated from the other input by spaces or new lines. The **vda** file may contain any number of input file specifications of the form:

```
file afile bfile {
        alias definitions
}
alias definitions
followed by optional runtime parameters and a final end statement.
```

The file, **afile,** is optional, and if given must be the name of an ASCII input file formatted in accordance with the VDA Surface Interface Definitions as defined by the German automobile and automotive supply industry. **bfile** is required, and is the name of a binary VDA file. In a first run **afile** is given and **bfile** is created. In any further run, if the definitions have not changed, **afile** can be dropped and only **bfile** is needed. The purpose of **bfile** is that it allows for much faster initialization if the same VDA surfaces are to be used in a future LS-DYNA run.

If **afile** is given, **bfile** will always be created or overwritten. The alias definitions are used for linking to LS-DYNA and between the various surface definitions in the files defined by **afile** and **bfile.**

The alias definitions are of the form

```
        alias name { el1 el2 ... eln }
```

where **name** is any string of up to 12 characters, and el1,...,eln are the names of VDA elements as specified in **afile**. The list of elements can be empty, in which case all the SURF and FACE VDA elements in **afile** will be used. Care should be taken to ensure that the alias **name** is unique, not only among the other aliases, but among the VDA

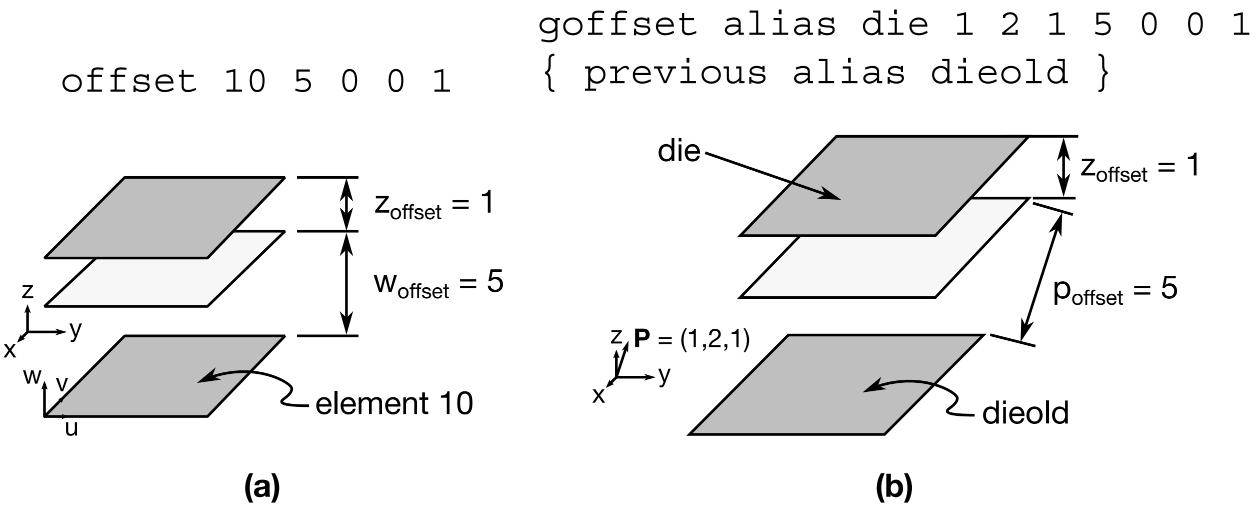


Figure 55-1. (a) a schematic illustration of offset version 1, and (b) is a schematic illustration of offset version 2.

element names in **afile**. This collection of VDA elements can later be indicated by the alias **name**. In particular, **name** may appear in later alias definitions.

Often it is required that a punch or die be created by a simple offset. This can be achieved in the **vda** files in two ways, either on VDA elements directly, or on parts defined by aliases. This feature offers great capability in generating and using surface data information.

Offset Version 1

As an option, the keyword **offset** may appear in the alias list which allows a new surface to be created as a normal offset (plus translation) of a VDA element in the file. The keyword **offset** my be applied to VDA elements only, not aliases. The usage of **offset** follows the form

offset elem **normal x y z**

where **normal** is the amount to offset the surface along the normal direction, and **x**,**y**,**z** are the translations to be applied. The default normal direction is given by the cross product of the local u and v directions on the VDA surface, taken in that order. **normal** can be negative.

Offset Version 2

Frequently, it is convenient to create a new alias **name** by offsetting and translating an existing **name**. The keyword **goffset** provides this function:

goffset **alias name xc yc zc normal x y z** { previous **alias name** } where **normal**, **x**, **y**, and **z** are defined as in the offset keyword. A reference point $\mathbf{x_c}$, $\mathbf{y_c}$, and $\mathbf{z_c}$ defines a point in space which determines the normal direction to the VDA surface, which is a vector from the origin to $P(x_c,y_c,z_c)$. See example below.

Finally, several parameters affecting the VDA surface iteration routines can be reset in the file **vda**. These parameters, and their default values in square brackets [ ], are:

gap [5.0] The maximum allowable surface gap to be filled in during the iterations. Points following the surface will effectively extend the edges of surfaces if necessary to keep them from falling through cracks in the surface smaller than this. This number should be set as small as possible while still allowing correct results. In particular, if your VDA surfaces are well formed (having no gaps), this parameter can be set to 0.0. The default value is 5.0.

t**rack** [2.0] A point must be within this distance of contact to be continually tracked. When a point not being tracked comes close to a surface, a global search is performed to find the near surface point. While a point is being tracked, iterations are performed every cycle. These iterations are much faster, but if the point is far away it is faster to occasionally do the global search. The default value is 2.0.

t**rack2** [5.0] Every VDA surface is surrounded by a bounding box. When a global search needs to be performed but the distance from a point to this box is > t**rack2**, the actual global search is not performed. This will require another global search to be performed sooner than if the actual distance to the surface were known, but also allows many global searches to be skipped. The default value is 5.0.

ntrack [4] The number of VDA surfaces for which each point maintains actual distance information. A global lower bound on distance is maintained for all remaining surfaces. Whenever the point moves far enough to violate this global lower bound, all VDA surfaces must have the global search performed for them. Hence, this parameter should be set to the maximum number of surfaces that any point can be expected to be near at one time (the largest number of surfaces that come together at one point). Setting **ntrack** higher will require more memory but result in faster execution. If **ntrack** is too low, performance may be unacceptably slow. The default value is 4.0.

toroid [.01] Any surface with opposing edges which are within distance [t] of each other is assumed to be cylindrical. Contacts occurring on one edge can pass to the adjacent edge. The default value is 0.01.

converge [.01] When surface iterations are performed to locate the near point, iteration is continued until convergence is detected to within this distance (all VDA coordinates are in mm). The default value is 0.01.

iterate [8] Maximum number of surface iterations allowed. Since points being tracked are checked every cycle, if convergence fails it will be tried again next cycle, so setting this parameter high does not necessarily help much. On the other hand, a point converging to a crease in the VDA surface (a crease between patches with discontinuous derivative, for example) may bounce back and forth between patches up to this many times, without actually moving. Hence, this value should not be too large. The default value is 8.

el_size [t mx mn]

Controls the generation of elements where:

t = surface tolerance for mesh generation,
mx = maximum element size to generate,
mn = minimum element size to generate.

The default values are [0.25 100. 1.0]

aspect [s1 s2] Controls the generation of elements where:

s1 = maximum difference in aspect ratio between elements generated in neighboring VDA patches,
s2 = maximum aspect ratio for any generated element.

The default values are [1.5 4.0]

cp_space [10] Determines the spacing around the boundaries of parts at which the size of elements is controlled. In the interior of the part, the element size is a weighted function of these control points as well as additional control points in the interior of the region. If there are too few control points around the boundary, elements generated along or near straight boundaries, but between control points, may be too small. The default value is 10.

meshonly The existence of this keyword causes LS-DYNA to generate a file containing the mesh for the VDA surfaces and then terminate.

onepatch The existence of this keyword causes LS-DYNA to generate a single element on each VDA patch.

somepatch [n] Like onepatch, but generates an element for 1 out of every [n] patches.

Example for file V = **vda**. It contains the following data:

```
file vda1 vda1.bin {
        alias die {
                sur0001
                sur0003
                offset fce0006 1.5 0 0 120
        }
        alias holder1 { sur008 }
}
file vda2 vda2.bin {
        alias holder2 { sur003 }
}
alias holder { holder1 holder2 }
ntrack 6
gap 0.5

end
```

Explanation:

vda1 This file contains the surfaces/face elements sur0001,sur0003, fce0006, and sur0008.

alias die face Combines the surface/face elements sur0001, sur0003, and the offsetted fce0006 to a global surface.

alias holder1 Defines the surface/face element sur0008 as holder1.

vda2 This file contains the surface/face element sur0003.

alias holder2 Defines the surface/face element sur0003 as holder2.

alias holder Combines the surfaces holder1 and holder2 into a combined surface holder.

ntrack 6 For each point the actual distances to 6 VDA surfaces are maintained.

gap 0.5 Surface gaps of 0.5mm or less are filled.

end Closes reading of this file.

APPENDIX M: Commands for Two-Dimensional Rezoning

The rezoner in LS-DYNA contains many commands that can be broken down into the following categories:

- general,
- termination of interactive rezoning,
- redefinition of output intervals for data,
- graphics window controls,
- graphics window controls for x versus y plots,
- mesh display options,
- mesh modifications,
- boundary modifications,
- MAZE line definitions,
- calculation graphics display control parameters,
- calculation graphics display,
- cursor commands.

The use of the rezoner is quite simple. Commands for rezoning material number n can be invoked after the material is specified by the "M n" command. To view material n, the command "V" is available. The interior mesh can be smoothed with the "S" command and the boundary nodes can be adjusted after the "B" command is used to display the part side and boundary node numbers. Commands that are available for adjusting boundary nodes following the "B" command include:

ER, EZ, ES, VS, BD, ERS, EZS, ESS, VSS, BDS, SLN, SLNS

Rezoning is performed material by material. An example is shown.

Do not include the graphics display type number (see the "TV" command below) when setting up a command file for periodic noninteractive rezoning. No plotting is done when the rezoner is used in this mode.

REZONING COMMANDS BY FUNCTION:

Interactive Real Time Graphics

SEQ n commands EXE	Every n time steps execute the graphics commands which follow. For example the line seq 100 g exe would

cause the grid to be updated on the graphics display device every 100 cycles. The real time graphics can be terminated by using ctrl-c and typing “sw7.”

General

C
Comment - proceed to next line.

FRAME
Frame plots with a reference grid (default).

HELP
Enter HELP package and display all available commands. Description of each command is available in the HELP package.

HELP/commandname
Do not enter HELP package but print out the description on the terminal of the command following the slash.

LOGO
Put LLNL logo on all plots (default). Retyping this command removes the logo.

NOFRAME
Do not plot a reference grid.

PHP ans
Print help package - If answer equals ‘y’ the package is printed in the high speed printer file.

RESO n_x n_y
Set the x and y resolutions of plots to n_x and n_y, respectively. We default both n_x and n_y to 1024.

TV n
Use graphics output device type n. The types are installation dependent and a list will be provided after this command is invoked.

TR t
At time t, LS-DYNA will stop and enter interactive rezoning phase.

Termination of Interactive Rezoning

F
Terminate interactive phase, remap, continue in execution phase.

FR
Terminate interactive phase, remap, write restart dump, and call exit.

T or END
Terminate.

Redefinition of Output Intervals for Data

PLTI t
Reset the node and element data dump interval t.

PRTI t
Reset the node and element printout interval t.

TERM t
Reset the termination to t.

Graphics Window Controls

ESET n
Center picture at element n with a r by z window. This window is set until it is released by the unfix command or reset with another window.

FF
Encircle picture with reference grid with tickmarks. Default grid is plotted along bottom and left side of picture.

FIX
Set the display to its current window. This window is set until it is reset by the “GSET, “FSET”, or “SETF” commands or released by the “UNFIX” command.

FSET n r z
Center display at node n with a rectangular $\Delta r \times \Delta z$ window. This window is set until it is reset with or the “UNFIX” command is typed.

GSET r z l
Center display picture at point (r,z) with square window of width l. This window is set until it is reset or the “UNFIX” command is typed.

GRID
Overlay graphics displays with a grid of orthogonal lines.

NOGRID
Do not overlay graphics displays with a grid of orthogonal lines (default).

SETF r z r z
Center display at point (r,z) with a rectangular $\Delta r \times \Delta z$ window. This window is set until it is reset or the “UNFIX” command is typed.

UNFIX
Release current display window set by the “FIX”, “GSET”, “FSET” or “SETF” commands.

UZ a b l
Zoom in at point (a,b) with window l where a, b, and l are numbers between 0 and 1. The picture is assumed to lie in a unit square.

UZG

Cover currently displayed picture with a 10 by 10 square grid to aid in zooming with the unity zoom, "UZ", command.

UZOU a b l

Zoom out at point (a,b) with window l where a, b, and l are numbers between 0 and 1. The current window is scaled by the factor 1/Δ.The picture is assumed to lie in a unit square.

Z r z Δ

Zoom in at point (r,z) with window Δ.

ZOUT r z Δ

Zoom out at point (r,z) with window Δ. The window is enlarged by the ratio of the current window and Δ. The cursor may be used to zoom out via the cursor command DZOU and entering two points with the cursor to define the window. The ratio of the current window with the specified window determines the picture size reduction. An alternative cursor command, DZZO, may be used and only needs one point to be entered at the location where the reduction (2×) is expected.

Graphics Window Controls for x versus y plots

The following commands apply to line plots, interface plots, etc.

ASCL f_a

Scale all abscissa data by f_a. The default is $f_a = 1$.

ASET amin amax

Set minimum and maximum values on abscissa to amin and amax, respectively. If amin = amax = 0.0 (default) LS-DYNA determines the minimum and maximum values.

OSCL f_o

Scale all ordinate data by f_o. The default is $f_o = 1$.

OSET omin omax

Set minimum and maximum values on ordinate to omin and omax, respectively. If omin = omax = 0.0 (default) LS-DYNA determines the minimum and maximum values.

SMOOTH n

Smooth a data curve by replacing each data point by the average of the 2n adjacent points. The default is n = 0.

Mesh Display Options

ELPLT

Plot element numbers on mesh of material n.

FSOFF Turn off the "FSON" command.

FSON Plot only free surfaces and slideline interfaces with "O" command. (Must be used before "O" command.)

G View mesh.

GO View mesh right of centerline and outline left of centerline.

GS View mesh and solid fill elements to identify materials by color.

M n Material n is to be rezoned.

MNOFF Do not plot material numbers with the "O", "G", and "GO" commands (default).

MNON Plot material numbers with "O", "G", and "GO" commands.

NDPLT Plot node numbers on mesh of material n.

O Plot outlines of all material.

RPHA Reflect mesh, contour, fringe, etc., plots about horizontal axis. Retyping "RPHA" turns this option off.

RPVA Reflect mesh, contour, fringe, etc., plots about vertical axis. Retyping "RPVA" turns this option off.

TN r z l Type node numbers and coordinates of all nodes within window (r ± Δl/2, z ± Δl/2).

UG Display undeformed mesh.

V Display material n on graphics display. See command M.

VSF Display material n on graphics display and solid fill elements.

Mesh Modifications

BACKUP Restore mesh to its previous state. This command undoes the result of the last command.

BLEN s

Smooth option where s = 0 and s = 1 correspond to equipotential and isoparametric smoothing, respectively. By letting $0 \leq s \leq 1$ a combined blending is obtained.

CN m r z

Node m has new coordinate (r,z).

DEB n f_1 l_1 ... f_n l_n

Delete n element blocks consisting of element numbers f_1 to l_1, f_2 to l_2 ... , and f_o l_n inclusive. These elements will be inactive when the calculation resume.

DE e_1 e_2

Delete elements e_1 to e_2.

DMB n m_1 m_2 ... m_n

Delete n material blocks consisting of all elements with material numbers m_1, m_2,..., and m_n. These materials will be inactive when the calculations resume.

DM n m_1 m_2 ... m_n

Delete n materials including m_1, m_2,..., and m_n.

DZER k d incr nrow

Delete element row where k is the kept element, d is the deleted element, incr is the increment, and nrow is the number of elements in the row.

DZLN number n_1 n_2 n_3...n_{last}

Delete nodal row where number is the number of nodes in the row and n_1, n_2, ... n_{last} are the ordered list of deleted nodes.

DZNR l j incr

Delete nodal row where l is the first node in the row, j is the last node in the row, and incr is the increment.

R

Restore original mesh.

S

Smooth mesh of material n. To smooth a subset of elements, a window can be set via the "GSET", "FSET", OR "SETF" commands. Only the elements lying within the window are smoothed.

Boundary Modifications

A

Display all slidelines. Slave sides are plotted as dashed lines.

B

Determine boundary nodes and sides of material n and display boundary with nodes and side numbers.

BD m n

Dekink boundary from boundary node m to boundary node n (counterclockwise).

BDS s	Dekink side s.
DSL n l_1 l_2...l_n	Delete n slidelines including slideline numbers l_1 l_2..., and l_n.
ER m n	Equal space in r-direction boundary nodes m to n (counterclockwise).
ERS s	Equal space in the r-direction boundary nodes on side s.
ES m n	Equal space along boundary, boundary nodes m to n (counterclockwise).
ESS s	Equal space along boundary, boundary nodes on side s.
EZ m n	Equal space in z-direction boundary nodes m to n (counterclockwise).
EZS s	Equal space in the z-direction boundary nodes on side s.
MC n	Check master nodes of slideline n and put any nodes that have penetrated through the slave surface back on the slave surface.
MD n	Dekink master side of slideline n. After using this command, the SC or MC command is sometimes advisable.
MN n	Display slideline n with master node numbers.
SC n	Check slave nodes of slideline n and put any nodes that have penetrated through the master surface back on the master surface.
SD n	Dekink slave side of slideline n; after using this command, the SC or MC command is sometimes advisable.
SLN m n	Equal space boundary nodes between nodes m to n on a straight line connecting node m to n.
SLNS n	Equal space boundary nodes along side n on a straight line connecting the corner nodes.
SN n	Display slideline n with slave node numbers.

VS m n r

Vary the spacing of boundary nodes m to n such that r is the ratio of the first segment length to the last segment length.

VSS s r

Vary the spacing of boundary nodes on side s such that r is the ratio of the first segment length to the last segment length.

MAZE Line Definitions

B

Determine boundary nodes and sides of material n and display boundary with nodes and side numbers. See command "M".

LD n k l

Line definition n for MAZE includes boundary nodes k to l

LDS n l

Line definition n for MAZE consists of side number l.

M n

Material n is active for the boundary command B.

Calculation Graphics Display Control Parameters

MOLP

Overlay the mesh on the contour, fringe, principal stress, and principal strain plots. Retyping "MOLP" turns this option off.

NLOC

Do not plot letters on contour lines.

NUMCON n

Plot n contour levels. The default is 9.

PLOC

Plot letters on contour lines to identify their levels (default).

RANGE r_1 r_2

Set the range of levels to be between r_1 and r_2 instead of in the range chosen automatically by LS-DYNA. To deactivate this command, type RANGE 0 0.

Calculation Graphics Display

CONTOUR c n m_1 m_2...m_n

Contour component number c on n materials including materials m_1, m_2, ..., m_n. If n is zero, only the outline of material m_1 with contours is plotted. Component numbers are given in Table 56-1.

FRINGE c n m_1 m_2...m_n

Fringe component number c on n materials including m_1, m_2,...,m_n. If n is zero, only the outline of material m_1 with contours is plotted. Component numbers are given in Table 56-1.

IFD n

Begin definition of interface n. If interface n has been previously defined, this command has the effect of destroying the old definition.

IFN l m

Include boundary nodes l to m (counterclockwise) in the interface definition. This command must follow the "B" command.

IFP c m

Plot component c of interface m. Component numbers are given in Table 56-2.

IFS m

Include side m in the interface definition. Side m is defined for material n by the "B" command.

IFVA r_c z_c

Plot the angular location of the interface based on the center point (r_c,z_c) along the abscissa. Positive angles are measured counterclockwise from the y-axis.

IFVS

Plot the distance along the interface from the first interface node along the abscissa (default).

LINE c n m_1 m_2...m_n

Plot variation of component c along line defined with the "NLDF", "PLDF", "NSDF", or the "NSSDF" commands given below. In determining variation, consider n materials including material number m_1, m_2,...m_n.

NCOL n

Number of colors in fringe plots is n. The default value for n is 6 which includes colors magenta, blue, cyan, green, yellow, and red. An alternative value for n is 5 which eliminates the minimum value magenta.

NLDF n n_1 n_2...n_3

Define line for "LINE" command using n nodes including node numbers n_1, n_2,...n_n. This line moves with the nodes.

NSDF m

Define line for "LINE" command as side m. Side m is defined for material n by the "B" command.

NSSDF l m

Define line for "LINE" command and that includes boundary nodes l to m (counterclockwise) in the interface definitions. This command must follow the “B” command.

PLDF n r_1 z_1...r_n z_n

Define line for “LINE” command using n coordinate pairs (r_1,z_1), (r_2,z_2), ...(r_n,z_n). This line is fixed in space.

PRIN c n m_1 m_2...m_n

Plot lines of principal stress and strain in the yz plane on n materials including materials m_1, m_2,...,m_n. If n is zero, only the outline of material m_1 is plotted. The lines are plotted in the principal stress and strain directions. Permissible component numbers in Table M.1 include 0, 5, 6, 100, 105, 106,...,etc. Orthogonal lines of both maximum and minimum stress are plotted if components 0, 100, 200, etc. are specified.

PROFILE c n m_1 m_2...m_n

Plot component c versus element number for n materials including materials m_1, m_2,...,m_n. If n is Ø then component c is plotted for all elements. Component numbers are given in Table M.1.

VECTOR c n m_1 m_2...m_n

Make a vector plot of component c on n materials including materials m_1, m_2,...,m_n. If n is zero, only the outline of material m_1 with vectors is plotted. Component c may be set to “D” and “V” for vector plots of displacement and velocity, respectively.

No.	Component	No.	Component
1	y	21*	ln (V/Vo) (volumetric strain)
2	z	22*	y-displacement
3	hoop	23*	z-displacement
4	yz	24*	maximum displacement
5	maximum principal	25*	y-velocity, y-heat flux
6	minimum principal	26*	z-velocity, y-heat flux
7	von Mises (Appendix A)	27*	maximum velocity, maximum heat flux
8	pressure or average strain	28	ij normal
9	maximum principal - minimum principal	29	jk normal
10	y minus hoop	30	kl normal
11	maximum shear	31	li normal
12	ij and kl normal (see appendix D)	32	ij shear
13	jk and li normal	33	jk shear
14	ij and kl shear	34	kl shear
15	jk and li shear	35	li shear
16	y-deviatoric	36*	relative volume V/Vo
17	z-deviatoric	37*	Vo/V-1
18	hoop-deviatoric	38*	bulk viscosity, Q
19*	effective plastic strain	39*	P + Q
20*	temperature/internal energy density	40*	density
41*-70*	element history variables		
71*	r-peak acceleration	76*	peak value of min in plane prin. stress
72*	z-peak acceleration	77*	peak value of maximum hoop stress
73*	r-peak velocity	78*	peak value of minimum hoopstress
74*	z-peak velocity	79*	peak value of pressure
75*	peak value of max. in plane prin. stress		

Table 56-1. Component numbers for element variables. By adding 100, 200 300, 400, 500 and 600 to the component numbers not followed by an asterisk, component numbers for infinitesimal strains, lagrange strains, almansi strains, strain rates, extensions, and residual strain are obtained. Maximum and minimum principal stresses and strains are in the rz plane. The corresponding hoop quantities must be examined to determine the overall extremum. ij, jk, etc. normal components are normal to the ij, jk, etc side. The peak value database must be flagged on Control Card 4 in columns 6-10 or components 71-79 will not be available for plotting.

No.	Component
1	pressure
2	shear stress
3	normal force
4	tangential force
5	y-force
6	z-force

Table 56-2. Component numbers for interface variables. In axisymmetric geometries the force is per radian.

Cursor Commands

DBD a b — Use cursor to define points a and b on boundary. Dekink boundary starting at a, moving counterclockwise, and ending at b.

DCN a b — Use cursor to define points a and b. The node closest to point a will be moved to point b.

DCSN n a — Move nodal point n to point a defined by the cursor.

DCNM a b — Use cursor to define points a and b. The node at point a is given the coordinate at point b.

DER a b — Use cursor to define points a and b on boundary. Equal space nodes in r-direction along boundary starting at a, moving counterclockwise, and ending at b.

DES a b — Use cursor to define points a and b on boundary. Equal space nodes along boundary starting at a, moving counterclockwise, and ending at b.

DEZ a b — Use cursor to define points a and b on boundary. Equal space nodes in z-direction along boundary starting at a, moving counterclockwise, and ending at b.

DTE a b — Use cursor to define points a and b on the diagonal of a window. The element numbers and coordinates of elements lying within the window are typed on the terminal.

DTN a b — Use cursor to define points a and b on the diagonal of a window. The node numbers and coordinates of nodal points lying within the window are typed on the terminal.

DTNC a
Use cursor to define point a. The nodal point number and nodal coordinates of the node lying closest to point a will be printed.

DVS a b r
Use cursor to define points a and b on boundary. Variable space nodes along boundary starting at a, moving counterwise, and ending at b. The ratio of the first segment length to the last segment length is give by r (via terminal).

DZ a b
Use cursor to define points a and b on the diagonal of a window for zooming.

DZOUT a b
Enter two points with the cursor to define the window. The ratio of the current window with the specified window determines the picture size reduction.

DZZ a
Use cursor to define point a and zoom in at this point. The new window is .15 as large as the previous window. The zoom factor can be reset by the crzf command for the .15 default.

DZZO a
Zoom out at point a by enlarging the picture two times.

APPENDIX N:
Rigid-Body Dummies

The two varieties of rigid body dummies available in LS-DYNA are described in this appendix. These are generated internally by including the appropriate *COMPONENT keyword. A description of the GEBOD dummies begins on this page and the HYBRID III family on page N.7.

GEBOD Dummies

Rigid body dummies can be generated and simulated within LS-DYNA using the keyword *COMPONENT_GEBOD. Physical properties of these dummies draw upon the GEBOD database [Cheng et al. 1994] which represents an extensive measurement program conducted by Wright-Patterson AFB and other agencies. The differential equations governing motion of the dummy are integrated within LS-DYNA separate from the finite element model. Interaction between the dummy and finite element structure is achieved using contact interfaces (see *CONTACT_GEBOD).

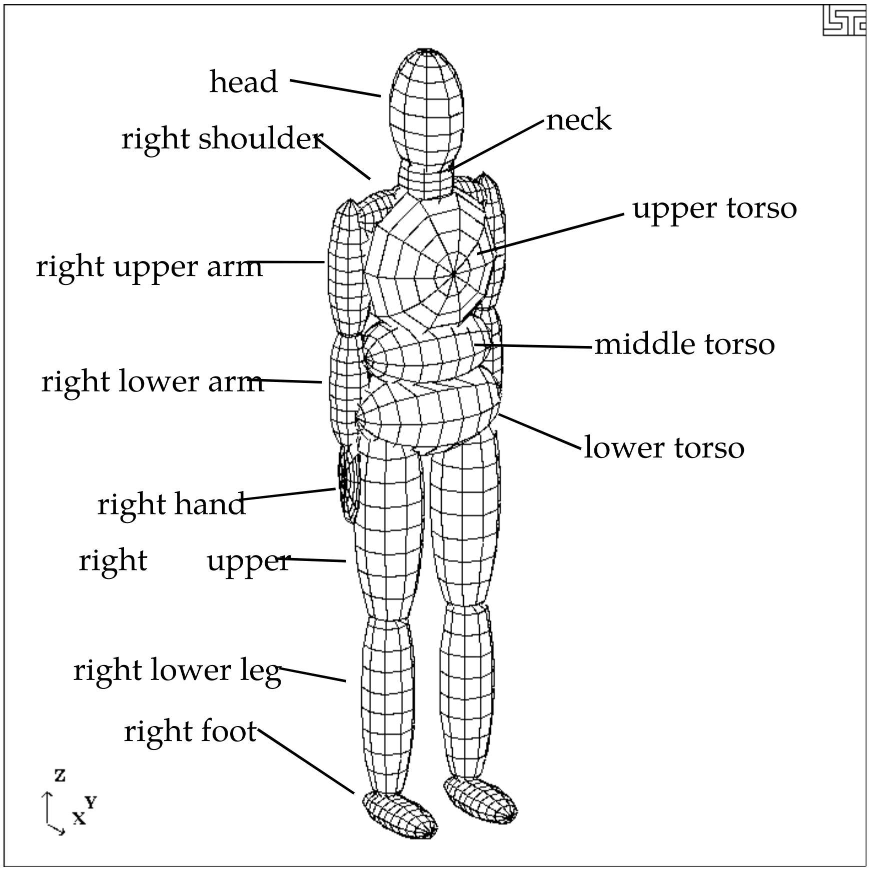


Figure 57-1. 50th percentile male dummy in the nominal position.

The dynamical system representing a dummy is comprised of fifteen rigid bodies (segments) and include: lower torso, middle torso, upper torso, neck, head, upper arms, forearms/hands, upper legs, lower legs, and feet. Ellipsoids are used for visualization and contact purposes. Shown in Figure 57-1 is a 50th percentile male dummy generated using the keyword command *COMPONENT_GEBOD_MALE. Note that the ellipsoids representing the shoulders are considered to be part of the upper torso segment and the hands are rigidly attached to the forearms.

Each of the rigid segments which make up the dummy is connected to its neighbor with a joint which permits various relative motions of the segments. Listed in the Table 57-2 are the joints and their applicable degrees of freedom.

Joint Name	Degree(s) of Freedom		
	1^{st}	2^{nd}	3^{rd}
pelvis	lateral flexion (x)	forward flexion (y)	torsion (z)
waist	lateral flexion (x)	forward flexion (y)	torsion (z)
lower neck	lateral flexion (x)	forward flexion (y)	torsion (z)
upper neck	lateral flexion (x)	forward flexion (y)	torsion (z)
shoulders	abduction-adduction (x)	internal-external rotation (z)	flexion-extension (y)
elbows	flexion-extension (y)	n/a	n/a
hips	abduction-adduction (x)	medial-lateral rotation (z)	flexion-extension (y)
knees	flexion-extension (y)	n/a	n/a
ankles	inversion-eversion (x)	dorsi-plantar flexion (y)	medial-lateral rotation (z)

Table 57-2. Joints and associated degrees of freedom. Local axes are in parentheses.

Orientation of a segment is effected by performing successive right-handed rotations of that segment relative to its parent segment - each rotation corresponds to a joint degree of freedom. These rotations are performed about the local segment axes and the sequence is given in Table 57-2. For example, the left upper leg is connected to the lower torso by the left hip joint; the limb is first abducted relative to lower torso, it then undergoes lateral rotation, followed by extension. The remainder of the lower extremity (lower leg and foot) moves with the upper leg during this orientation process.

By default all joints are assigned stiffnesses, viscous characteristics, and stop angles which should give reasonable results in a crash simulation. One or all default values of a joint may be altered by applying the *COMPONENT_GEBOD_JOINT_*OPTION* command to the joint of interest. The default shape of the resistive torque load curve used by all joints is shown in Figure 57-6. A scale factor is applied to this curve to obtain the proper stiffness relationship. Listed in Table 57-3 are the default values of joint characteristics for dummies of all types and sizes. These values are given in the

English system of units; the appropriate units are used if a different system is specified in card 1 of *COMPONENT_GEBOD_*OPTION*.

joint degrees of freedom	load curve scale factor (in·lbf)	damping coef. (in·lbf·s/rad)	low stop angle (degrees)	high stop angle (degrees)	neutral angle (degrees)
pelvis - 1	65000	5.77	-20	20	0
pelvis - 2	65000	5.77	-20	20	0
pelvis - 3	65000	5.77	-5	5	0
waist - 1	65000	5.77	-20	20	0
waist - 2	65000	5.77	-20	20	0
waist - 3	65000	5.77	-35	35	0
lower neck - 1	10000	5.77	-25	25	0
lower neck - 2	10000	5.77	-25	25	0
lower neck - 3	10000	5.77	-35	35	0
upper neck - 1	10000	5.77	-25	25	0
upper neck - 2	10000	5.77	-25	25	0
upper neck - 3	10000	5.77	-35	35	0
l. shoulder - 1	100	5.77	-30	175	0
r. shoulder - 1	100	5.77	-175	30	0
shoulder - 2	100	5.77	-65	65	0
shoulder - 3	100	5.77	-175	60	0
elbow - 1	100	5.77	1	-140	0
l. hip - 1	10000	5.77	-25	70	0
r. hip - 1	10000	5.77	-70	25	0
hip - 2	10000	5.77	-70	70	0
hip - 3	10000	5.77	-140	40	0
knee - 1	100	5.77	-1	120	0
l. ankle - 1	100	5.77	-30	20	0
l. ankle - 1	100	5.77	-20	30	0
ankle - 2	100	5.77	-20	45	0
ankle - 3	100	5.77	-30	30	0

Table 57-3. Default joint characteristics for all dummies.

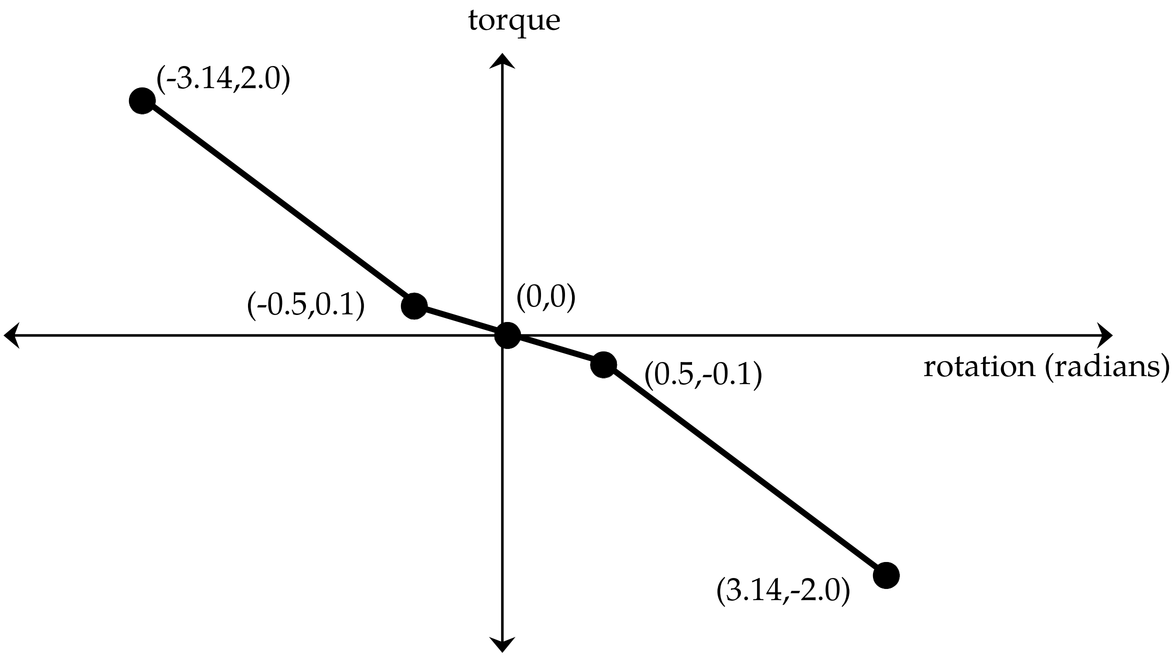


Figure 57-4 Characteristic torque curve shape used by all joints.

The dummy depicted in Figure 57-1 appears in what is referred to as its "nominal" position. In this position the dummy is standing upright facing in the positive x direction and the toe-to-head direction points in positive z. Additionally, the dummy's hands are at the sides with palms facing inward and the centroid of the lower torso is positioned at the origin of the global coordinate system. Each of the dummy's segments has a local coordinate system attached to it and in the nominal position all of the local axes are aligned with the global axes.

When performing a simulation involving a *COMPONENT_GEBOD dummy, a positioning file named "gebod.*did*" must reside in the directory with the LS-DYNA input file; here the extension *did* is the dummy ID number, see card 1 of *COMPONENT_GEBOD_*OPTION*. The contents of a typical positioning file is shown in Table 57-5; it consists of 40 lines formatted as (59a1,e30.0). All of the angular measures are input as degrees, while the lower torso global positions depend on the choice of units in card 1 of *COMPONENT_GEBOD_*OPTION*. Setting all of the values in this file to zero yields the so-called "nominal" position.

Table 57-5. Typical contents of a dummy positioning file.

lower torso	centroid global x position	0.0
lower torso	centroid global y position	0.0
lower torso	centroid global z position	0.0
total body	global x rotation	0.0
total body	global y rotation	-20.0
total body	global z rotation	180.0

pelvis	lateral flexion	+ = tilt right	0.0
pelvis	forward flexion	+ = lean fwd	0.0
pelvis	torsion	+ = twist left	0.0
waist	lateral flexion	+ = tilt right	0.0
waist	forward flexion	+ = lean fwd	0.0
waist	torsion	+ = twist left	0.0
lower neck	lateral flexion	+ = tilt right	0.0
lower neck	forward flexion	+ = nod fwd	0.0
lower neck	torsion	+ = twist left	0.0
upper neck	lateral lexion	+ = tilt right	0.0
upper neck	forward flexion	+ = nod fwd	0.0
upper neck	torsion	+ = twist left	0.0
left shoulder	abduction-adduction	+ = abduction	30.0
left shoulder	internal-external rotation	+ = external	-10.0
left shoulder	flexion-extension	- = fwd raise	-40.0
right shoulder	abduction-adduction	- = abduction	-30.0
right shoulder	internal-external rotation	- = external	10.0
right shoulder	flexion-extension	- = fwd raise	-40.0
left elbow	flexion-extension	+ = extension	-60.0
right elbow	flexion-extension	+ = extension	-60.0
left hip	abduction-adduction	+ = abduction	0.0
left hip	medial-lateral rotation	+ = lateral	0.0
left hip	flexion-extension	+ = extension	-80.0
right hip	abduction-adduction	- = abduction	0.0
right hip	medial-lateral rotation	- = lateral	0.0
right hip	flexion-extension	+ = extension	-80.0
left knee	flexion-extension	+ = flexion	50.0
right knee	flexion-extension	+ = flexion	50.0
left ankle	inversion-eversion	+ = eversion	0.0
left ankle	dorsi-plantar flexion	+ = plantar	0.0
left ankle	medial-lateral rotation	+ = lateral	0.0
right ankle	inversion-eversion	- = eversion	0.0
right ankle	dorsi-plantar flexion	+ = plantar	0.0
right ankle	medial-lateral rotation	- = lateral	0.0

In Figure 57-6 the 50th percentile male dummy is shown in a seated position and some of its joints are labeled. The file listed in Table 57-5 was used to put the dummy into the position shown. Note that the dummy was first brought into general orientation by setting nonzero values for two of the lower torso local rotations. This is accomplished by performing right-handed rotations successively about local axes fixed in the lower torso, the sequence of which follows: the first about local x, next about local y, and the last about local z. The dummy in Figure 57-6 was made to pitch backward by setting "total body global y rotation" equal to -20. Setting the "total body global z rotation" equal to 180 caused the dummy to rotate about the global z-axis and face in the -x direction.

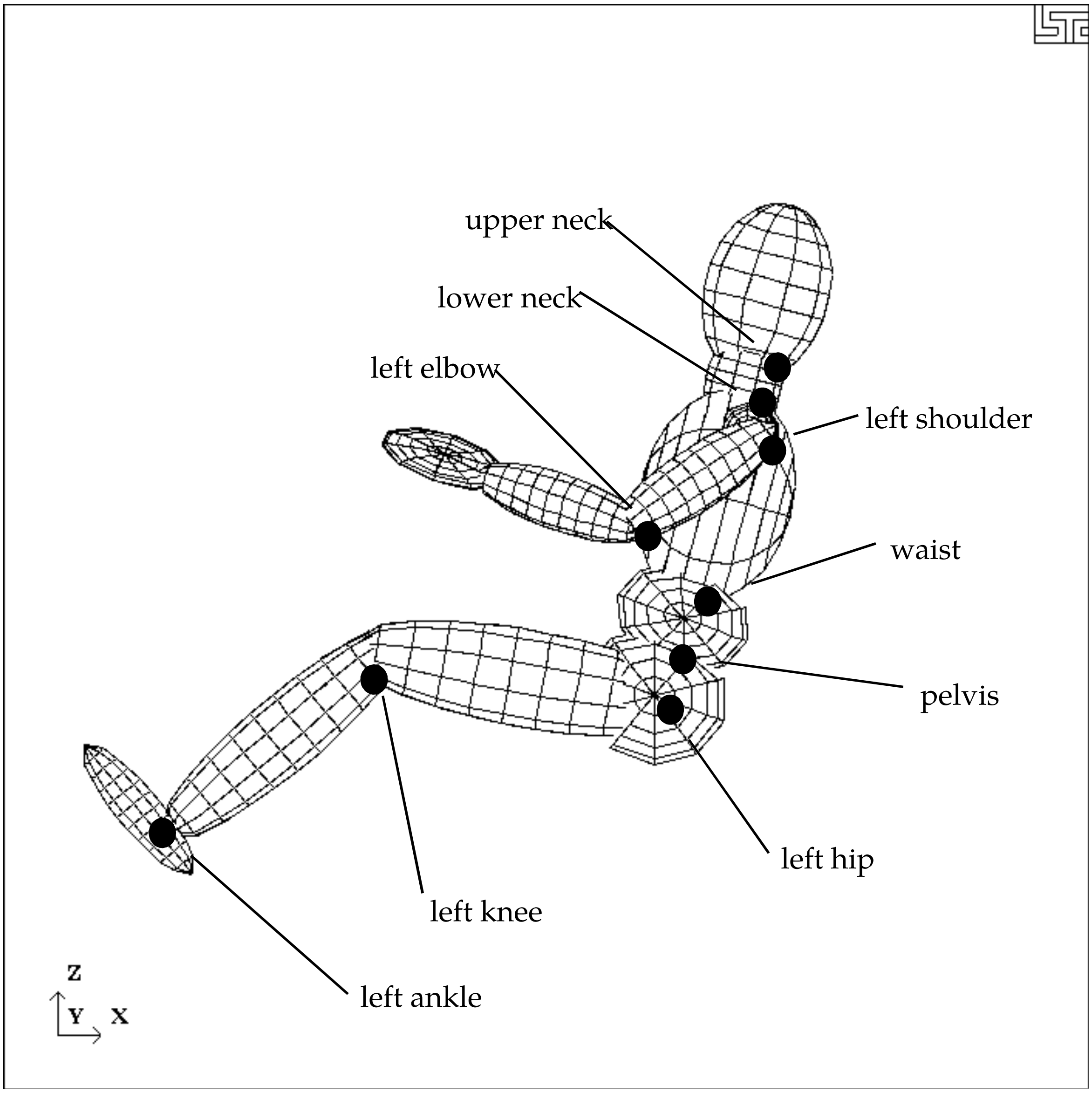


Figure 57-6. Dummy seated using the file listed in Table 57-5.

HYBRID III Dummies

A listing of applicable joint degrees of freedom of the Hybrid III dummy is given below.

Joint Name	Degree(s) of Freedom		
	1st	2nd	3rd
lumbar	flexion (y)	torsion (z)	
lower neck	flexion (y)	torsion (z)	
upper neck	flexion (y)	torsion (z)	
shoulders	flexion-extension (y)	abduction-adduction (x)	n/a
elbows	flexion-extension (y)	n/a	n/a
wrists	flexion-extension (x)	n/a	n/a
hips	abduction-adduction (x)	medial-lateral rotation (z)	flexion-extension (y)
knees	flexion-extension (y)	n/a	n/a
ankles	inversion-eversion (x)	medial-lateral rotation (z)	dorsi-plantar flexion (y)
sternum	translation (x)	rotation (y)	rotation (z)
knee sliders	translation (z)		

Table 57-7. Joints and associated degrees of freedom. Local axes are in parentheses.

Joint springs of the *COMPONENT_HYBRIDIII dummies are formulated in the following manner.

$$\begin{aligned} T &= a_{\mathrm{lo}}(q - q_{\mathrm{lo}}) + b_{\mathrm{lo}}(q - q_{\mathrm{lo}})^3, && q \le q_{\mathrm{lo}} \\ T &= a_{\mathrm{hi}}(q - q_{\mathrm{hi}}) + b_{\mathrm{hi}}(q - q_{\mathrm{hi}})^3, && q \ge q_{\mathrm{hi}} \\ T &= 0, && q_{\mathrm{lo}} < q < q_{\mathrm{hi}} \end{aligned}$$

Where,

T is the joint torque
q is the joint generalized coordinate
a_{lo} and b_{lo} are the linear and cubic coefficients, respectively, for the low regime
a_{hi} and b_{hi} are the linear and cubic coefficients, respectively, for the high regime
q_{lo} and q_{hi} are the activation values for the low and high regimes, respectively

APPENDIX O: LS-DYNA MPP User Guide

This is a short user's guide for the MPP version of LS-DYNA. For a general guide to the capabilities of LS-DYNA and a detailed description of the input, consult the LS-DYNA User's Manual. If you have questions about this guide, find errors or omissions in it, please email manual@lstc.com.

SUPPORTED FEATURES

The only input formats currently supported are 920 and later, including keyword input. Models in any of the older formats will need to be converted to one of these input formats before they can be run with the current version of LS-DYNA for massively parallel processors, mpp.

The large majority of LS-DYNA options are available on MPP computers. Those that are not supported are being systematically added. Unless otherwise noted here, all the options of LS-DYNA version 93x are supported by MPP/LS-DYNA.

Here is the list of **unsupported** features:

*BOUNDARY_THERMAL_WELD

*BOUNDARY_USA_SURFACE

*CONTACT_1D

*DATABASE_AVS

*DATABASE_MOVIE

*DATABASE_MPGS

*LOAD_SUPERPLASTIC_OPTION

*USER

*TERMINATION_NODE

CONTACT INTERFACES

MPP/LS-DYNA uses a completely redesigned, highly parallel contact algorithm. The contact options currently **unsupported** include:

*CONTACT_FORCE_TRANSDUCER_CONSTRAINT

Because these options are all supported via the new, parallel contact algorithms, slight differences in results may be observed as compared to the serial and SMP versions of LS-DYNA. Work has been done to minimize these differences, but they may still be evident in some models.

For each of the supported CONTACT_control cards, there is an optional string_MPP which can be appended to the end. Adding these characters triggers the reading of a new control card immediately following (but after the TITLE card, if any). See the section on *CONTACT for details of the parameters and their meanings.

OUTPUT FILES AND POST-PROCESSING

For performance reasons, many of the ASCII output files normally created by LS-DYNA have been combined into a new binary format used by MPP/LS-DYNA. There is a post-processing program l2a, which reads this binary database of files and produces as output the corresponding ASCII files. The new binary files will be created in the directory specified as the global directory in the pfile (See section pfile). The files (up to one per processor) are named binout*nnnn*, where *nnnn* is replaced by the four-digit processor number. To convert these files to ASCII simply feed them to the l2a program like this:

```
l2a binout*
```

LS-PREPOST is able to read the binout files directly, so conversion is not required, it is provided for backward compatibility.

The **supported** ASCII files are:

*DATABASE_SECFORC

*DATABASE_RWFORC

*DATABASE_NODOUT

*DATABASE_NODOUTHF

*DATABASE_ELOUT

*DATABASE_GLSTAT

*DATABASE_DEFORC

*DATABASE_MATSUM

*DATABASE_NCFORC

*DATABASE_RCFORC

*DATABASE_SPCFORC

*DATABASE_SWFORC

*DATABASE_DEFGEO

*DATABASE_ABSTAT

*DATABASE_NODOFR

*DATABASE_BNDOUT

*DATABASE_GCEOUT

*DATABASE_RBDOUT

*DATABASE_SLEOUT

*DATABASE_JNTFORC

*DATABASE_SBTOUT

*DATABASE_SPHOUT

*DATABASE_TPRINT

Some of the normal LS-DYNA files will have corresponding collections of files produced by MPP/LS-DYNA, with one per processor. These include the d3dump files (new names = d3dump.*nnnn*), the messag files (now mes*nnnn*) and others. Most of these will be found in the local directory specified in the pfile.

The format of the d3plot file has not been changed. It will be created in the global directory, and can be directly handled with your current graphics post-processor.

PARALLEL SPECIFIC OPTIONS

There are a few new command line options that are specific to the MPP version of LS-DYNA.

In the serial and SMP versions of LS-DYNA, the amount of memory required to run the problem can be specified on the command line by adding "memory=*XXX*", where *XXX* is the number of words of memory to be allocated. For the MPP code, this will result in each processor allocating *XXX* words of memory. If pre-decomposition has not been performed, one processor must perform the decomposition of the problem. This can require substantially more memory than will be required once execution has started. For this reason, there is a second memory command line option, "memory2=*YYY*". If used together with adding "memory=*XXX*", the decomposing processor will allocate *XXX* words of memory, and all other processors will allocate *YYY* words of memory.

For example, in order to run a 250,000 element crash problem on 4 processors, you might need memory=80m and memory2=20m. To run the same problem on 16 processors, you still need memory=80m, but can set memory2=6m. The value for *memory2* drops nearly linearly with the number of processors used to run the program, which works well for shared-memory systems.

Execution of the implicit solver in MPP requires a balance of memory across all of the processes. The user should not use memory2= specification for runs involving the implicit solver. If the model decomposition cannot be performed for the given memory= specification, one can try a pre-decomposition but the user would be advised to use a compute cluster with more real memory. It is suggested that the memory= specification be such to use no more than 75% of the real memory available to that process. On a compute cluster with each compute node having 48 Gbytes of memory and using 8 MPI processes, there is only 6 Gbytes of real memory per process. Converting to 8 byte words and using only the suggested 75% would have memory=560M as the maximum specification.

The full deck restart capability is supported by the MPP version of LS-DYNA, but in a manner slightly different than the SMP code. Each time a restart dump file is written, a separate restart file is also written with the base name D3FULL. For example, when the third restart file d3dump03 is written (one for each processor, d3dump03.0000, d3dump03.0001, etc), there is also a single file written named d3full03. This file is required in order to do a full deck restart and the d3dump files are not used in this case by the MPP code. In order to perform a full deck restart with the MPP code, you first must prepare a full deck restart input file as for the serial/SMP version. Then, instead of giving the command line option r=d3dump03 you would use the special option n=d3full03. The presence of this command line option tells the MPP code that this is a restart, not a new problem, and that the file d3full03 contains the geometry and stress data carried over from the previous run.

PFILE

There is a new command line option: p = pfile. pfile contains MPP specific parameters that affect the execution of the program. The file is split into sections, with several options in each section. Currently, these sections: **directory**, **decomposition**, **contact**, and **general** are available. First, here is a sample pfile:

```
directory {
        global rundir
        local /tmp/rundir
}
contact {
        inititer 3
}
```

The file is case insensitive and free format input. The sections and options currently supported are:

directory: Holds directory specific options

transfer_files

If this option is given, then processor 0 will write all output and restart files to the **global** directory (see "global" below), and scratch files to the **local** directory. All other processors will write all data to their **local** directory. At normal termination, all restart and output files will be copied from the processor specific **local** directories to the **global** directory. Also, if this is a restart from a dump file, the dump files will be distributed to the processors from the **global** directory. With this option enabled, there is no need for the processors to have shared access to a single disk for output – all files will be transferred as needed to and from the **global** directory.
Default = disabled.

global ***path***

Path to a directory where program output should be written. If **transfer_files** is not given, this directory needs to be accessible to all processors – otherwise it is only accessed by processor 0. This directory will be created if necessary.
Default = current working directory

global_message_files

If this option appears, the message files are written in the global directory rather than the local directory
Default = disabled (message files go in the local directory)

local ***path***

Path to a processor specific local directory for scratch files. This directory will be created if necessary. This should be a local disk on each processor, for performance reasons.
Default = global path

rmlocal

If this option is given and **transfer_files** is active, the program attempts to clean up the **local** directories on each processor. In particular, it deletes files that are successfully transferred back to the **global** directory, and removes the **local** directory if it was created. It will not delete any files if there is a failure during file copying, nor will it delete directories it did not create.
Default = disabled

repository ***path***

Path to a safe directory accessible from processor 0. This directory will be created if necessary. This is intended to be used as a safekeeping/backup of files during execution and should only be used if **transfer_files** is also given. If this directory is specified then the following actions occur:

- At program start up, any required files (d3dump, binout, etc) that cannot be located in the **global** directory are looked for in the **repository** for copying to the **local** processor directories.
- Important output files (d3dump, runrsf, d3plot, binout and others) are synchronized to the repository regularly. That is, every time one of these files is updated on the node local or the global directories, a synchronized copy is updated in the repository.

The intention is that the repository be on a redundant disk, such as NAS, to allow restarting the problem if a hardware failure should occur on the machine running the problem. It must be noted that some performance penalty must be paid for the extra communication and I/O. Effort has been made to minimize this overhead, but this option is not recommended for general use.
Default = unspecified

decomposition: Holds decomposition specific options

file *filename*

The name of the file that holds the decomposition information. This file will be created in the current working directory if it does not exist. If this option is not specified, MPP/LS-DYNA will perform the decomposition.
Default = None

numproc *n*

The problem will be decomposed for n processors. If $n > 1$ and you are running on 1 processor, or if the number of processors you are running on does not evenly divide *n*, then execution terminates immediately after decomposition. Otherwise, the decomposition file is written and execution continues. For a decomposition only run, both **numproc** and **file** should be specified.
Default = the number of processors you are running on.

method *name*

Currently, there are two decomposition methods supported, namely *rcb* and *greedy*. Method *rcb* is Recursive Coordinate Bisection. Method *greedy* is a simple neighborhood expansion algorithm. The impact on overall runtime is problem dependent, but rcb generally gives the best performance.
Default = rcb

region rx ry rz sx sy sz c2r s2r 3vec mat

See the section below on Special Decompositions for details about these decomposition options.

show

If this option appears in the decomposition section, the d3plot file is doctored so that the decomposition can be viewed with the post processor. Displaying material 1 will show that portion of the problem assigned to processor 0, and so on. The problem will not actually be run, but the code will terminate once the initial d3plot state has been written.

rcblog *filename*

This option is ignored unless the decomposition method is RCB. A record is written to the indicated file recording the steps taken during decomposition. This is an ascii file giving each decomposition **region** (see the section on Special Decompositions) and the location of each subdivision for that **region**. Except for the addition of this decomposition information, the file is otherwise equivalent to the current pfile. Thus it can be used directly as the pfile for a subsequent problem, which will result in a decomposition as similar as possible between the two runs. For example, suppose a simulation is run twice, but the second time with a slightly different mesh. Because of the different meshes the problems will be distributed differently between the processors, resulting in slightly different answers due to roundoff errors. If an rcblog is used, then the resulting decompositions would be as similar as possible.

vspeed

If this option is specified a brief measurement is taken of the performance of each processor by timing a short floating point calculation. The resulting information is used during the decomposition to distribute the problem according to the relative speed of the processors. This might be of some use if the cluster has machines of significantly different speed.

automatic

If this option is given, an attempt is made to automatically determine a reasonable decomposition, primarily based on the initial velocity of nodes in the model. Use of the **show** option is recommended to verify a reasonable decomposition.

aledist

Distribute ALE elements to all processors.

dcmem *n*

It may be in some cases that the memory requirements during the first phase of decomposition are too high. If that is found to be the case (if you get out of memory errors during decomposition phase 1), then this may provide a work around. Specifying a value n here will cause some routines to process the model in blocks of n items, when normal processing would read the whole set (of nodes, elements, whatever) all at once. This will reduce memory requirements at the cost of greater communication overhead. Most users will not need this option. Values in the range of 10,000 to 50,000 would be reasonable.

dunreflc

Time dependent load curves are usually applied to the following boundary or loading conditions on each node. By default, those curves are copied to all MPP subdomain without checking for the presence of that node in the domain or not. The curves are evaluated every cycle and may consume substantial CPU time. This command will remove curves which are not referenced in the MPP subdomain for the following keywords.

*BOUNDARY_PRESCRIBED_MOTION_NODE

*LOAD_NODE

*LOAD_SHELL_ELEMENT

*LOAD_THERMAL_VARIABLE_NODE

timing_start *n*

Begin timing of element calculations on cycle n. The appearance of this option will trigger the generation of a file named DECOMP_TIMINGS.OUT during normal termination. This file will contain information about the actual time spent doing element calculations, broken down by part.

timing_end *n*

End timing of element calculations on cycle n. A reasonable value is probably timing_start + 50 or 100.

timing_file *filename*

The file filename is assumed to be the output file DECOMP_TIMINGS.OUT from a previous run of this or a similar model. The computational cost of each part that appears in this file is then used during the decomposition instead of the built in internal value for that part. Matching is based strictly on the user part ID. The first two lines of the file are skipped, and only the first two entries on each of the remaining lines are relevant (the part ID and the cost per element).

contact:

This section has been largely replaced by the_MPP option on the normal contact card. The only remaining useful option here is:

alebkt n

Sets the bucket sort frequency for FSI (fluid structure interaction) to once every **n** cycles.
default = 50

general: Holds general options

lstc_reduce

If this option appears, LSTC's own reduce routine is used to get consistent summation of floating point data among processors. See also, *CONTROL_MPP_-IO_LSTC_REDUCE which has the same effect..

nodump

If this option appears, all restart dump file writing will be suppressed: d3dump, runrsf, and d3full files will not be written

nofull

If this option appears, writing of d3full (full deck restart) files will be suppressed.

nod3dump

If this option appears, writing of d3dump and runrsf files will be suppressed.

runrsfonly

If this option appears, writing of d3dump files will not occur – runrsf files will be witten instead. Any time a d3dump OR runrsf file would normally be written, a runrsf file will be written.

nofail

If this option appears, the check for failed elements in the contact routines will be skipped. This can improve efficiency if you do not have element failure in the model.

swapbytes

If this option appears, the d3plot and interface component analysis files are written in swapped byte ordering.

nobeamout

Generally, whenever a beam, shell, or solid element fails, and element failure report is written to the d3hsp and message files. This can generate a lot of output. If this option appears, the element failure report is suppressed.

Special Decompositions:

These options appear in the "decomposition" section of the pfile and are only valid if the decomposition method is **rcb**. The rcb decomposition method works by recursively dividing the model in half, each time slicing the current piece of the model perpendicularly to one of the three coordinate axes. It chooses the axis along which the current piece of the model is longest. The result is that it tends to generate cube shaped domains aligned along the coordinate axes. This is inherent in the algorithm, but is often not the behavior desired.

This situation is addressed by providing a set of coordinate transformation functions which are applied to the model before it is decomposed. The resulting deformed geometry is then passed to the decomposition algorithm, and the resulting domains are mapped back to the undeformed model. As a simple example, suppose you wanted rectangular domains aligned along a line in the xy plane, 30 degrees from the x axis, and twice as long along this line as in the other two dimensions. If you applied these transformations:

```
    sx 0.5
    rz -30
```

then you would achieve the desired effect.

Furthermore, it may be desirable for different portions of the model to be decomposed differently. It is now possible to specify different regions of the model to be decomposed with different transformations. The general form for a special decomposition would look like this:

```
decomposition {
        region { <region specifiers> <transformation> <grouping> }
        region { <region specifiers> <transformation> <grouping> }
        <transformation>
}
```

Where the region specifiers are logical combinations of **box, sphere, clinder, parts,** and **silist**.
The transformation is a series of **sx, sy, sz, rx, ry, rz, c2r, s2r, 3vec,** and **mat**. The grouping is either **lumped** or empty. The portion of the model falling in the first region will be decomposed according to the given transformation. Any remaining part of the model in the second region will then be treated, and finally anything left over will be decomposed according to the final transformation. Any number of regions may be given, including 0. Any number of transformations may be specified. They are applied to the region in the order given.

The region specifiers are:

box xmin xmax ymin ymax zmin zmax
A box with the given extents.

sphere xc yc zc r
The sphere centered at (**xc,yc,zc**) and having radius **r**. If **r** is negative it is treated as infinite.

cylinder xc yc zc ax ay az r d
A cylinder with center at (**xc,yc,zc**) and radius **r**, extending out in the direction of (**ax,ay,az**) for a distance of **d**. If **d** is 0, the cylinder is infinte in both directions.

parts n1 n2 n3 n4....

All parts whose user id matches one of the given values are included in the region. Any number of values may be given.

partsets n1 n2 n3 n4....

All partsets whose user id matches one of the given values will have all of their parts included in the region. Any number of values may be given.

silist n1 n2 n3 n4....

All elements involved in a contact interface whose user id matches one of the given values are included in the region.

The transformations available are:

local

This is only useful in conjunction with the *CONTROL_MPP_PFILE keyword option, when used inside a file included via *INCLUDE_TRANSFORM. At this point in the region processing, a transformation is inserted to invert the transformation used by *INCLUDE_TRANSFORM. The effect is that if this option appears before any other transformation options, all those options (and the subsequent decomposition) are performed in the coordinate system of the include file itself, not the global coordinate system..

sx t

scale the current x coordinates by t.

sy t

scale the current y coordinates by t.

sz t

scale the current z coordinates by t.

rx t

rotate around the current x axis by t degrees.

ry t

rotate around the current y axis by t degrees.

rz t

rotate around the current z axis by t degrees.

txyz x y z

translate by (x,y,z).

mat m11 m12 m13 m21 m22 m23 m31 m32 m33

transform the coordinates by matrix multiplication:

$$\text{Transformed Coordinates} = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

3vec v11 v12 v13 v21 v22 v23 v31 v32 v33

Transform the coordinates by the inverse of the transpose matrix:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} v_{11} & v_{21} & v_{31} \\ v_{12} & v_{22} & v_{32} \\ v_{13} & v_{23} & v_{33} \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

$$= \mathbf{V}^{\mathrm{T}} \times \text{transformed coordinates}$$

This appears complicated, but in practice is very intuitive: instead of decomposing into cubes aligned along the coordinate axes, rcb will decompose into parallelipipeds whose edges are aligned with the three vectors (v11, v12, v13), (v21, v22, v23), and (v31, v32, v33). Furthermore, the relative lengths of the edges of the decomposition domains will correspond to the relative lengths of these vectors.

C2R x0 y0 z0 vx1 vy1 vz1 vx2 vy2 vz2

The part is converted into a cylindrical coordinate system with origin at (x0, y0, z0), cylinder axis (vx1, vy1, vz1) and theta = 0 along the vector (vx2, vy2, vz2). You can think of this as tearing the model along the (vx2, vy2, vz2) vector and unwrapping it around the (vx1, vy1, vz1) axis. The effect is to create decomposition domains that are “cubes” in cylindrical coordinates: they are portions of cylindrical shells. The actual transformation is:

$$\text{new}(x, y, z) = \text{cylindrical coordinates}(r, \theta, z)$$

Knowing the order of the coordinates is important if combining transformations, as in the example below.

S2R x0 y0 z0 vx1 vy1 vz1 vx2 vy2 vz2

Just like the above, but for spherical coordinates. The (vx1,vy1,vz1) vector is the phi = 0 axis.

$$\text{new}(x, y, z) = \text{spherical coordinates}(\rho, \theta, \phi)$$

The grouping qualifier is:

lumped

Group all elements in the region on a single processor. If this qualifier is not given, the elements in the region are distributed across all processors.

Examples:

rz 45

will generate domains rotated -45 degrees around the z axis.

```
C2R 0 0 0 0 0 1 1 0 0
```

will generate cylindrical shells of domains. They will have their axis along the vector (0,0,1), and will start at the vector (1,0,0) Note that the part will be cut at (1,0,0), so no domains will cross this boundary. If there is a natural boundary or opening in your part, the "theta = 0" vector should point through this opening. Note also that if the part is, say, a cylinder 100 units tall and 50 units in radius, after the C2R transformation the part will fit inside the box x=[0,50], y=[0, 2PI), z=[0,100]. In particular, the new y coordinates (theta) will be very small compared to the other coordinate directions. It is therefore likely that every decomposition domain will extend through the complete transformed y direction. This means that each domain will be a shell completely around the original cylinder. If you want to split the domains along radial lines, try this pair of transformations:

```
C2R 0 0 0 0 0 1 1 0 0
SY 5000
```

This will do the above C2R, but then scale y by 5000. This will result in the part appearing to be about 30,000 long in the y direction -- long enough that every decomposition domain will divide the part in this (transformed) y direction. The result will be decomposition domains that are radial "wedges" in the original part.

General combinations of transformations can be specified, and they are applied in order:

```
SX 5 SY .2 RZ 30
```

will scale x, then y, then rotate.

A more general decomposition might look like:

```
decomposition { rx 45 sz 10
    region { parts 1 2 3 4 5 and sphere 0 0 0 200 lumped }
    region { box 0 100 –1.e+8 1.e+8 0 500 or sphere 100 0 200 200 rx 20 }
}
```

This would take elements that have user ID 1, 2, 3, 4, or 5 for their part, AND that lie in the sphere of radius 200 centered at (0,0,0), and place them all on one processor.

Then, any remaining elements that lie in the given box OR the sphere of radius 200 centered at (100,0,200) would be rotated 20 degress in x then decomposed across all processors. Finally, anything remaining would be rotated 45 degrees in x, scaled 10 in z, and distributed to all processors. In general, region qualifiers can be combined using the logical operations **and**, **or**, and **not**. Grouping using parentheses is also supported.

EXECUTION OF MPP/LS-DYNA

MPP/LS-DYNA runs under a parallel environment which provided by the hardware vendor. The execution of the program therefore varies from machine to machine. On some platforms, command line parameters can be passed directly on the command line. For others, the use of the names file is required. The names file is supported on all systems.

The serial/SMP code supports the use of the SIGINT signal (usually Ctrl-C) to interrupt the execution and prompt for user input, generally referred to as "sense switches." The MPP code also supports this capability. However, on many systems a shell script or front end program (generally "mpirun") is required to start MPI applications. Pressing Ctrl-C on some systems will kill this process, and thus kill the running MPP-DYNA executable. As a workaround, when the MPP code begins execution it creates a file "bg_switch" in the current working directory. This file contains the following single line:

```
rsh <machine name> kill -INT <PID>
```

where < machine name > is the hostname of the machine on which the root MPP-DYNA process is running, and <PID> is its process id. (on HP systems, "rsh" is replaced by "remsh"). Thus, simply executing this file will send the appropriate signal.

Here is a simple table to show how to run the program on various platforms. Of course, scripts are often written to mask these differences.

Platform	Execution Command
DEC Alpha	dmpirun –np n mpp-dyna
Fujitsu	jobexec –vp n –mem m mpp-dyna
Hitachi	mpirun –np n mpp-dyna
HP	mpp-dyna –np n
IBM	#!/bin./ksh export MP_PROC=n export MP_LABELIO=no export MP_EUILIB=us export MPI_EUIDEVICE=css0 poe mpp-dyna
NEC	mpirun –np n mpp-dyna
SGI	mpirun –np n mpp-dyna
Sun	tmrun –np n mpp-dyna

Where ***n*** is the number of processors, ***mpp-dyna*** is the name of the MPP/LS-DYNA executable, and ***m*** is the MB of real memory.

APPENDIX P: Implicit Solver

INTRODUCTION

The terms implicit and explicit refer to time integration algorithms. In the explicit approach, internal and external forces are summed at each node point, and a nodal acceleration is computed by dividing by nodal mass. The solution is advanced by integrating this acceleration in time. The maximum time step size is limited by the Courant condition, producing an algorithm which typically requires many relatively inexpensive time steps.

While explicit analysis is well suited to dynamic simulations such as impact and crash, it can become prohibitively expensive to conduct long duration or static analyses. Static problems such as sheet metal springback after forming are one application area for implicit methods. In the implicit method, a global stiffness matrix is computed, inverted, and applied to the nodal out-of-balance force to obtain a displacement increment. The advantage of this approach is that time step size may be selected by the user. The disadvantage is the large numerical effort required to form, store, and factorize the stiffness matrix. Implicit simulations therefore typically involve a relatively small number of expensive time steps.

In a dynamic implicit simulation these steps are termed *time steps*, and in a static simulation they are *load steps*. Multiple steps are used to divide the nonlinear behavior into manageable pieces, to obtain results at intermediate stages during the simulation, or perhaps to resolve a particular frequency of motion in dynamic simulations. In each step, an equilibrium geometry is sought which balances dynamic, internal and external forces in the model. The *nonlinear equation solver* performs an iterative search using one of several Newton based methods. *Convergence* of this iterative process is obtained when norms of displacement and/or energy fall below user-prescribed tolerances. Within each implicit iteration there is a *line search* performed for enhancing robustness of the procedure.

The implicit analysis capability was first released in Version 950. Initially targeted at metal forming springback simulation, this new capability allowed static stress analysis. Version 970 added many additional implicit features, including new element formulations for linear and modal analysis. A milestone in implicit was the advent of Version 971 in which implicit analysis was carried over to MPP and thus allowed much larger problems to be solved, and from there it has evolved extensively to presently contend well among competitive softwares. Still, it can be a gruesome task to set up an implicit input problem that runs all the way to completion with acceptable results, especially when contacts are involved. The purpose of this text is to explain keywords of interest and suggest values of important parameters in order to give users tools for developing own solution strategies to these kinds of problems.

A prerequisite for running implicit is to use a double precision version of LS-DYNA, and a machine with a significant amount of memory. A theoretical overview of implicit can be found in the LS-DYNA Theory Manual implicittheorychapter.

NONLINEAR IMPLICIT ANALYSIS

Activating Implicit Analysis

The keyword *CONTROL_IMPLICIT_GENERAL is used to activate the implicit method, and in principle it is sufficient to set

IMFLAG = 1
DT0 = some reasonable initial time step.

The initial time step should be chosen small enough to resolve the frequency spectrum of interest and/or provide decent convergence properties, but large enough to benefit from an implicit analysis. With no other implicit options, this converts an explicit input deck to a static implicit input deck with a constant time step throughout the simulation and the problem will terminate upon convergence failure. Time stepping strategies to prevent this are discussed Time Stepping Strategies

Singularities and Eigenvalue Analysis

The first concern in implicit analysis is to prevent singularities in the stiffness matrix, otherwise the chance of succeeding is close to none. The major source to a singular stiffness matrix is the presence of rigid body modes in a static problem, and these can be revealed in an eigenvalue analysis. This is done by using *CONTROL_IMPLICIT_EIGENVALUE and putting

NEIG = number of modes to extract.

Run an eigenvalue analysis and extract (if practically possible) enough modes to see all the rigid body modes just to get an impression of the properties of the model. The frequencies are in the output text file “eigout” and the mode shapes can be animated in the binary output file “d3eigv” using LS-PrePost. Some rigid body modes may come as a surprise and should be constrained, a typical example of this are beams that are free to rotate around its own axis. Other rigid body modes are a consequence of the nature of the problem and cannot be constrained without destroying the connection to reality, an example of this could be components that are to be constrained with contacts but are initially free to move. There are several strategies to deal with these latter rigid body modes, some are discussed in the context of contacts but here dynamics is used.

Dynamics and Intermittent Eigenvalue Analysis

Dynamics alleviates singular rigid body modes and is activated on *CONTROL_IMPLICIT_DYNAMICS by putting

IMASS = 1 or a negative number.

If the purpose is to solve a dynamic problem in the first place, then just use IMASS = 1 and don't bother about the other parameters, otherwise some other strategy is necessary. The idea is to get off to a start using dynamic analysis, and as the simulation gets to a point where rigid modes have been constrained by contacts, dynamics is turned off. The simplest way of doing this is to put

TDYBIR = 0
TDYDTH = time when contacts have been established and rigid body modes are constrained
TDYBUR = time after TDYDTH for fading out dynamics between TDYDTH and TDYBUR.

If the dynamic results are of no interest but just a way to proceed to the static solution, then it is recommended to use numerical damping to prevent unnecessary oscillations, i.e., put

GAMMA = 0.6
BETA = 0.38.

This should be enough to start up most problems of interest. A restriction with this is that when dynamics has been turned off it cannot be turned on again. If this is necessary for some reason, then a negative number of IMASS should be used to control dynamics through a load curve.

If it is hard to deduce how to choose the time when to turn off dynamics, the use of intermittent eigenvalues may be of great help. By putting

NEIG = a negative number

on *CONTROL_IMPLICIT_EIGENVALUE the user may extract eigenvalues at given time points during a nonlinear simulation and deduce from that if all rigid body modes have been eliminated.

The Geometric Stiffness Contribution

Stiffness singularities may also occur during simulation due to a complex global phenomenon involving the stress state and geometry. For instance could slender components with compressive stresses give rise to zero eigenvalues, commonly known as buckling. The mathematical explanation to these kinds of singularities is that the material and geometric stiffness contributions cancel out, and for this reason the geometric contribution to the stiffness matrix is in LS-DYNA optional. It is activated by putting

IGS = 1

on *CONTROL_IMPLICIT_GENERAL. Most often singularities due to this contribution have a negative effect on convergence although it sometimes helps. It is recommended to leave this as default and turn it on if other strategies fail. A controlled way of getting past singular points of this type is to use so called arc length methods for which the geometric stiffness should be turned on, this way of dealing with the problem is discussed in the Theory Manual arclengthchapter.

BFGS or Full Newton

The nonlinear solver parameters are set on *CONTROL_IMPLICIT_SOLUTION where the solver type is specified as

NSOLVR = nonlinear or linear implicit analysis option

By default, a nonlinear BFGS solution strategy is used where the stiffness matrix is reformed every 11th iteration and a maximum of 15 reformations is allowed (for a linear solution set NSOLVR = 1). These parameters are set by

ILIMIT = iterations between reforming stiffness matrix
MAXREF = maximum number of stiffness reformations

on *CONTROL_IMPLICIT_SOLUTION. Reforming the stiffness matrix is computationally expensive but stabilizes the solution procedure, and the best strategy in this context is very much problem dependent. The recommendation is to start with the default strategy and if necessary change them. For hard problems decrease ILIMIT and for problems that converge well increase ILIMIT. For really bad problems switch to full Newton (the safe bet) by for instance putting

ILIMIT = 1
MAXREF = 30.

Keep the maximum number of reformations reasonably low or otherwise it may take an unnecessary amount of time just to reach convergence failure.

Convergence

Tolerances

Convergence is based on changes in displacements, energies and optionally residual forces, and the tolerance levels are set by

DCTOL = Relative displacement tolerance DMTOL = Maximum displacement tolerance
ECTOL = Relative energy tolerance EMTOL = Maximum energy tolerance

RCTOL = Relative residual tolerance RMTOL = Maximum residual tolerance
ABSTOL = Absolute tolerance

on *CONTROL_IMPLICIT_SOLUTION. The tolerances on the left are based on Euclidian norms of the involved vectors, whereas the ones on the right are based on maximum norms. The maximum norm tolerances are optional and should be activated if increased accuracy is desired at the price of more implicit iterations. To keep things simple, the discussion that follows pertains only to the Euclidian norm tolerances.
By default, the progress of the equilibrium search is not shown to the screen but can be activated using

NLPRINT = nonlinear output diagnostics

input parameter. The box below shows a typical iteration sequence, where the norms of displacement and energy are displayed. When these norms are reduced below user prescribed tolerances (default 0.001 and 0.01, respectively), the iteration process is said to have *converged*, and the solution proceeds to the next time step. The message files, messag in SMP and mesXXXX in MPP, typically contain a whole lot more information that will be dealt with further in Output and Stretching for Convergence.

```
BEGIN static time step     3
=============================================================
               time =  1.50000E-01
  current step size =  5.00000E-02
Iteration:   1     *|du|/|u| =  3.4483847E-01     *Ei/E0 =  1.0000000E+00
Iteration:   2     *|du|/|u| =  1.7706435E-01     *Ei/E0 =  2.9395439E-01
Iteration:   3     *|du|/|u| =  1.6631174E-03     *Ei/E0 =  3.7030904E-02
Iteration:   4     *|du|/|u| =  9.7088516E-05     *Ei/E0 =  9.6749731E-08
```

Premature Convergence

The last of these parameters (ABSTOL) overrides the other three (DCTOL, ECTOL, RCTOL) in the sense that if the residual force is small enough, convergence is detected regardless if the other three criteria are fulfilled or not. It has been seen that this sometimes give rise to so called *premature convergence*, converged states are not *really* converged in the sense that the residual norm is small enough. This gives bad results and it is usually recommended to tighten this tolerance to 10^{-20} to prevent this. On the other hand, if the problem is very hard this may prevent convergence to the extent that going back to the default is more or less necessary. It is difficult to give a general recommendation that holds for every problem.

As for the other three parameters, the one that usually comes into practice is the displacement criterion DCTOL. The energy criterion ECTOL is often easy to fulfill and the residual criterion is disabled by default and there may be no reason to activate this (that is unless a completely Using Residual Tolerance is used, or if Stretching for Convergence arise). Using the default values on all three is usually good enough to give acceptable results in decent time. For problems with poor convergence, the question of tightening or relaxing these parameters (that is the displacement convergence criterion) is up for debate. It is tempting to believe that relaxing the constraints will give better convergence which may or may not be true. Sloppy convergence criteria may once again result in premature convergence, and this will have a negative effect on the subsequent steps. The general recommendation would have to be to leave these unchanged, and be aware that relaxing them may not result in getting further in the implicit simulation.

Using Residual Tolerance

A novel strategy that seems to make sense and works well for some problems is to only use tolerances on residual forces RCTOL. This relies much on that the degree of nonlinearity of the simulation model is moderate, but it could be well worth trying. The first thing to do would be to put DCTOL and ECTOL to large numbers to disable them and put RCTOL to a relatively small number, start with 0.01 and see how that works out. This must be complemented with an absolute tolerance on the residual forces to get past the initial stages where the residual forces are small enough to not fulfill the relative tolerance. This is done by putting ABSTOL to a negative number, which is a different absolute criterion compared to putting it to a positive number. Now, this number is very problem dependent as this says that convergence is attained as soon as the Euclidian norm of the residual force vector is smaller than the absolute value of ABSTOL. It goes without saying that this require running the problem for a few steps to monitor the residual norm in the message files as described below, this should give an indication on how to determine the ABSTOL value.

Convergence Norms

The degrees of freedom encountered in an LS-DYNA simulation are either *translational* or *rotational,* where the latter comes from the presence of beams and shells. Historically, convergence check is on norms of the translational degrees of freedom only, which is unsettling from the fact that rotational residual forces (moments) are equally important in an implicit simulation. It is therefore recommended to use

NLNORM = choice of convergence norms

on *CONTROL_IMPLICIT_SOLUTION to change this default (which is = 2). While NLNORM = 3 will incorporate residual degrees of freedom separately and is a more satisfying approach, the recommended option is to use NLNORM = 4 or NLNORM a negative number. NLNORM = 4 will treat the entire force and displacement vector, respectively, as one complete mathematical vector and the convergence norms and scalar products used for checks are simply the Euclidian norms of these vectors. This will make the energy norm unit consistent since each term in the scalar product is an energy quantity, either through force times displacement or moment times angular increment. The displacement norm and residual norm, however, contains a mix of translational and rotational quantities which means that displacements are summed to angular increments and forces are summed to moments. This is of course not pretty, but it can be interpreted as using a length scale of 1 for the rotational degrees of freedom, meaning that displacements are summed to 1 length unit times angular increments, and moments are summed to 1 length unit times forces. If the model is in length units of *mm*, and the element sizes are in the order of 1 *mm*, this seems to be a reasonable length scale. But if the length unit is something else, it is convenient to use NLNORM = -(length scale) to scale with the number that corresponds to 1 *mm*. If for instance the length unit is *m*, then NLNORM = -0.001 is presumably a reasonable choice.

Time Stepping Strategies

By default, LS-DYNA will terminate when a step fails to converge. This is unfortunate since it may just be that the step is too large to achieve convergence and taking smaller steps would solve this problem. Instead of starting all over with a smaller step size, set

IAUTO = 1 to activate automatic time stepping

on *CONTROL_IMPLICIT_AUTO. When the problem fails to converge LS-DYNA will with this option go back to the previously converged state and retry with a smaller step size. If the problem converges well, the step size is increased for subsequent steps. This process of backing up and retrying difficult steps lends much persistence to the analysis, and is often the only procedure for solving highly nonlinear problems short of adjusting the step size manually and the recommendation is to always turn automatic time stepping on.

Another parameter worth mentioning in this context is

DTMAX = Max time step allowed, or a negative number

on the same card. This is the maximum step size possible and should be chosen so as to not loose necessary information in the results, i.e., be sure to resolve frequencies, contacts and nonlinear material response to a satisfactory degree. A negative value of this parameter will give the maximum step size as a function of the simulation time and also allows for hitting key points that are of interest. The user may for instance look for the peak stress for a certain external load and hitting the point in time when this happens is crucial. It also allows for taking smaller steps during critical stages in the simulation without wasting resources away from these stages.

Line Search

A good line search strategy is crucial in solving nonlinear implicit problems, without it only simple problems would be solvable. There are a few line search options available and these are activated by

LSMTD = Line search method

on *CONTROL_IMPLICIT_SOLUTION. The default method is based on minimizing a potential energy along the search direction and at the same time keeping track of the residual force magnitude. It also detects when a BFGS step results in negative initial energy and the stiffness matrix is reformed for robustness (this can only happen when the stiffness matrix has negative eigenvalues), and even suppresses the occurrence of negative volumes. Line search type 2 is based on residual forces and is more robust than the others, the drawback is that it typically results in too small steps and is not practically useful. Line search type 3 is very similar to type 4, but is not tracking the residual force or avoids negative volumes to the same extent, and in sum there is no practical reason for choosing methods 1 through 3. Worth mentioning is however line search type 5 which combines the energy and residual method in a stricter sense. That is, it minimizes the potential energy just like type 4 but it only allows the residual norm to double at each implicit iteration. This has shown to be robust but on the other hand slow in convergence and is to be used for problems that have difficulties to converge with the default line search strategy. This has had a huge impact on the treatment of rubber models for instance. An interesting combination is to use line search method 5 together with the strategy of converging based onUsing Residual Tolerance.

Finally, the line search tolerance

LSTOL = Line search tolerance

on *CONTROL_IMPLICIT_SOLUTION is fine as it is, don't change it.

Output

As previously mentioned, the output to the message file is more extensive than that to the screen/stdout. To novice or average implicit users this information may be more than can be digested at first and this section may be LINEAR EQUATION SOLVER, but with experience the output can become really useful when running into convergence issues. For this discussion it is important to distinguish between different *types* of iterations, and we use the term *time/load step* to mean the actual advance in simulation time, *implicit iteration* to indicate points in the iterative solution procedure where a new search direction is obtained (corresponding to either Newton or BFGS) and *line search iteration* to indicate the search for an optimal step size along a given search direction. We begin by showing a typical iteration output for when NLPRINT = 3 following by a bulletin list explaining the content.

```
[1]BEGIN implicit dynamics step      2 t = 2.5849E-01
 ============================================================
                time =  2.58489E-01
   current step size =  1.58489E-01
.
.
.

[2]Newton step computed
 Initial translational energy =  1.4031740E-01
[3]Initial total energy         =  1.4031740E-01
 Initial residual norm        =  7.5755334E-01
[4]Translational nodes norm     =  7.1314670E-01
 Translational nodes max      =  1.9159278E-01 at node         440788
 Translational rigid body norm =  3.5324028E-03
 Translational rigid body max =  2.3897586E-03 at body        4000024
 Rotational rigid body norm   =  2.5553155E-01
 Rotational rigid body max    =  1.7834357E-01 at body        4000024

[5]Evaluated residual for full step
 Current translational energy =  9.0141343E-02
[6]Current total energy         =  9.0141343E-02
 Current residual norm        =  2.2038666E+00
 Translational nodes norm     =  2.1950775E+00
 Translational nodes max      =  1.0295509E+00 at node         440984
 Translational rigid body norm =  3.4677960E-02
 Translational rigid body max =  2.6701145E-02 at body        4000024
 Rotational rigid body norm   =  1.9354585E-01
 Rotational rigid body max    =  1.2669458E-01 at body        4000024

[7]Evaluated residual for step size  6.6666667E-01
 Current translational energy =  1.1123844E-01
 Current total energy         =  1.1123844E-01
 Current residual norm        =  1.5007375E+00
 Translational nodes norm     =  1.4922142E+00
 Translational nodes max      =  7.1187918E-01 at node         440984
 Translational rigid body norm =  1.6121536E-02
 Translational rigid body max =  1.2659274E-02 at body        4000024
 Rotational rigid body norm   =  1.5890244E-01
 Rotational rigid body max    =  1.0852930E-01 at body        4000024

 Evaluated residual for step size  3.3333333E-01
 Current translational energy =  1.2543971E-01
 Current total energy         =  1.2543971E-01
 Current residual norm        =  8.3754668E-01
 Translational nodes norm     =  8.1775945E-01
 Translational nodes max      =  3.0290699E-01 at node         440984
 Translational rigid body norm =  5.4770134E-03
 Translational rigid body max =  4.5377150E-03 at body        4000024
 Rotational rigid body norm   =  1.8089755E-01
 Rotational rigid body max    =  1.2913973E-01 at body        4000024

 Line search continues
 Max relative step            =  0.3499001E+00
[8]Lower bound                  =  3.3333333E-01
 Upper bound                  =  1.0000000E+00

 Evaluated residual for step size  7.7777778E-01
 Current translational energy =  1.0487486E-01
 Current total energy         =  1.0487486E-01
 Current residual norm        =  1.7421066E+00
 Translational nodes norm     =  1.7342352E+00
 Translational nodes max      =  8.2830775E-01 at node         440984
 Translational rigid body norm =  2.1464316E-02
 Translational rigid body max =  1.6693623E-02 at body        4000024
 Rotational rigid body norm   =  1.6402030E-01
 Rotational rigid body max    =  1.0799892E-01 at body        4000024

 Evaluated residual for step size  5.5555556E-01
 Current translational energy =  1.1674955E-01
 Current total energy         =  1.1674955E-01
 Current residual norm        =  1.2575584E+00
 Translational nodes norm     =  1.2472759E+00
 Translational nodes max      =  5.8455890E-01 at node         440984
 Translational rigid body norm =  1.1655353E-02
 Translational rigid body max =  9.2884622E-03 at body        4000024
 Rotational rigid body norm   =  1.6006264E-01
 Rotational rigid body max    =  1.1222751E-01 at body        4000024

[9]Line search converged in    2 iterations
[10]Max relative step           =  7.1106671E-01

[11]Iteration:    5    displacement            energy              residual
 ----------          not conv.             not conv.            converged
 norm ratio          1.013E+00             1.000E+00               n/a
 current norm        8.446E-01             1.403E-01            1.258E+00
 initial norm        8.338E-01             1.403E-01            7.575E+00
-------------    trans       rot       trans       rot        trans       rot
max  node  norm     8.533E-02   0.000E+00   2.052E-02   0.000E+00   5.846E-01
0.000E+00
   at node ID   4000674    9007526     440984    9007526     440984    9007526
-------------
       RB max   3.449E-02  6.613E-04 -5.507E-04  7.681E-05  9.288E-03  1.122E-
01
     at RB ID   4000024    4000024    4000024    4000024    4000024    4000024
```

Referring to the superscripts at the beginning of the lines in the excerpt above, and taking them in order of appearance, we have at

1. Beginning a *dynamic* implicit time/load step, the goal is to get to a given time point (in this case 2.5849E-01 using a step size of 1.58489E-01). Implicit time/load steps are either dynamic, *static* or *semidnmc*, where the last one indicates a transition phase between dynamics and statics.
2. A *Newton* implicit iteration is performed, meaning that a full reformation of the stiffness matrix has just been made. Other implicit iterations can be either a *linear* if at the first iteration of a time/load step, or *BFGS* if the implicit iteration corresponds to a quasi-Newton step.
3. Initial norms are displayed at the beginning of each implicit iteration, these numbers are used as reference for the line search algorithm when checking line search convergence.
4. Detailed initial norms are displayed, separating out translational/rotational as well as nodal/rigid body Euclidian/max norms. The node number and rigid body attaining the global maximum norms are displayed and can be used to spot critical points in the model, in this particular case node 440984 and part 4000024 would be the likely candidates to trouble.
5. A full step along the search direction is taken, as a candidate for the next implicit iteration.
6. Norms for the full step is displayed, the line search algorithm compares these with the ones from 3 to deduce if line search convergence criterion is fulfilled.
7. Line search is in this case needed, and information for further iterates along the search direction is displayed in analogy to previous line search iterates. If no line search had been needed, the string *Line search is skipped* would have been displayed.
8. The way line search is performed, each line search iteration narrows in on the optimal step and the bounds within which the optimal step is sought is continuously displayed. The size of this interval becomes smaller and smaller with

increasing number of line search iterations, but convergence should preferably be attained before it tends to zero.

9.Line search converged here in 2 iterations; as a rule of thumb the number of iterations for line search should be about 10 or less. For critical steps one might accept 20 iterations but that should be among the exceptions. If the string *Line search did not converge* is displayed, that is an indication of something being terribly wrong.

10.After each implicit iteration, *Max relative step* indicates how large steps have been taken during line search so far. It also indicates how well prescribed motion is approximated where a value of 1 is perfect. For values below 1 and in the presence of non-zero prescribed motion, convergence will not be accepted even though all norms are within prescribed tolerances. This will instead issue the warning *Convergence prevented due to unfulfilled boundary conditions* and iterations continue.

11.Diagnostics for iteration 5 is displayed, similar numbers as shown during line search but in another format. Furthermore, the initial norms here refer to the initial norms from the beginning of the implicit time/load step and not from the beginning of the implicit iteration. This table can be used to deduce how close the implicit time/load step is at converging. The single important number for comfortably assessing the accuracy of an implicit iterate is the residual norm, currently displayed as 1.258E+00 (bold-faced in the table), since this should be small for good results.

At equilibrium, the output ends with a summary on how convergence was achieved. It may look like

```
Convergence detected as a combination of
      1.Ratio of euclidian displacement
         Value =   2.3417E-03 vs Tolerance =  2.5000E-3
      2.Ratio of euclidian energy
         Value =   3.2269E-08 vs Tolerance =  1.0000E-2
```

which tells us that the ratio of displacement was below DCTOL = 2.5E-3 and the ratio of energy was below ECTOL = 1E-2. If other criteria are satisfied, these will be listed too.

So, with NLPRINT = 3 information from every force evaluation is given, including type of step taken, the line search step size, the potential energy value used in line search and the magnitude of the residual force. A point is made here, already Premature Convergence and will be repeated, and it is the following. The nonlinear implicit solver is solving for zero residual force, so basically the number observed for the magnitude of the residual force should be small upon convergence (bold-faced in the table). If it is not, then the convergence is premature and the results may not be correct and subsequent steps are in danger. What "small" means in this context is hard to say since this depends on units as well as loads and geometry of the problem, but a good sign is that the residual force does not grow by more than the external loads in the problem do. Another thing to observe is how the Line Search is doing, if the line search starts needing many iterations and very large variations in the residual force along the search direction is observed, then it is likely that something in the model is causing this behavior. A safe bet is that contact states are changed along the line search direction causing discontinuities in the residual force and indicates that some remodeling has to be done.

We will come back to this section when discussion Taking Advantage of ASCII Information and strategies to prevent them.

LINEAR EQUATION SOLVER

General

Within each equilibrium iteration, a linear system of equations of the form $\boldsymbol{K}\Delta\boldsymbol{u} = \boldsymbol{R}$ must be solved. To do this, the stiffness matrix $\boldsymbol{K}$ is factorized and applied to the out-of-balance load or residual $\boldsymbol{R}$, yielding a displacement increment $\Delta\boldsymbol{u}$. Storing and solving this linear system represents a large portion of the memory and CPU costs of an implicit analysis. LS-DYNA uses a multi-frontal sparse direct solver and nested dissection, and for a problem with N number of nodes, the number of operations and memory storage is asymptotically proportional to $N^{3/2}$ and $NlogN$, respectively. This should give a ballpark indication on the growth when increasing the size of a given model. It must be stressed however that it is *a priori* difficult to predict the actual value of these numbers as they are highly dependent on the problem solved, in particular on the nodal connectivity through elements and contacts. In the end we are left at qualitative guessing how to set memory flags or determining model size in order to push the limits for a given computer architecture.

We here focus on the memory, and the purpose is twofold. First it is a documentation of the memory diagnostics that is written to the output message files (messag in SMP and mesXXXX in MPP) of LS-DYNA for the linear solvers. The level of information is regulated by the

LPRINT = Linear solver diagnostics level

input parameter on *CONTROL_IMPLICIT_SOLVER, and this text will cover LPRINT = 2 as the information for higher values is of no particular interest to others than developers. Second it should help understanding the interrelationship between the size of a model and the memory required to obtain a solution. We emphasize however that different classes of problems most likely require different guidelines (shell and solid structures typically result in different matrix topologies for instance) and to this end the user is left at earning this experience on his own.

Memory

The memory is specified and reported in terms of *words*, where 1 word is equivalent to 4 bytes in single precision and 8 bytes in double precision arithmetic. To simplify things, a word can in this context be seen as the equivalent of a real number or an integer. For implicit calculations the executable used is typically double precision, so to convert from words to bytes one needs to multiply by 8.

In starting a simulation the user typically specifies the *static* memory, for instance in SMP

```
ls-dyna i = in.k memory = 200m
```

meaning that LS-DYNA tries to allocate 200 Mwords for storing most of the data. In MPP, the physical memory is distributed among processes and this option applies to each individual process. In MPP there is also an option to add a second memory flag, for instance

```
mpp-dyna i = in.k memory = 1000m memory2 = 200m
```

and this means that 200 Mwords are allocated for each process while 1000 Mwords is allocated on the first process to handle everything up to the point when the model data has been distributed among all the involved processes. The memory size may also be specified on *KEYWORD. *For implicit it is recommended to avoid memory2 in order to keep the memory balance between processors.*

When a certain feature requires a slot in the memory, this is reported in the output text files, for instance as

```
contracting memory to    2306940 implicit friction
expanding   memory to    2306955 joints
```

In the example above, the memory pointer after having allocated for implicit friction is at 2306940 and joints reserves 15 words and increment the memory pointer accordingly. So the memory used up to this point is roughly 2.3 Mwords for this particular process. At the end of the initialization a memory report is written and can look like

```
S t o r a g e   a l l o c a t i o n
Memory required to begin solution      :    2680153
Linear Alg dynamically allocated memory:     880685
Additional dynamically allocated memory:    1553130
                                 Total:    5113968
```

The total amount of memory used up to this point is roughly 5.1 Mwords, which is partitioned in the static memory specified and *dynamic* memory that is allocated on the fly so to speak, typically if a certain amount of memory needed cannot be estimated in due time to adequately include it in the static chunk of memory. Whenever a memory violation occurs, for instance when trying to allocate more memory than physically available or if more memory needs to be specified, LS-DYNA terminates with appropriate error information. The static memory for the linear solver is allocated thereafter, which is the topic for the next section.

Linear Solver Memory Consumption

For LPRINT = 2, some memory information is given in the output files, and this is here presented in order of execution, thus as they appear in the files. We repeat that this is the information obtained in the message files and is thus referring to the memory consumption for this particular process if MPP is considered. For the sake of completeness we also cover the diagnostics related to the CPU time for the involved stages in the linear solution.

First the memory required for handling constraints is reserved and is reported as

```
expanding   memory to    2893375 implicit lsolvr allocation 1
```

After this memory is reserved for symbolic factorization, which is performed prior to the actual solve to predict the storage requirement for the factorized system matrix. This allocation is reported as

```
expanding   memory to    7639758 implicit matrix storage
```

Also, some memory is reserved for the sparse matrix, including index pointers to non-zero elements, and is reported as

```
expanding   memory to   15077248 linear eqn.  solver allocation 2d
```

Here the last information (2d) is referring to the solver used and may change depending on user options.

Before the symbolic factorization takes place, current information on the system matrix and workspace reserved for implicit is reported

```
local number of rows      =             68490
local size of matrix      =           2338945
local len of workspace    =          15077248
ptr to start of wrkspc    =           2893375
```

The first number refers to the number of independent degrees of freedom in the model, i.e., the number of rows in the system matrix, and the second to the number of nonzero entries before factorization takes place, i.e., after assembly. The third row of information is somewhat misleading as this is actually the total amount of memory reserved and not the size of the workspace itself. The workspace here means the memory reserved for the linear solution, and its size is obtained by subtracting the pointer to start of workspace and the local length of workspace.

The symbolic factorization is now performed and information on the CPU time for doing this together with the estimated size of the factorized matrix is reported

```
CPU: symbolic factor     =         0.200
WCT: symbolic factor     =         0.203
storage currently in use =       1038915
storage needed           =       12913838
factor speedup           =     1.9197E+00
solve  speedup           =     1.9378E+00
est.  factor nonzeroes    =     5.1432E+07
est.  factor operations   =     1.2099E+11
est.  total factor nz.    =     9.9666E+07
est.  max.  factor op.    =     1.2099E+11
est.  total factor op.    =     2.3293E+11
est.  max.  factor nz.    =     5.1432E+07
```

The first two rows reports the CPU and wall clock time for doing the symbolic factorization in terms of seconds. The next two are storage requirements for the symbolic factorization, of which the latter is the memory reserved for this. After this we have estimations of the speedup in the factorization and subsequent solve of the linear system of equations, this is a report from an MPP run on 2 processors. The last lines refer to the estimated requirement for storing and factorizing the system matrix, and distinguish between the local, total and maximum number of each of these two entries. This allows for determining the memory scaling in the case of an MPP run.

Now LS-DYNA is in the position of reserving space for the factorization of the system matrix, and a report on the storage needed for this is given.

```
symbolic storage       1 =           1038915
in-core numeric storg  1 =          62762269
out-of-core num.  storg 1 =          13168676
symbolic         storg 1 =             1.04 Mw
in-core numericl storg 1 =            69.06 Mw
out-of-core num.  storg 1 =             14.51 Mw
```

Here basically the same information is repeated, except for a slight increment in order to account for potential penalty when doing the actual factorization. The first is again the memory reserved for the symbolic factorization and the other two are the memory required to perform an in-core and out-of-core factorization, respectively. If the memory available is not sufficient for doing an in-core factorization, LS-DYNA warns about this and attempts to do an out-of-core factorization, which means that the hard disk is used to store information during factorization. If this warning appears it is recommended to restart the simulation and using more memory since out-of-core significantly (writing to disk is more expensive than accessing memory) adds to the wall clock time for solving the problem. If the memory for doing an out-of-core factorization is insufficient, LS-DYNA will terminate with an error message. Otherwise, memory is reserved and reported

```
expanding   memory to  77737167 linear eqn.  solver numerical phase 1
```

meaning that the total amount of memory reserved at this point is roughly 78 Mwords. At this point a redistribution of the system matrix is performed and a short report is given on this

```
symbolic storage       2 =           1038915
in-core numeric storg  2 =          68238926
out-of-core num.  storg 2 =          11553502
```

Now the actual factorization takes place and a report from this is given where the information of interest is

```
CPU: factorization       =             59.025
WCT: factorization       =             50.151
act.  factor nonzeroes    =        5.1236E+07
act.  factor operations   =        1.2092E+11
act.  max.  factor nz.    =        5.1236E+07
act.  max.  factor op.    =        1.2092E+11
act.  total factor nz.    =        9.9666E+07
act.  total factor op.    =        2.3293E+11
```

which basically is the same information as from the symbolic factorization except that now it is the actual instead of the estimated values that is reported.

Finally, the forward and backward substitutions are allocated for and performed. The information given is

```
symbolic storage       3 =           1038915
in-core numeric storg  3 =          68299122
out-of-core num.  storg 3 =          16476693
CPU: numeric solve       =              0.191
WCT: numeric solve       =              0.194
```

Concluding, the wall clock time for solving the system of linear equations is reported

```
WCT: total imfslv_mf2    =             75.079
```

ELEMENTS AND MATERIALS

Implicit Accuracy

Implicit and explicit analysis differ in many respects, an important one is that the deformation during a single step is much larger in implicit than a typical one in explicit. In the context of elements and materials, the demand for stronger objectivity and higher accuracy in implicit is obvious. The notion of implicit executing the same algorithms as a corresponding explicit analysis is whence not adopted in general. This can optionally be further extended, using

IACC = 1 to increase implicit accuracy

on *CONTROL_ACCURACY will make selected elements strongly objective and enhance the accuracy for selected materials. For instance will finite rotations be represented exactly and fully iterative plasticity adopted. In addition, the flag applies to tied contacts as will be elaborated on below. Currently the following elements are supported for this option

Solid elements -2,-1,1,2
Shell elements 4,-16,16,23,24
Beam elements 1,2,9

and materials 24 and 123 use fully iterative plasticity.

CONTACTS

Contacts are probably among the hardest features to treat in a nonlinear implicit simulation. They are divided into the categories *tied* (bilateral) and *sliding* (unilateral) contact, and the characteristics of the two are quite different. Tied contacts are most often applied as constraints, only sometimes using a penalty formulation, and are fairly easy to deal with as they are only moderately nonlinear. In contrast, sliding contacts are exclusively implemented as penalty contacts and hard to deal with because of the unilateral condition. While the number of contacts, see *CONTACT, in LS-DYNA is abundant, we will here only present the few contacts that make most sense to use in implicit.

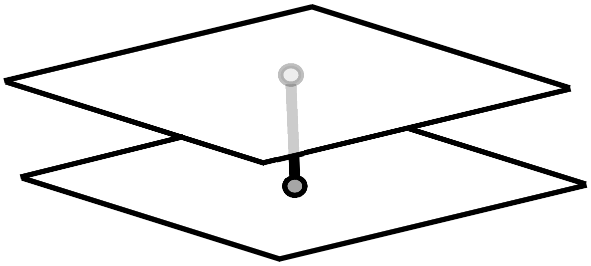
Tied Contacts

Tied contacts in implicit analysis should be accompanied with the Implicit Accuracy option, as this essentially reduces the number of relevant contacts to use to only six. These six contacts are

*CONTACT_TIED_NODES_TO_SURFACE
*CONTACT_TIED_NODES_TO_SURFACE_CONSTRAINED_OFFSET
*CONTACT_TIED_NODES_TO_SURFACE_OFFSET
*CONTACT_TIED_SHELL_EDGE_TO_SURFACE
*CONTACT_TIED_SHELL_EDGE_TO_SURFACE_CONSTRAINED_OFFSET
*CONTACT_TIED_SHELL_EDGE_TO_SURFACE_BEAM_OFFSET

The first and fourth of these project the nodes on the slave side to the master surface, while the rest are offset contacts for which nodes will remain in place. The fourth, fifth and sixth constrain rotations while the others do not. The third and sixth are penalty based, while the others are constraint based. In sum, these six contacts cover most reasonable scenarios.

The implicit accuracy option will in this context make these six tied contact strongly objective, and built-in intelligence will detect whether nodes contain rotations to constrain or not. A nice feature with the latter is that torsion is automatically applied with physical consistency, so it is for instance no problem to use single beams to model spotwelds between solid elements, and when using the same connection technique between shells the beam axial rotational degrees of freedom will be constrained to the translational degrees of the shells and thus avoiding the relatively weak (and non-physical) drilling degree of freedom. This situation is depicted below.



The choice of contact depends on the situation, but from a conceptual point of view it should be ok to use *CONSTRAINED_TIED_SHELL_EDGE_TO_SURFACE_CONSTRAINED_OFFSET for most cases. The only problem then would be if complicated geometry results in termination due to overconstraining, for which a switch to the corresponding penalty version *CONSTRAINED_TIED_SHELL_EDGE_TO_SURFACE_BEAM_OFFSET is motivated. See remarks in the LS-DYNA keyword manual for more information.

Sliding Contact

The choice of contact in this section is the Mortar contact as this seems to be best implicit contact algorithm when considering a combination of speed, accuracy and robustness. The Mortar contact features smoothness and continuity that is highly appreciated in implicit analysis, but is on the other hand expensive enough to not be recommended for explicit analysis. Many other contact algorithms are supported for implicit analysis and execute faster than the Mortar contact, but this is often seen when the infamous IGAP flag is set to default. This flag manipulates the stiffness matrix to the extent that accuracy may be deteriorated, and if used the results should be thoroughly checked. For Mortar contact IGAP has a different meaning as will be described IGAP. See remarks in the LS-DYNA keyword manual for more information.

Basics

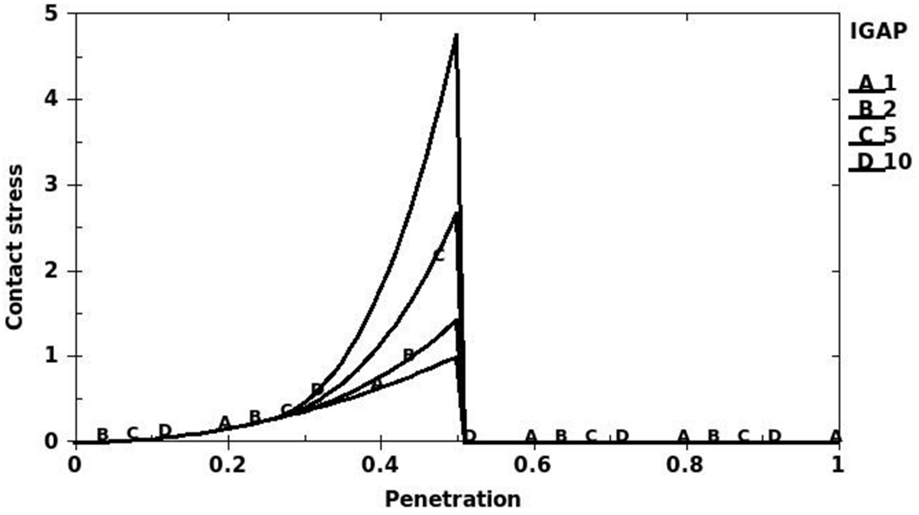
The Mortar contact is activated by typically appending the suffix MORTAR to the automatic single surface, automatic surface to surface or forming surface to surface keywords. It can also be run as tied and tiebreak contacts. All Mortar contacts are segment to segment and penalty based and the tied and tiebreak contacts are always offset, i.e., the tie occurs on the outer surfaces of shells and not on the mid surfaces. For automatic contacts, edge contact with flat edges is always active. At this point, it possesses features that are of particular interest to implicit and that are not available for other contacts. It is supported in both SMP and MPP but the option MPP does not apply, and the SMP ignore flag applies. The SOFT flag does not apply, to summarize it is a contact algorithm especially intended for implicit analysis.

Recommendations

For the forming contact the rigid tools must be meshed so that the normals are directed towards the blank, and contacts from above and below must be separated into two or more interfaces because contact can only occur from one side of the blank for a given contact interface. For the forming contact, rigid shells on the master side have no contact thickness. This is not the case for automatic contacts, here there are no restrictions on the mesh and even rigid shells have contact thickness. For all Mortar contacts, part or part set based slave and master sides are recommended although not mandatory. If the two sides in the contact interface have different stiffness, use the weak part as slave in order to get the best possible implicit convergence behavior. This is automatically taken care of in a single surface contact.

Characteristics

The contact pressure in the Mortar contact is a parabolic function of the penetration in combination with a cubic stiffening phase. In short, the contact stiffness depends on the slave side material and a characteristic length of the slave side segment. The characteristic length is for shells the shell thickness and for solids it is a median of the edges in the slave side of the contact interface, and the maximum penetration allowed is given as 95% of the average characteristic lengths on master and slave sides. For solids the definition of the characteristic length may have the consequence that the stiffness becomes unnecessarily high if some elements are much smaller than most, and stiffness adjustments may be necessary. Furthermore, the characteristic length also determines the maximum penetration as well as the search radius for finding contact pairs, for this reason the characteristic length can optionally be increased by assigning it on PENMAX on optional card B. In most cases it is expected that default value (=0.0) for this parameter will suffice.



IGAP

The contact stiffness is parabolic with respect to penetration up to a penetration depth corresponding to half of the maximum penetration. For IGAP = 1 it will remain parabolic for even larger penetrations but the user may increase IGAP which means that the contact will stiffen for larger penetrations, in fact it will become cubic according to the picture above. The purpose of increasing IGAP could be to prevent the penetration from becoming larger than the maximum allowed penetration, because if convergence is attained with penetrations larger than this value this contact will be released in subsequent steps and the simulation is likely destroyed. Penetrations of this depth are likely to cause discontinuities along line searches and other discouraging phenomena. The user may of course scale the stiffness by increasing SFS but this also scales the stiffness for small penetrations and probably has a negative effect on convergence.

Output for debugging

Just as for implicit in general, Output is always a good thing to have when convergence starts deteriorating and considering the release of contact in the previous section, it would be interesting to know if penetrations are large enough for this to be a potential danger. First, initial penetrations are always reported in the message file(s), including the maximum penetration and how initial penetrations are to be handled. The latter depends on the value of the IGNORE flag and this is dealt with Initial Penetrations. In addition, by putting

MINFO = 1

on *CONTROL_OUTPUT, LS-DYNA will report the absolute maximum penetration as well as the maximum penetration in percentage after each equilibrium. If the relative maximum penetration reaches above 99% a warning message is printed as this particular contact is close to being released. The output is exemplified in the following.

```
    Contact sliding interface          1
    Number of contact pairs          527

    Maximum penetration is  0.2447797E+00 between
    elements     306774 and     306672

    Maximum relative penetration is  0.2266694E+02 % between
    elements     306742 and     306733

    Contact sliding interface          2
    Number of contact pairs        16209

    Maximum penetration is   0.5027643E+00 between
    elements     219492 and      94935

    Maximum relative penetration is  1.0366694E+02 % between
    elements     219492 and      94935
*** Warning Penetration is close to maximum before release
```

This percentage value should ideally be kept below some 90 % to have some sort of comfort margin. There are three ways to reduce maximum relative penetrations, and these are *(i)* to increase IGAP, *(ii)* to increase SST for solids or *(iii)* to increase SFS. Note that by increasing SST for solids the contact stiffness will automatically be decreased, and this should be accompanied by increasing SFS by the square of the fraction increase of SST. That is, if SST is doubled then SFS should be increased four times, and if SST is tripled then SFS should be increased nine times, and so on. In this case even IGAP may have to be increased by some amount if being larger than unity in the first place.

Initial Penetrations

As mentioned above, initial penetrations are always reported in the message file(s), including the maximum penetration and how initial penetrations are to be handled. The IGNORE flag governs the latter and the options are

IGNORE < 0 Same functionality as the corresponding absolute value, but contact between segments belonging to the same part is ignored completely

IGNORE = 0 Initial penetrations will give rise to initial contact stresses, i.e., the slave contact surface is not modified

IGNORE = 1 Initial penetrations will be tracked, i.e., the slave contact surface is translated to the level of the initial penetrations and subsequently follow the master contact surface on separation until the unmodified level is reached

IGNORE = 2 Initial penetrations will be ignored, i.e., the slave contact surface is translated to the level of the initial penetrations, optionally with an initial contact stress governed by MPAR1

IGNORE = 3 Initial penetrations will be removed over time, i.e., the slave contact surface is translated to the level of the initial penetrations and pushed back to its unmodified level over a time determined by MPAR1

IGNORE = 4 Same as IGNORE = 3 but it allows for large penetrations by also setting MPAR2 to at least the maximum initial penetration

The use of IGNORE depends on the problem, if no initial penetrations are present there is no need to use this parameter at all. If penetrations are relatively small in relation to the maximum allowed penetration, then IGNORE = 1 or IGNORE = 2 seems to be the appropriate choice. For IGNORE = 2 the user may specify an initial contact stress small enough to not significantly affect the physics but large enough to eliminate rigid body modes and thus singularities in the stiffness matrix. The intention with this is to constrain loose parts that are initially close but not in contact by pushing out the contact surface using SFST and applying the IGNORE = 2 option. It is at least good for debugging problems with many singular rigid body modes. IGNORE = 3 is the Mortar interference counterpart, used for instance if there is a desire to fit a rubber component in a structure. With this option the contact surfaces are restored linearly in time from the beginning of the simulation to the time specified by MPAR1. A drawback with IGNORE = 3 is that initial penetration must be smaller than half the characteristic length of the contact or otherwise they will not be detected in the first place. For this reason IGNORE = 4 was introduced where initial penetrations may be of arbitrary size, but it requires that the user provides crude information on the level of penetration of the contact interface. This is done in MPAR2 which must be larger than the maximum penetration or otherwise and error termination will occur. IGNORE = 4 only applies to solid elements at the moment.

When eliminating penetrations by simulation for models with many parts, some parts may contain thin members that cause spurious self-contacts. These may be difficult to work around by only adjusting contact parameters, but fortunately there is rarely any loss of generality in ignoring contact within parts completely since those are usually not of interest in such a context. The option IGNORE < 0 was implemented for this purpose and is a way to approach this problem.

Damping

Damping can be activated in dynamic implicit analysis using VDC. A problem with contact damping in implicit is that the time step is usually large enough to not resolve the time in contact to get the desired damping effect. Often the situation becomes even worse, it is therefore not recommended to use damping.

TROUBLESHOOTING CONVERGENCE PROBLEMS

Convergence of the nonlinear equilibrium iteration process presents one of the greatest challenges to using the implicit mode of LS-DYNA. At risk of repeating what has already been mentioned, below are some useful troubleshooting approaches.

Eigenvalue Analysis

Many convergence problems in static implicit analysis are caused by unconstrained rigid body modes. These are created when an insufficient number of constraints are applied to the model, or when individual model parts are left disconnected. Eigenvalue analysis is an excellent diagnostic tool to check for these problems, both initially and at critical points in the simulation. The procedure for performing an eigenvalue analysis was discussed Singularities and Eigenvalue Analysis.

D3Iter Plot Database

To diagnose convergence trouble which develops in the middle of a simulation, get a picture of the deformed mesh. Adjust the d3plot output interval to produce an output state after every step leading up to the problematic time. An additional binary plot database named “d3iter” is available which shows the deformed mesh during each equilibrium iteration. This output is activated by

D3ITCTL = 1 to activate d3iter plot database

on *CONTROL_IMPLICIT_SOLUTION. View this database using LS-PrePost to detect abnormal displacements. The problem may become obvious, especially as deformation is magnified. If not, there is yet another flag to activate to get the residual forces into both this database as well as d3plot for fringing. Setting

RESPLT = 1 to get residual data to binary databases

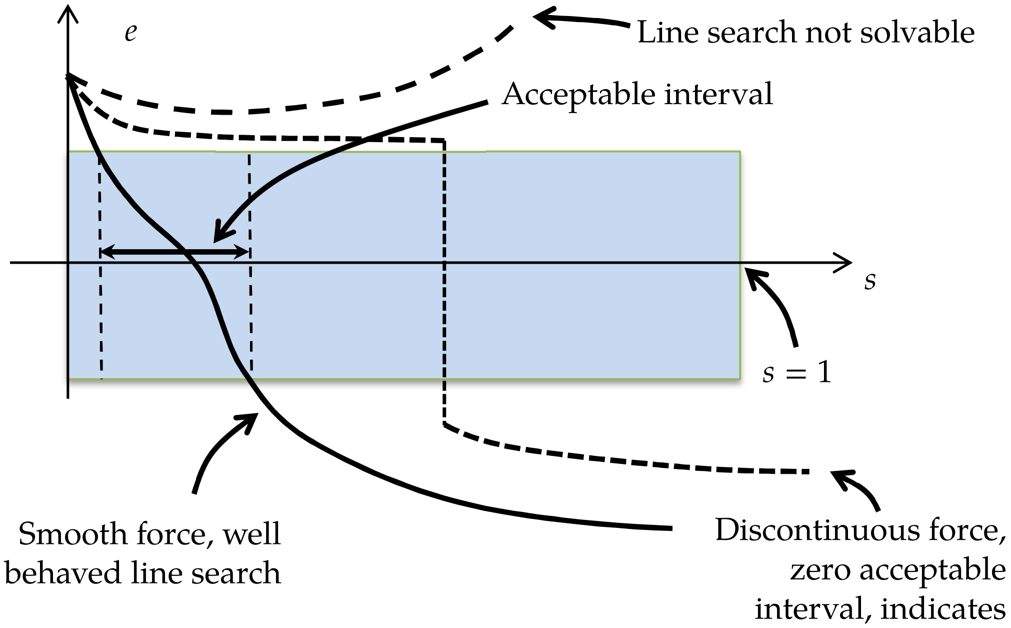
on *DATABASE_EXTENT_BINARY will do just that. With this option the residual forces are output to the d3plot and d3iter databases for fringing under the "NdV" menu. This is a great tool for locating areas in the model where the residual forces are not being reduced to a satisfactory level and take appropriate actions.

Taking Advantage of ASCII Information

If requested through Output on *CONTROL_IMPLICIT_SOLUTION and Output for debugging on *CONTROL_OUTPUT, a lot can be drawn from the information in the message file(s). This may be considered as a piece of advanced topic but may become useful in due time.

Residual Norm

Starting with the nonlinear output diagnostics, the following basic principle should be held in mind; *if the residual norm is zero, the problem is solved*. So it all comes down to make this number (11) small enough to trust the results. Therefore we suggest to monitor the residual norm and interpret it as an indicator of whether the convergence characteristics is good. Hopefully you would see this number decrease with implicit iterations and finally reach a relatively small number at convergence. Although it may (and will) increase on occasion, the trend should be a decreasing one. If this is a problem from the get-go, and you checked the model integrity through Singularities and Eigenvalue Analysis and common practice, it may be that a feature is not properly supported in implicit.



e
Line search not solvable
Acceptable interval
s
$s = 1$
Smooth force, well behaved line search
Discontinuous force, zero acceptable interval, indicates

Line Search

Another thing to look at is the line search convergence (9), as the relatively loose line search convergence tolerance should render a reasonably small number of line search iterations. A rule of thumb would be less than 10, at most 20. If more is used, or if the line search does not converge, there may be something in the model causing a jump in the residual forces. This is not supposed to happen in implicit, and may suggest a bug. Either of the two following scenarios for a feature discontinuity (and many line search iterations) is possible

- The line search step size becomes ridiculously small, and the current residual norm (7) is not approaching the initial one (3).
- The interval in which the optimal step is to be found (8) becomes ridiculously small, and the residual norm on the left and right interval points are not approaching eachother. This situation is depicted in the figure above (dashed line).

Many line search iterations can of course be due to high nonlinearities but if the above is observed it should probably be investigated, for instance by trying to identify model features that may be causing the bad behavior. If it is obvious that a discontinuity exist, like if the step size goes down to ~1^{-100} or the interval size becomes zero, and it is not due to a model error, please send a bug report.

Stretching for Convergence

If the problem runs fine for a significant number of load/time steps with subsequent convergence issues, then the information received up to this point may be used intelligently and suggest a different (unorthodox) implicit strategy. To justify the approach, we begin by recalling that the convergence is detected by the following numbers becoming small; $d = \|\Delta \boldsymbol{u}\|$ (displacement norm), $r = \|\boldsymbol{R}\|$ (residual norm) and $e = |\boldsymbol{R}^T \Delta \boldsymbol{u}|$ (energy norm). But all these numbers are linked through the General, $\boldsymbol{K}\Delta \boldsymbol{u} = \boldsymbol{R}$, so if $\boldsymbol{K}$ reasonably "nice" at all times it doesn't matter which numbers we use for detecting convergence. A mathematical way to state this is; if the condition number of $\boldsymbol{K}$ is "good", then all the norms are equivalent throughout the solution process. An intuitive statement would be to say that the problem is only moderately nonlinear, and convergence is usually never a problem. *But*, what happens if the properties of $\boldsymbol{K}$ are not that nice, or if $\boldsymbol{K}$ shifts character every time it is reformed? This is sort of saying that the problem is highly nonlinear, for instance due to frequent change of contact state or onsets/offsets of plasticity. In those cases the equivalence between d and r is lost, and typically an oscillatory behavior of the displacement norm is observed *even though the residual norm is reducing*, which in turn may lead to potential convergence problems just because the displacement criterion cannot be satisfied. Or equally bad, it could lead to Premature Convergence just by the coincidence that the displacement norm happens to become small. In those cases we essentially want to come up with a strategy where the displacement criterion is taken out of the convergence check and instead detect convergence based on residual only.

So assume we have say 10 converged states, after which the convergence problems begin. Then we can look at what the residual norm is for each of these converged states (11), for instance we may see the sequence

Step 1, residual norm 2859
Step 2, residual norm 1581
Step 3, residual norm 2119
Step 4, residual norm 2511
Step 5, residual norm 11570
Step 6, residual norm 4904
Step 7, residual norm 3586
Step 8, residual norm 3157
Step 9, residual norm 3315
Step 10, residual norm 2825.

For each of these steps we may validate the results, by post-processing contact force curves, checking force/moment balance etc., in LS-PrePost, and usually there is a correlation between "good results" and small residual norms. Step 5 above is for instance an outlier and may be associated with a prematurely converged step, something that can be confirmed or rejected from looking at the result. The goal with these observations is to establish a reasonable residual norm for which we can safely say that the results are good, and the numbers above indicates that 3000 may be a good candidate. The strategy is then to simply put DCTOL = 1.e-16 to ignore the displacement and set ABSTOL = -3000, which means we can be assured that convergence will not be detected until the residual norm is below 3000, and we "know" from having learned about the problem that this will yield good results. It should be mentioned that this may not work if the character (force level) changes significantly later on in the simulation, as 3000 then may not be the number we would trust.

Contacts

Continuing with the Output for debugging, a general rule is that there should be *no* warnings of large penetrations and preferably the maximum relative penetration should be below 90%. These numbers can be monitored after each load/time step. If large penetrations occur, either increase IGAP or contact stiffness, whatever makes most sense, but first make sure that the converged step is *really* converged (no Premature Convergence) by monitoring the residual forces and checking results as indicated above.

CHECKLIST

So, to summarize

✓Activate implicit by setting IMFLAG = 1 on *CONTROL_IMPLICIT_GENERAL

- Set DT0 to a reasonable initial time step

✓Check possible singularities in an eigenvalue analysis by requesting NEIG eigenvalues on *CONTROL_IMPLICIT_EIGENVALUE

- •Find a way to constrain spurious modes

✓Initially use dynamic analysis if rigid body modes are present in a static problem by setting IMASS = 1 on *CONTROL_IMPLICIT_DYNAMICS

- •Use TDYBIR, TDYDTH and TDYBUR
- •Use numerical damping by putting GAMMA = 0.6 and BETA = 0.38

✓Only activate geometric stiffness contribution as a last resort, except for arc length methods

✓Use default BFGS parameters ILIMIT and MAXREF on *CONTROL_IMPLICIT_SOLUTION

- •Increase or decrease ILIMIT based on convergence characteristics
- •Use full Newton (ILIMIT = 1) for hard problems
- •Keep MAXREF to a reasonably low number

✓Use default convergence tolerances DCTOL, ECTOL and RCTOL on *CONTROL_IMPLICIT_SOLUTION

- •ABSTOL may be set to 1.e-20 to prevent premature convergence
- •Relaxing DCTOL may not necessarily result in better convergence
- •Future strategy may be to focus on residual

✓Activate automatic time stepping on *CONTROL_IMPLICIT_AUTO

- •Set DTMAX to a number that resolves features of interest
  - ▪Contacts
  - ▪Material nonlinearities
  - ▪Frequencies

✓Use default line search method on *CONTROL_IMPLICIT_SOLUTION

- •Switch to LSMTD = 5 if hard problem (typically rubbers)
- •Don’t change LSTOL

✓Use at least NLPRINT = 2 on *CONTROL_IMPLICIT_SOLUTION to get convergence diagnostics into log files

- •Use NLPRINT = 3 to thoroughly track the residual norm and monitor line search behavior if debugging model is necessary

✓Use *CONTROL_IMPLICIT_SOLVER only in special occasions

✓Set D3ITCTL = 1 on *CONTROL_IMPLICIT_SOLUTION to get a database with Newton iterates when debugging a model

- •Complement this with RESPLOT = 1 on *DATABASE_EXTENT_BINARY to get the possibility to fringe the residual force vector

✓For forming Mortar contact, make sure

- •Tools are oriented towards the blank

- •Contact on top and bottom of blank are separated among contact interfaces

✓Use part (set) definitions of slave and master in Mortar contact

✓Always use weak part as slave in a Mortar contact definition to get best possible convergence

✓Set SST to a reasonable characteristic length for slave side consisting of solid elements

- •With this option, separate solids and shells into different contact interfaces in order to not manipulate the contact thickness for shells

✓ If penetrations are large, activate penetration diagnostics on *CONTROL_CONTACT

✓To avoid release of contact pairs, either

- •Increase stiffness scaling factor SFS
- •Increase IGAP for progressive stiffness increase
- •Increase SST for solids while at the same time increasing SFS

✓Use IGNORE appropriately to deal with initial penetrations

- •Check initial penetrations in message file

✓Don't use contact damping

APPENDIX Q: User Defined Weld Failure

The addition of a user weld failure subroutine into LS-DYNA is relatively simple. The UWELDFAIL subroutine is called every time step when OPT = 2 is specified in MAT_-SPOTWELD. As data, the identification number for the spotweld material, six constants specified in the input by thfe locations NRR through MTT, the radius of the cross section of the spotwelds, the current time, and the current values of the resultants for the spotwelds, which are stored in array STRR, are passed to the subroutine. The subroutine loops over the welds from LFT through LLT, and sets the values of the failure flag array FLAG.

```
      SUBROUTINE UWELDFAIL(IDWELD,STRR,FAIL,FIBL,CM,TT,LFT,LLT)
C******************************************************************
C|  LIVERMORE SOFTWARE TECHNOLOGY CORPORATION  (LSTC)             |
C|  ------------------------------------------------------------  |
C|  COPYRIGHT 2002 JOHN O.  HALLQUIST, LSTC                        |
C|  ALL RIGHTS RESERVED                                           |
C******************************************************************
C
C***  SPOTWELD FAILURE ROUTINE
C
C***  LOCAL COORDINATES: X IS TANGENT TO BEAM, Y & Z ARE NORMAL
C
C***  VARIABLES
C          IDWELD ---- WELD ID NUMBER
C          STRR ------ STRESS RESULTANTS
C                      (1) AXIAL (X DIRECTION) FORCE
C                      (2) Y SHEAR FORCE
C                      (3) Z SHEAR FORCE
C                      (4) MOMENT ABOUT Z
C                      (5) MOMENT ABOUT Y
C                      (6) TORSIONAL RESULTANT
C          FAIL ------ FAILURE FLAG
C                      = 0 NOT FAILED
C                      = 1 FAIL ELEMENT
C          FIBL ------ LOCATION (1,*) GIVES THE SPOTWELD DIAMETER
C          CM -------- 6 CONSTANTS SUPPLIED BY USER
C          TT -------- CURRENT SIMULATION TIME
C          LFT,LLT --- DO-LOOP RANGE FOR STRR
C
      DIMENSION IDWELD(*),STRR(6,*),FAIL(*),CM(*),FIBL(5,*)
C
C
      RETURN
      END
```

APPENDIX R: User Defined Cohesive Model

The addition of a user cohesive material subroutine into LS-DYNA is relatively simple. The UMATiC subroutine is called every time step where i ranges from 41 to 50**.** Input for the material model follows the *MAT_USER_DEFINED_MATERIAL definition. The user has the option of providing either a scalar or vectorized subroutine. As discussed in the Remarks for the user-defined material, the first two material parameters are reserved to specify how the density is treated and the number of integration points required for the failure of the element.

The cohesive model calculates the tractions on the mid-surface of the element as a function of the differences of the displacements and velocities of the upper (defined by nodes 5-6-7-8) and lower surfaces (defined by nodes 1-2-3-4). The displacements, velocities, and the calculated tractions are in the local coordinate system of the element, where the first two components of the vectors are in the plane of the mid-surface and the third component is normal to the mid-surface.

A stiffness must also be calculated by the user for the explicit time step calculation in LS-DYNA. This stiffness must provide an upper bound on the stiffness in all three directions.

The material fails at an integration point when `ifail=.true`. For an element to be deleted from the calculation, the number of integration points specified by the second material parameter must fail. If the second parameter is zero, elements cannot fail regardless of the specification of IFAIL in the user-defined material input. For example, the user may choose to reject an implicit step is the displacement increment is too

For implicit analysis, the subroutine is called with `maketan=.true.` and the user must provide the elastic moduli in the three local directions in the respective diagonal terms of the `dsave` array. The parameter `reject`, if set to `.true.` by the user, will signal to LS-DYNA that the current implicit iteration is unacceptable. For example, the user may choose to reject an implicit step if the traction changes too much from the last time step. In this situation, LS-DYNA will print a warning message `Material model rejected current iterate' and retry the step with a smaller time step. If chosen carefully (by way of experimenting), this may result in a good trade-off between the number of implicit iterations per step and the step size for overall speed.

The following example is a vectorized model with two elastic constants and failure:

```
      subroutine umat41c(idpart,cm,lft,llt,fc,dx,dxdt,aux,ek,
     & ifail,dt1siz,crv,nnpcrv,nhxbwp,cma,maketan,dsave,ctmp,elsiz,
     & reject,ip,nip)
      include 'nlqparm'
c
```

```
c***  vector cohesive material user model example
c
c***  variables
c          idpart ---- Part ID
c          cm -------- material constants
c          lft,llt --- start and end of block
c          fc -------- components of the cohesive force
c          dx -------- components of the displacement
c          dxdt ------ components of the velocity
c          aux ------- history storage
c          ek -------- max.  stiffness/area for time step calculation
c          ifail ----- =.false.  not failed
c                      =.true.  failed
c          dt1siz ---- time step size
c          crv ------- curve array
c          nnpcrv ---- # points per curve for crv array
c          nhxbwp ---- internal element id array, lqfinv(nhxbwp(i),2)
c                      gives external element id
c          cma ------- additional memory for material data defined by
c                      LMCA in 2nd card, 6th field of *MAT_USER_DEFINED
c          maketan --- true for implicit
c          dsave ----- material stiffness array, define for implicit
c          ctmp ------ current temperature
c          elsiz ----- characteristic element size (=sqrt(area))
c          reject ---- set to .true.  if this implicit iterate is
c                      to be rejected for some reason (implicit only)
c          ip -------- integration point number
c          nip ------- total number of integration points
c
c***  dx, dxdt, and fc are in the local coordinate system:
c     components 1 and 2 are in the plane of the cohesive surface
c     component 3 is normal to the plane
c
c***  cm storage convention
c     (1) =0 density is per area
c         =1 density is per volume
c     (2) number of integration points for element deletion
c         =0 no deletion
c     (3:48) material model constants
c
      logical ifail,maketan,reject
      dimension cm(*),fc(nlq,*),dx(nlq,*),dxdt(nlq,*),
     &          aux(nlq,*),ek(*),ifail(*),dt1siz(*),crv(101,2,*),
     &          nhxbwp(*),cma(*),dsave(nlq,6,*),ctmp(*),elsiz(*)
      integer nnpcrv(*)
c
c
      et=cm(3)
      en=cm(4)
      eki=max(et,en)
      fcfail=cm(5)
c
      do i=lft,llt
        fc(i,1)=et*dx(i,1)
        fc(i,2)=et*dx(i,2)
        fc(i,3)=en*dx(i,3)
        ek(i)=eki
        ifail(i)=fc(i,3).gt.fcfail
      enddo
c
      if(maketan) then
        do i=lft,llt
          dsave(i,1,1)=et
          dsave(i,2,1)=0.
          dsave(i,3,1)=0.
          dsave(i,1,2)=0.
          dsave(i,2,2)=et
          dsave(i,3,2)=0.
          dsave(i,1,3)=0.
          dsave(i,2,3)=0.
          dsave(i,3,3)=en
        enddo
      endif
      return
      end
```

The second example implements the Tveergard-Hutchinson cohesive model with failure in both the vectorized (UMAT42C) and scalar (UMAT43C) forms. Note the LFT and LLT are passed to the scalar version, however their value is zero.

```
      subroutine umat42c(idpart,params,lft,llt,fTraction,jump_u,dxdt,
     & aux,ek,ifail,dt1siz,crv,nnpcrv,nhxbwp,cma,maketan,dsave,ctmp,elsiz,
     & reject,ip,nip)
      include 'nlqparm'
c
c***  vector cohesive material user model example
c
c     Tveergard-Hutchinson model based on:
c     tahoe/src/elements/cohesive_surface/cohesive_models/TvergHutch3DT.cpp
c
c     the declaration below is processed by the C preprocessor and
c     is real*4 or real*8 depending on whether LS-DYNA is single or double
c     precision
c
      REAL L,jump_u
      logical ifail,maketan,reject
      dimension params(*),fTraction(nlq,*),jump_u(nlq,*),dxdt(nlq,*),
     &          aux(nlq,*),ek(*),ifail(*),dt1siz(*),crv(101,2,*),
     &          nhxbwp(*),cma(*),dsave(nlq,6,*),ctmp(*),elsiz(*)
      integer nnpcrv(*)
c
      fsigma_max=params(3)
      fd_c_n=params(4)
      fd_c_t=params(5)
      fL_1=params(6)
      fL_2=params(7)
      fpenalty=params(8)
c
      fK=fpenalty*fsigma_max/(fL_1*fd_c_n)
c
      fac=min(fd_c_n/fd_c_t**2,1./fd_c_n)
c
      do i=lft,llt
      u_t1 = jump_u(i,1)
      u_t2 = jump_u(i,2)
      u_n = jump_u(i,3)
c
      r_t1 = u_t1/fd_c_t
      r_t2 = u_t2/fd_c_t
      r_n = u_n/fd_c_n
      L = sqrt(r_t1*r_t1 + r_t2*r_t2 + r_n*r_n)
c
      if (L .lt.  fL_1) then
         sigbyL=fsigma_max/fL_1
      else if (L .lt.  fL_2) then
         sigbyL = fsigma_max/L
      else if (L .lt.  1.) then
         sigbyL = fsigma_max*(1.  - L)/(1.  - fL_2)/L
      else
         sigbyL = 0.0
         ifail(i)=.true.
      endif
c
      fTraction(i,1) = sigbyL*r_t1*(fd_c_n/fd_c_t)
      fTraction(i,2) = sigbyL*r_t2*(fd_c_n/fd_c_t)
      fTraction(i,3) = sigbyL*r_n
c
c     penetration
      if (u_n .lt.  0) fTraction(i,3)=fTraction(i,3)+fK*u_n
c
c     approximate stiffness for time step
      if (u_n .lt.  0) then
        ek(i)=fac*sigbyL+fK
      else
        ek(i)=fac*sigbyL
      endif
c
      if (maketan) then
        dsave(i,1,1)=sigbyL/fd_c_t*(fd_c_n/fd_c_t)
        dsave(i,2,1)=0.
        dsave(i,3,1)=0.
        dsave(i,1,2)=0.
        dsave(i,2,2)=sigbyL/fd_c_t*(fd_c_n/fd_c_t)
        dsave(i,3,2)=0.
        dsave(i,1,3)=0.
        dsave(i,2,3)=0.
        dsave(i,3,3)=sigbyL/fd_c_n
        if (u_n.lt.0) dsave(i,3,3)=dsave(i,3,3)+fk
      endif
      enddo
c
      return
      end
c
c
c
      subroutine umat43c(idpart,params,lft,llt,fTraction,jump_u,dxdt,
     & aux,ek,ifail,dt1siz,crv,nnpcrv,nhxbwp,cma,maketan,dsave,ctmp,elsiz,
     & reject,ip,nip)
c
c***  scalar cohesive material user model example
c
c     Tveergard-Hutchinson model based on:
c     tahoe/src/elements/cohesive_surface/cohesive_models/TvergHutch3DT.cpp
c
c     the declaration below is processed by the C preprocessor and
c     is real*4 or real*8 depending on whether LS-DYNA is single or double
c     precision
c
      REAL L,jump_u
      logical ifail,maketan,reject
      dimension params(*),fTraction(nlq,*),jump_u(nlq,*),dxdt(nlq,*),
     &          aux(nlq,*),ek(*),ifail(*),dt1siz(*),crv(101,2,*),
     &          nhxbwp(*),cma(*),dsave(nlq,6,*),ctmp(*),elsiz(*)
      integer nnpcrv(*)
c
      fsigma_max=params(3)
      fd_c_n=params(4)
      fd_c_t=params(5)
      fL_1=params(6)
      fL_2=params(7)
      fpenalty=params(8)
c
      fK=fpenalty*fsigma_max/(fL_1*fd_c_n)
c
      fac=min(fd_c_n/fd_c_t**2,1./fd_c_n)
c
      u_t1 = jump_u(1)
      u_t2 = jump_u(2)
      u_n = jump_u(3)
c
      r_t1 = u_t1/fd_c_t
      r_t2 = u_t2/fd_c_t
      r_n = u_n/fd_c_n
      L = sqrt(r_t1*r_t1 + r_t2*r_t2 + r_n*r_n)
c
      if (L .lt.  fL_1) then
         sigbyL=fsigma_max/fL_1
      else if (L .lt.  fL_2) then
         sigbyL = fsigma_max/L
      else if (L .lt.  1.) then
         sigbyL = fsigma_max*(1.  - L)/(1.  - fL_2)/L
      else
         sigbyL = 0.0

      ifail=.true.
      endif
c
      fTraction(1) = sigbyL*r_t1*(fd_c_n/fd_c_t)
      fTraction(2) = sigbyL*r_t2*(fd_c_n/fd_c_t)
      fTraction(3) = sigbyL*r_n
c
c     penetration
      if (u_n .lt.  0) fTraction(3)=fTraction(3)+fK*u_n
c
c     approximate stiffness for time step
      if (u_n .lt.  0) then
        ek=fac*sigbyL+fK
      else
        ek=fac*sigbyL
      endif
c
      if (maketan) then
        dsave(1,1)=sigbyL/fd_c_t*(fd_c_n/fd_c_t)
        dsave(2,1)=0.
        dsave(3,1)=0.
        dsave(1,2)=0.
        dsave(2,2)=sigbyL/fd_c_t*(fd_c_n/fd_c_t)
        dsave(3,2)=0.
        dsave(1,3)=0.
        dsave(2,3)=0.
        dsave(3,3)=sigbyL/fd_c_n
        if (u_n.lt.0) dsave(3,3)=dsave(3,3)+fk
      endif
      return
      end
```

APPENDIX S: User Defined Boundary Flux

A user defined boundary flux interface is provided in LS-DYNA where it is possible to define the thermal heat flux (power per surface area) in or out of a surface segment as an arbitrary function of temperature and history. The user may associate history variables with each individual flux interface and also use load curves.

The user flux interface is invoked using the keyword *BOUNDARY_FLUX_*OPTION.* This is accomplished with the parameter NHISV. When it is defined with a value greater than 0, the user subroutine

```
subroutine usrflux(fl,flp,…)
```

is called to compute the flux (`fl)` defined as heat (energy) per time and per surface area.

Other parameters that are passed to the user flux subroutine include the segment nodal temperatures at the previous (T_0) and current time (T_1), the segment nodal coordinates and the time integration parameter α. Also, the current thermal simulation time t, the time step Δt and average segment temperature (T_α) at time $t+\alpha\Delta t$ is provided together with the curve array for accessing defined load curves in the keyword input file. For computing load curve values, note that load curve IDs need to be transformed to internal numbers or the subroutine crvval should be used, see the appendix on user defined materials for details.

The segment coordinates available in the subroutine are such that the outward normal vector follows the well-known right-hand rule, thus segments corresponding to the lower surface of thick thermal shells are reversed before passed to the subroutine. For shells in general, the segment connectivity should follow the connectivity of the actual shell element to avoid problems.

Optionally, the user may define the derivative of the flux `fl` with respect to the average segment temperature (T_α) at time $t+\alpha\Delta t$, `flp`. This value is used in the nonlinear thermal solver for assembling the correct stiffness matrix and must be set by the user. If possible, it is recommended to use a value that reflects the nonlinearity of the flux model, otherwise the value 0 should be used.

An array of history variables, identical with the input parameters defined in the keyword input file, are passed to the subroutine that can be updated with time or kept constant throughout the simulation. An example of usage would be to integrate the flux with time to keep track of the dissipated energy per surface area in order to simulate the effects of spray cooling in hot-stamping.

```
      subroutine usrflux(fl,flp,x,tnpl,tnl,nodes,
     .     alpha,atime,atemp,dt,time,fhsv,nfhsv,crv)
C**************************************************************************
C|  LIVERMORE SOFTWARE TECHNOLOGY CORPORATION  (LSTC)                     |
C|  ------------------------------------------------------------------    |
C|  COPYRIGHT © 2007 JOHN O.  HALLQUIST, LSTC                              |
C|  ALL RIGHTS RESERVED                                                   |
C**************************************************************************
c
c     User subroutine for boundary thermal flux
c
c     Purpose:  To define thermal flux parameter (heat per surface area and
c      time)
c
c     Variables:
c
c     fl              = flux intensity (output)
c     flp             = flux intensity derivative wrt atemp (output)
c     x(3,nodes)      = global segment coordinates (input)
c     tnpl(nodes)     = temperatures at time time (input)
c     tnl(nodes)      = temperatures at time time-dt (input)
c     nodes           = number of nodes in segment (3,4 or 6) (input)
c     alpha           = time integration parameter (input)
c     atime           = time+(alpha-1)*dt
c     atemp           = average segment temperature at time atime
c     dt              = time step size (input)
c     time            = time at which the new temperature is sought (input)
c     fhsv(nfhsv)     = flux history variables (input/output)
c     nfhsv           = number of flux history variables for this segment
c      (input)
c     crv             = curve array (input)
c
      include 'nlqparm'
      dimension x(3,*),tnpl(*),tnl(*)
      dimension fhsv(*),crv(lq1,2,*)
c
c     Define flux by linear convection
c     that optionally decays (in an ad-hoc way) as power
c     dissipates from surface
c
c     fhsv(1) = convection coefficient
c     fhsv(2) = ambient temperature
c     fhsv(3) = total amount of energy per surface area available
c     fhsv(4) = dissipated energy per surface area at current
c
      hcon=fhsv(1)
      tinf=fhsv(2)
      flin=hcon*(tinf-atemp)
      if (nfhsv.gt.2) then
         q=(1.-fhsv(4)/fhsv(3))/
     .        (1.+.5*dt*flin/fhsv(3))
         flp=-q*hcon
         if (q.gt.1.) then
            q=1.
            flp=-hcon
         elseif (q.lt.0.) then
            q=0.
            flp=0.
         endif
         fl=q*flin
         fhsv(4)=fhsv(4)+dt*.5*fl
         fhsv(4)=min(fhsv(3),fhsv(4))
      else
         fl=flin
         flp=-hcon
      endif
```

```
c
      return
      end
```

APPENDIX T: Metal Forming Glossary

A TYPICAL DRAW DIE ENGINEERING PROCESS

Clay models of a new vehicle are scanned and the outer shell surfaces are created in a design studio. Body-in-white engineers and designers are responsible to create all the inner parts and various structure and underbody parts. Flanges are created on outer surface parts to be joined with the inner panels. These parts are typically created in the car axis; with global X-axis runs from the front to the back along the car's center line, global Y-axis from the driver side to the passenger side and Z-axis going straight up.

During the clay design and shaping, simultaneous and multidisciplinary engineering may be practiced involving divisions/departments from design, engineering and manufacturing. Material suppliers may also be involved in this stage for consultation if advanced materials are to be applied. Multidisciplinary involvement in this early stage of a vehicle development allows manufacturing engineers to capture any part designs with "no make" conditions that would be costly if they were allowed to proceed to a later stage; while design envelopes can be pushed to their maximum potential within the state-of-art manufacturing capabilities.

During the advance feasibility phase, parts from the design studio are processed quickly to go through an engineering process involving mainly the process engineer and FEA simulation engineer. Addendum and binder are roughly built in order to conduct a reliable draw die simulation. The exact process eventually would be used to build the die may not yet be established, therefore similar processes from knowledge base are used as references. Rarely are secondary dies (all dies except draw die) simulated. The main task here is to provide some quick assessment of the part's manufacturability through fast design/engineering iterations.

During the hard die design and construction phase, stamping manufacturing processes (some in three dimensional) are established, by referring to existing knowledge base, with necessary modifications needed for the current part design, and with the limits of manufacturing equipment such as press type, shut height, maximum tonnage and automation, etc. The stamping process includes the number of dies (to make a complete part of the final shape), part tipping, draw height, trimming (direct or aerial), flanging, springback compensation requirements, etc. Not all areas of the part may be formed to its final shape in one draw die. Some involve redraw die if one area of the part is especially deep compared to the rest of the part, which may otherwise cause uncontrollable wrinkles, splits, or exceed the draw height limit without the redrawing. Some areas of the part (which may have "undercut" or "die locked" conditions) may be unfolded to the addendum, then trimmed and flanged in a flanging die later. A typical example of such can be found along the hood line of a fender outer. There may be multiple trimming and flanging dies, since not all areas of the drawn panel may be feasible for trimming or flanging all in one trim or flanging die.

Referring to Figure 63-1, a typical draw die development process flow for an outer part is illustrated. The part is tipped from the global car axis to a 'draw die' axis, which takes into account of balancing the internal draw angles over the entire part, and minimizing the overall draw height, etc. Hem flanges are unfolded off the part breakline (Figure 63-1) in one of the following ways:

1)Tangent extension of the part surface,
2)Horizontal surface,
3)Vertical down-standing surface,
4)Any angled surface between horizontal and vertical surface,
5)To the addendum surfaces to be designed; in this case, the unfolding will happen after the addendum design is complete.

Flange unfold must take into account the trimming condition later in the trim dies. Direct trim represents the trim steel going down in a draw direction (vertical); while aerial (cam) trim steel covers all other directions, driven by a 'cam driver' driven further by the vertical downward motion of the trim die. There are specific trim angle requirements for direct and aerial trim operations. Next, the part boundary is smoothed, filling up any gaps, holes and sharp features. The boundary will likely be modified later during the addendum build. Binder is created, based on the unfolded part boundary and overall part curvature. A developable binder surface, which consists of planes, cylindrical surfaces or a combination of both, is preferred; however, in some cases a doubly curved binder surface (undevelopable surface) must be designed for the purpose of reducing the draw depth at critical locations for material utilization, alleviating thinning, or both. Theoretical punch opening line (P.O. line, Detail #2 in Figure 63-1) can be offset from the smoothed part boundary in the plane normal to the draw axis, projected to the binder surface in the draw direction; some smoothing of the P.O. lines may be required. Generally, the finished P.O. lines should be consisted of mostly straight lines and radii, with generous transition among corners. P.O. lines do not follow tight corners, avoiding formation of wrinkles during the draw. In addition, design of P.O. lines must take into consideration the material utilization issue, especially in the blank sizing critical locations. Once P.O. lines design is complete, binder surfaces inside of the P.O. are trimmed and the remaining binder surface design is sent for binder closing simulation. Given a blank initial shape, simulation of the binder closing action can determine the quality of the binder design. Typically, for exterior outer panels, no buckles or wrinkles are allowed during the closing; for inner panels, some wrinkles are acceptable, or even desirable, depending on the part shape. Lower punch/post support may be introduced to reduce the draw height, alleviate thinning, or remove the initial blank drape into the binder cavity. Based on the binder closing simulation results, unacceptable binder design is sent back for rework, and the satisfactory binder design proceeds into the next step of addendum build, which is used to fill the space between the part boundary and binder surface. It is noted that the P.O. lines may need to be adjusted during the addendum design. The trimming condition may also be affected and modified during the addendum design. Next, draw simulation ensues, assessing the overall formablity of the draw design. If quality targets are not achieved in the simulation, the process may go all the way back to the tipping for redesign/reprocess; otherwise successful simulation will direct the draw surface design to the next stage of draw die structure design.

TYPES OF DRAW DIES

There are many different types of draw dies used to punch the part to their intended shape, as shown in Figure 63-2. They basically can be divided into either single action or double action dies. Specifically (names may vary among different companies),

1)Air draw ¯Single action. It is a 3-piece die system with 1 piece upper (cavity) and 2-piece (binder and punch/post) lowers. The upper cavity (driven by press ram) moves down in one action to close with the lower binder, and then closes with the lower punch to draw the part to the home position. The lower binder is either sitting on an air cushion through pins that go through the press bed, or directly on nitrogen cylinders arranged uniformly between the bottom of the binder structure and the press bed; the punch is fixed onto the press bed. This is the most popular draw type, mainly because of its speed and efficiency. Its limitation includes a maximum draw height of 10”.

2)Toggle draw ¯Double action. It is a 3-piece die system with 2-piece upper (binder and punch) and 1 piece (cavity) lower. Upper binder, driven by the outer ram of the press, moves down to clamp the blank with the lower cavity; then the upper punch, driven down by the inner ram, closes with the lower cavity to complete the draw. Since this adds another action, it is slower than the air draw; however, this type of draw die is well suited to control the wrinkles created during the forming of difficult part, such as liftgate and door inner. Furthermore, this type of draw has a relatively large draw height, which is limited only by the press.

3)Air draw with pressure pad ¯Single action. This is very similar to 1), except an additional pressure pad, driven by nitrogen cylinders mounted on the upper die structure, closes first with the lower punch, then the entire upper comes down to finish the draw. This is similar to 1) in efficiency.

4)Stretch Draw (four piece) ¯Double action. It is a 4-piece die system with 2-piece uppers (upper binder and punch) and 2-piece lowers (lower binder and cavity). Upper binder moves down to close with lower binder, moving together for a certain distance (up to 2”), then upper punch comes down to completely close with lower cavity. Finally the binders move down together to their home position. This process is not used as often as 1), 2) and 3), however, it is very capable in forming difficult inner parts, especially those prone for wrinkles, such as liftgate inner, door inner and floor pan. Since there is a ‘pre-stretch’ action with the binders clamping the blank and moving down together, strain path change in the part during the forming is expected. This is the slowest draw type.

5)Crash Form Die ¯Single action. It is a 2-piece die system with no upper or binders. Upper and lower tool takes the same shape, with upper moving down to close with the lower tool. This is obviously a very simple die, which can handle simple parts, with not too much draw depth variation (near constant draw depth all around).

TYPES OF FLANGING DIES

There are three types of flanging dies, as shown in Figure 63-3. All three types have a fixed lower (trim) post upon which the drawn (or partially trimmed part) is sitting, and a pressure pad (or multiple pads) which holds the part (which is loaded onto the post in a vertical direction) in place against the post. In a direct flanging process, the flanging steel (Figure 63-3) moves vertically down to 'wipe' or 'bend' the part to its flanged position. In an aerial flanging process, the flanging steel moves to form the flanges in an angle rather than the vertical direction. The steel is held by the cam slide, driven by a cam driver which in turn is driven by the trim die's downward movement. Since the finished flange forms a 'die lock' condition, meaning the flanged part will not be able to be lifted (retract) out of the trim die into the next die (or station), the filler cam (Figure) is moved horizontally out of the way so the part can be lifted up and out. Once the part is removed, the filler cam moves back into its home position ready for the next drawn panel to be loaded. In the rotary cam flanging process, the filler cam is called a 'rotor', which rotates out of the way for the flanged part to be lifted up and out. Some parts of the rotary cam flanging design are a patented process. In comparison to the conventional cam flanging, rotary flanging has the advantage of being very compact.

TYPES OF HEMMING DIES

There are two types of hemming dies, as shown in Figure 63-4. Both processes have a fixed hemming bed, on which the flanged outer part is loaded; the inner panel is then loaded onto the outer panel; proper clamps are applied to hold tight the panels together and in place. In press (or table top) hemming, a pre-hemming tool is moved to form the flange into a halfway position, followed by the final hemming tool pressing down on the flange against the inner and outer panels to its final position. Many different shapes of hem tips can be achieved, as shown in the figure. In roller hemming, a pre-roller moves in a three dimensional curvilinear path following the hem tip, to form the flange partially. This is followed by a final roller, moving in another three dimensional path, to finish the hem shape. In the hemming of complex or high-end parts, many passes (rollers) are needed to achieve high quality hem surfaces. Similarly, with the design of different roller shapes, many shapes of hem tips can be achieved.

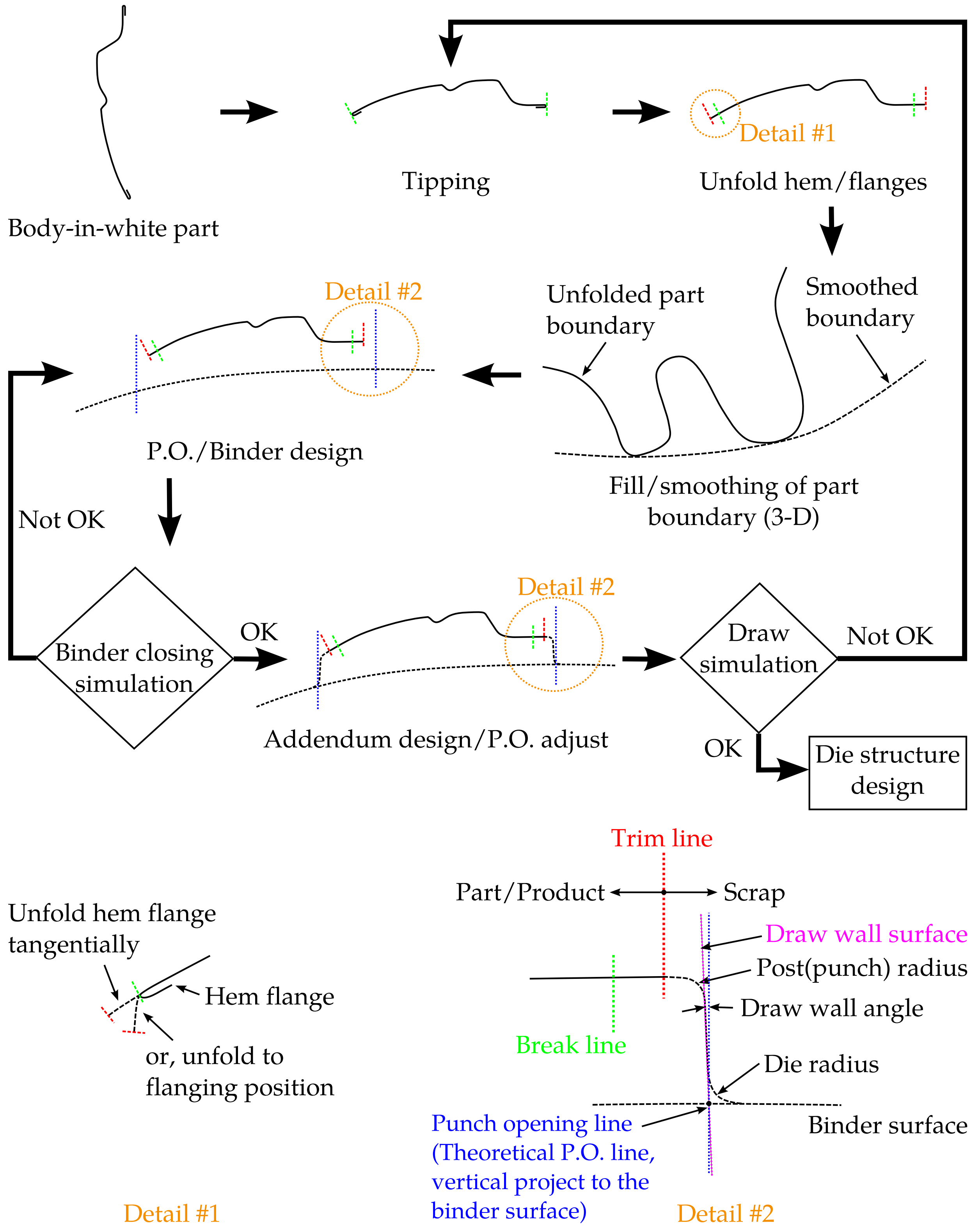
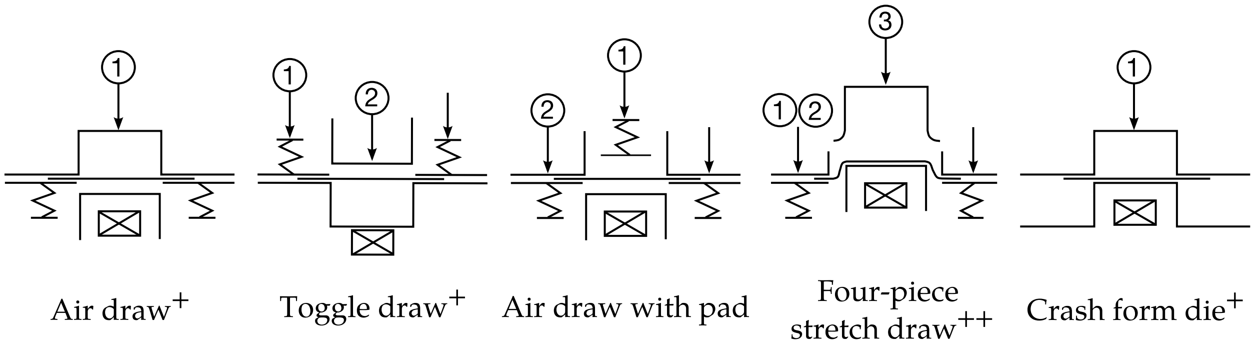
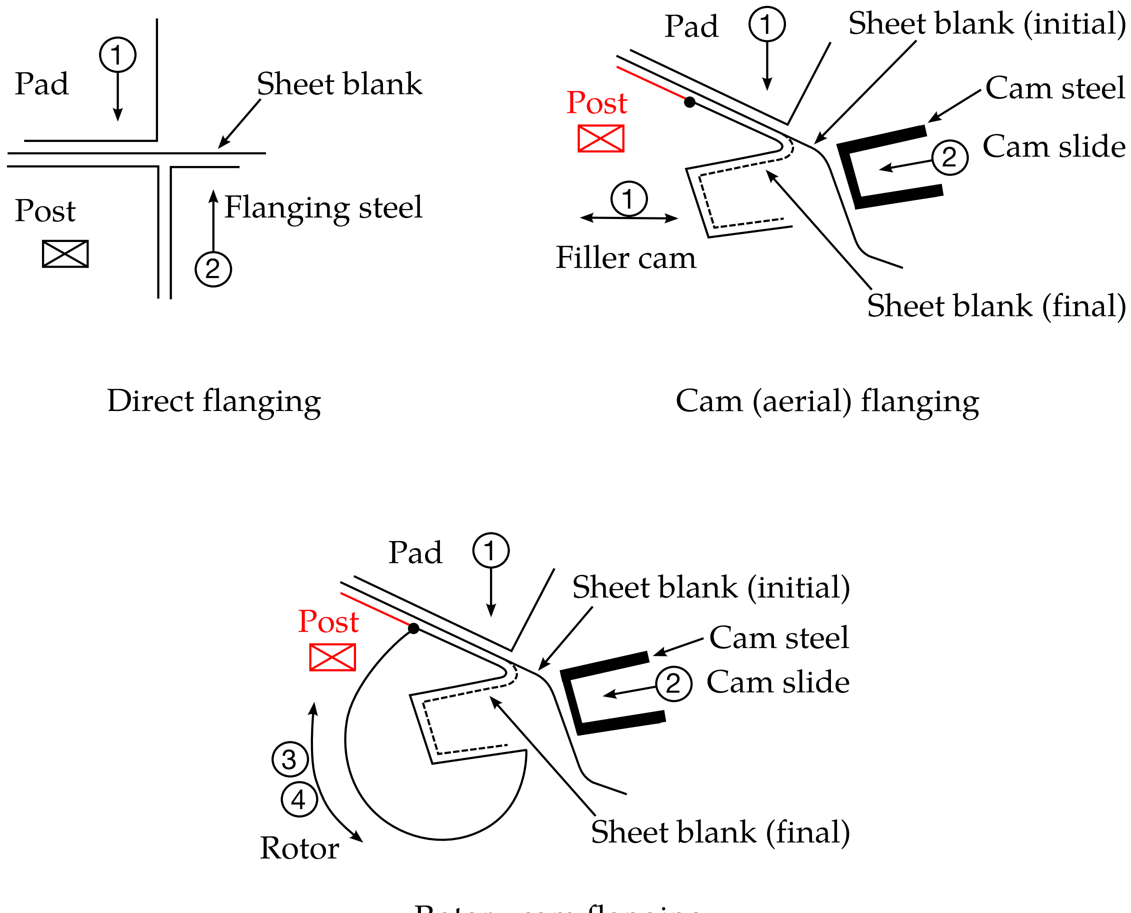


Figure 63-1. A typical draw die engineering process



+ : single action
++ : double action

Figure 63-2. Types of draw dies

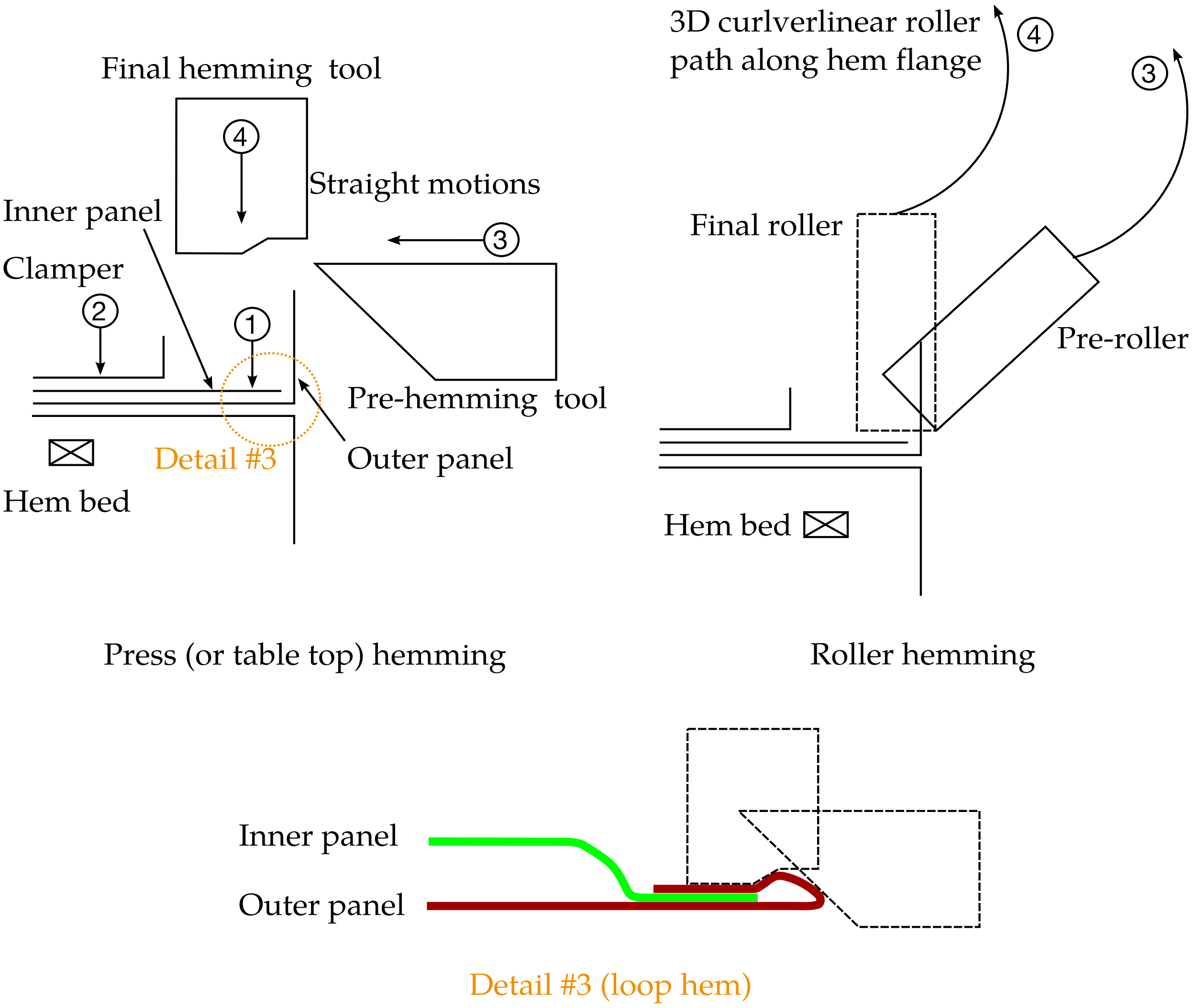


Direct flanging

Cam (aerial) flanging

Rotary cam flanging

Figure 63-3. Types of flanging dies



Press (or table top) hemming

Roller hemming

Detail #3 (loop hem)

Figure 63-4. Types of hemming dies