Implizite Berechnungen mit LS-DYNA

- Einführung und Anwendungsbeispiele -

Tobias Erhart, 30. Juni 2005

DYNA*more* GmbH
Industriestr. 2, D-70565 Stuttgart
Tel. 07 11 - 45 96 00 - 18   Fax 07 11 - 45 96 00 - 29
e-mail: tobias.erhart@dynamore.de
Internet: www.dynamore.de

Motivation: Why Implicit?

- **prestressed**, **quasi statically loaded** structures
- **long duration** analysis > 200 ms
- **different time scales** in process
  e.g. static loading followed by transient loading
  or transient loading followed by static loading
- **applications**
  e.g. metalforming, roof crush, door sag, dummy seating, ...

➡ **LS-DYNA provides explicit and implicit solution schemes**

- **one data structure**
- **one input / output**

Explicit vs. Implicit (Dynamics)

$$\rho\,\mathbf{u}_{,tt} = \nabla \cdot \boldsymbol{\sigma} + \mathbf{f}$$

explicit

$$\mathbf{M}\mathbf{a}_n = \mathbf{f}_n^{ext} - \mathbf{f}_n^{\text{int}}$$

- solution: directly
- decoupled: efficient, fast
- many small time steps
- conditionally stable (Courant)
- equilibrium?

short time dynamics:
high frequency response,
wave propagation

➡ impact, crash, ...

implicit

$$\mathbf{M}\Delta\mathbf{a}_{n+1} + \mathbf{K}\Delta\mathbf{u}_{n+1} = \mathbf{f}_{n+1}^{ext} - \mathbf{f}_n^{\text{int}} - \mathbf{M}\mathbf{a}_n$$

- solution: iteratively
- linearization necessary
- few large time/load steps
- unconditionally stable
- equilibrium! convergence?

structural dynamics:
low frequency response,
vibration, oscillation

➡ earthquake, machines, ...

Types of Implicit Analyses

Linear Analysis

- static or dynamic
- single, multi-step

Eigenvalue Analysis

- frequencies and mode shapes
- linear buckling loads and modes
- modal analysis: extraction and superposition

Nonlinear Analysis

- Newton, Quasi-Newton, Arclength solution
- static or dynamic
- default LS-DYNA: static and nonlinear!

Simplified Implicit Flowchart: Terms

0. Initialisierung $\mathbf{u}^0, \dot{\mathbf{u}}^0, \ddot{\mathbf{u}}^0$
1. Zeitschleife $(t^{n+1} = t^n + \Delta t)$

   Bestimme ‘Lasten’ (Dirichletwerte, Lasten, ...) für t^{n+1}

   Initialisiere Iterationszähler $i = 0$

   2. Setze Prädiktor-Größen $\mathbf{u}_0^{n+1} = \mathbf{u}^n; \dot{\mathbf{u}}_0^{n+1} = \dot{\mathbf{u}}^n; \ddot{\mathbf{u}}_0^{n+1} = \ddot{\mathbf{u}}^n$
   3. Berechne iterationsunabhängige RHS-Anteile
   4. Iterationsschleife
      5. Berechne und assembliere die effektive RHS $\hat{\mathbf{R}}$
      6. Berechne und assembliere die effektive Steifigkeitsmatrix $\hat{\mathbf{K}}$
      7. Löse das Gesamtgleichungssystem $\Delta\mathbf{u} = \hat{\mathbf{K}}^{-1}\hat{\mathbf{R}}$ ← *linear problem*
      8. Aktualisiere die Verschiebungen, Geschwindigkeiten und Beschleunigungen
   9. Konvergenzcheck: if (*Residuum*>TOL) goto 4. else goto 10.
10. $n = n + 1$ goto 1.

nonlinear problem

Activating Implicit Analysis

Use ***CONTROL_IMPLICIT_GENERAL** to activate implicit

- specify time step size
- all other *CONTROL_IMPLICIT keywords are optional
- default is nonlinear, static analysis

Use a **double precision executable** for implicit analysis

- better convergence for nonlinear
- mandatory for linear, eigenvalue accuracy

Stiffness Matrix requires lots of **memory**

- huge speed penalty for out-of-core jobs

ls-dyna i=input.k memory=200m

200,000,000 words:
800 Mbytes in single precision
1600 Mbytes in double precision

Most keywords apply to explicit and implicit

- *NODE, *ELEMENT, *SECTION, *MAT, ...
- easy to run a model with either method, but: carefully inspect input deck

Activating Implicit Analysis

Three types of analyses can be performed

- fully explicit (default)
- fully implicit
- switching: explicit - implicit, implicit - explicit (prescribed or automatic)

All keywords for implicit

*CONTROL_IMPLICIT_GENERAL	*CONTROL_IMPLICIT_SOLVER
*CONTROL_IMPLICIT_SOLUTION	*CONTROL_IMPLICIT_AUTO
*CONTROL_IMPLICIT_STABILIZATION	*CONTROL_IMPLICIT_DYNAMICS
*CONTROL_IMPLICIT_MODES	*CONTROL_IMPLICIT_EIGENVALUE
*CONTROL_IMPLICIT_BUCKLE	

Proper selection of LS-DYNA features

- not all features are available in implicit mode
- warning & error messages, feature substitution

Implicit Keywords

*CONTROL_IMPLICIT_GENERAL (required for implicit)

- activates implicit mode, explicit-implicit switching
- defines implicit time step size (standard LS-DYNA termination time is used too)

*CONTROL_IMPLICIT_SOLVER (optional)

- parameters for linear equation solver, which inverts stiffness matrix: [K]{x}={f}

*CONTROL_IMPLICIT_SOLUTION (optional)

- parameters for nonlinear equation solver (Newton-based methods)
- controls iterative equilibrium search, convergence
- "linear" analysis selected here (a special case where no iterations are performed)

*CONTROL_IMPLICIT_AUTO (optional)

- activates automatic time step control
- default is fixed time step size, error termination if any steps fail to converge

Implicit Keywords

*CONTROL_IMPLICIT_DYNAMICS (optional)

$$\mathbf{M}\Delta\mathbf{a}_{n+1} + \mathbf{K}\Delta\mathbf{u}_{n+1} = \mathbf{f}_{n+1}^{ext} - \mathbf{f}_{n}^{\text{int}} - \mathbf{M}\mathbf{a}_{n}$$

- include inertia terms
- problem “time” must now be real, physical time
- can improve convergence, especially when rigid body modes are present

*CONTROL_IMPLICIT_EIGENVALUE (optional)

- signals LS-DYNA to perform eigenvalue analysis, then stop
- number of eigenvalues/vectors, optional frequency shift
- great for debugging/model checking

*CONTROL_IMPLICIT_STABILIZATION (optional)

- Allows multi-step springback

➡ User‘s manual contains helpful notes on each input parameter, **Appendix M** gives survey of LS-DYNA/implicit features

Linear Static Analysis $\mathbf{Ku = R}$

Activate the implicit method

- ***CONTROL_IMPLICIT_GENERAL**: imflag = 1

Select a stepsize and termination time

- for static analysis, choice of time is arbitrary
- ***CONTROL_IMPLICIT_GENERAL**: dt0 = 1.0
- ***CONTROL_TERMINATION**: term = 1.0

nstep = term/dt0 = 1

Select linear solution method (no equilibrium iterations)

- ***CONTROL_IMPLICIT_SOLUTION**: nsolvr = 1

Select a linear element type

- shell # 18, 20, 21
- brick # 18

Use a double precision LS-DYNA executable

Element Formulations for Linear Analyses

Linear and nonlinear element formulations are different

- linear: integrate stress over undeformed geometry
- infinitesimal deformation eliminates some locking problems
- enhanced strain fields accurately represent linear elasticity

Brick Elements

- type 18: linear solid

$$\mathbf{f} = \int_{\Omega_0} \mathbf{B}^T \boldsymbol{\sigma} \, d\Omega$$

Shell Elements

- type 18: linear thin shell (Kirchoff)
- type 20: linear thick shell (Mindlin)
- type 21: linear enhanced shell (CQUAD4)

Boundary Constraints

Boundary conditions and rigid body modes

- static implicit simulation requires boundary constraints
- rigid body modes must be eliminated
  (otherwise stiffness matrix is singular / not invertible)
- apply translational constraints to three nodes

a - reference node, dx=dy=dz=0

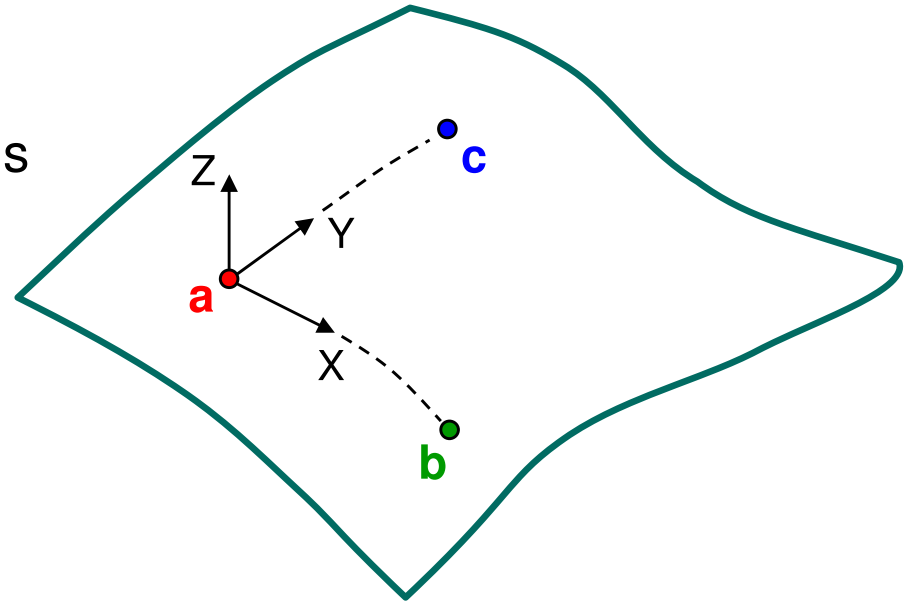
→ eliminates all translational modes

b - node along X-axis, dy=dz=0

→ eliminates rotations about y- and z-axis

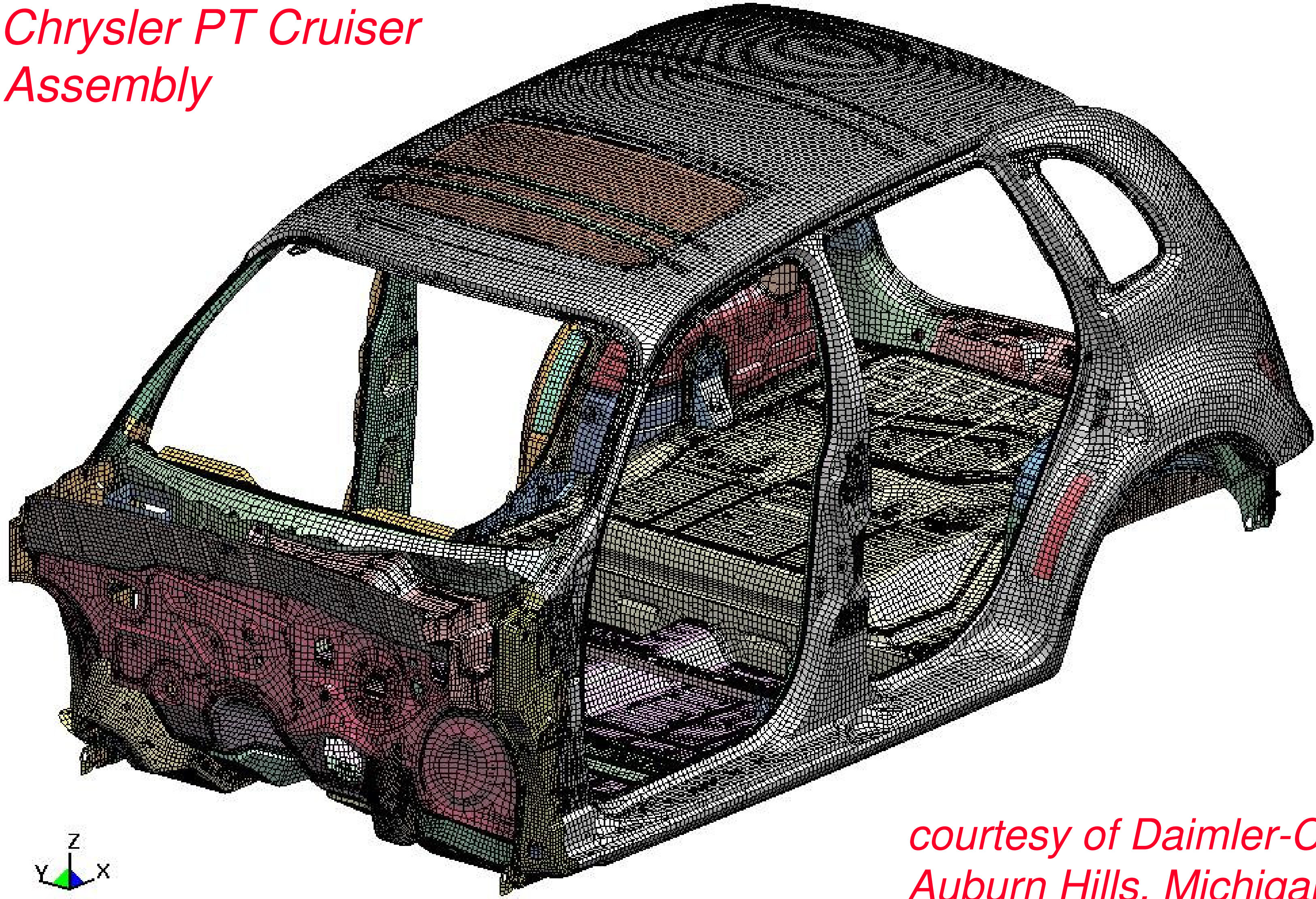
c - node along Y-axis, dz=0

→ eliminates rotation about x-axis



Example: Linear Static Analysis

2000 Chrysler PT Cruiser
Body Assembly

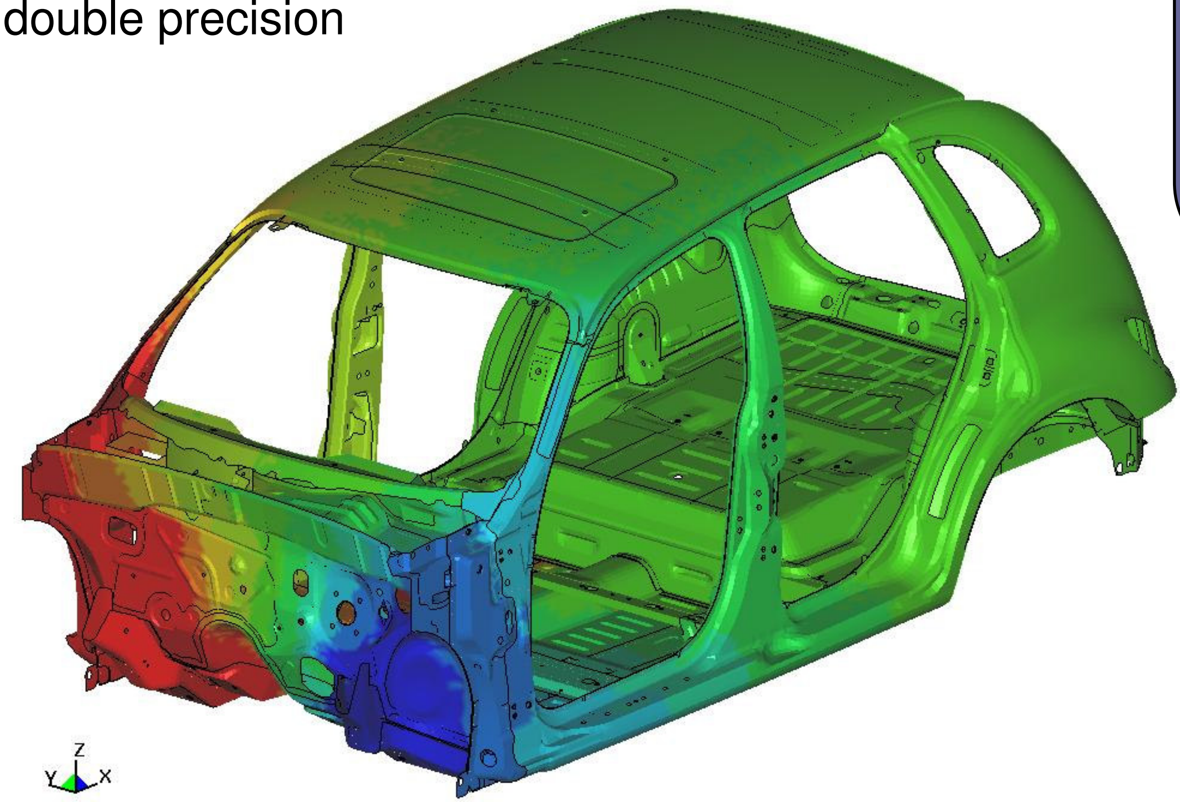


courtesy of Daimler-Chrysler
Auburn Hills, Michigan, USA

Example: Linear Static Analysis

Static Linear Torsion Analysis

- NSOLVR=1
- shell type 21
- double precision



model details

240,000 nodes
3,300 type 100 spotwelds
1,450,000 equations
memory=740m (~6 Gbyte)

SMP parallel performance

1 cpu: 753 sec
4 cpu: 485 sec

Attention: larger models demand 64-bit O/S to exceed 2 Gbyte memory limit

DYNA MORE

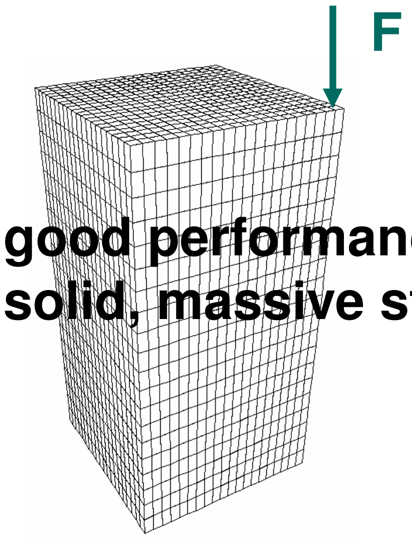
Linear Equation Solver

During each nonlinear iteration, the linear system $\hat{\mathbf{K}}\Delta\mathbf{u} = \hat{\mathbf{R}}$ is solved.

Direct Methods

LS-DYNA default

- gaussian elimination
- inexpensive backsolve (quasi-Newton)
- The sparse direct solver
- costly: CPU and memory
- robust and reliable



Iterative Methods

good performance for solid, massive structures

- iteration: improve approximate solution
- potentially low operation count
- convergence difficult for some problems
- promising future developments

Keyword ***CONTROL_IMPLICIT_SOLVER**

Direct Solvers

A sparse, direct linear equation solver is used by default (LSOLVR=4)

- serial or SMP parallel execution
- automatic out-of-core mode if insufficient memory available for incore
- double precision version also available (LSOLVR=5)
  - improved convergence for a few models
  - 2x memory penalty
  - better to use a double precision version of LS-DYNA
- BCSLIB-EXT solver from Boeing also available (LSOLVR=6)
  - double precision only
  - best for very large models (excellent out-of-core performance)

All sparse direct solvers execute in three phases

- symbolic factorization
- numeric factorization
- forward elimination / back substitution

Iterative Solvers

LS-DYNA offers six iterative linear equation solvers

- LSOLVR = 10: “best” iterative solver (currently activates #16)
  LSOLVR = 11: Conjugate Gradient method
  LSOLVR = 12: CG with Jacobi preconditioning
  LSOLVR = 13: CG with Incomplete Choleski preconditioning
  LSOLVR = 14: Lanczos method
  LSOLVR = 15: Lanczos with Jacobi preconditioning
  LSOLVR = 16: Lanczos with Incomplete Choleski preconditioning

All iterative solvers use the sparse matrix storage scheme

- eliminates all zero entries inside bandwidth
- minimizes total storage requirement
- Boeing Harwell format for portability

Dynamic Implicit Analysis

$$\mathbf{M}\Delta\mathbf{a}_{n+1} + \mathbf{K}\Delta\mathbf{u}_{n+1} = \mathbf{f}_{n+1}^{ext} - \mathbf{f}_{n}^{\text{int}} - \mathbf{M}\mathbf{a}_{n}$$

Newmark Method relates displacement, velocity, acceleration

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \mathbf{v}_n \Delta t + \left[\left(\tfrac{1}{2} - \beta\right)\mathbf{a}_n + \beta\,\mathbf{a}_{n+1}\right]\Delta t^2$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \left[(1-\gamma)\mathbf{a}_n + \gamma\,\mathbf{a}_{n+1}\right]\Delta t$$

$\beta = 0, \ \gamma = 1/2$	: explicit central difference method
$\beta = 1/4, \ \gamma = 1/2$	: implicit undamped trapezoidal rule
$\gamma > 1/2$	: numerical damping

LS-DYNA default

- convergence may be possible with large DT
- small DT may be needed to resolve high frequency response
- stabilizing effect to nonlinear equilibrium iterations
- rigid body modes OK! (mass terms eliminate singularity)

$$\hat{\mathbf{K}} = \mathbf{K} + \left(\frac{1}{\beta\,\Delta t^2}\right)\mathbf{M}$$

DYNA MORE

Activating Dynamic Implicit Analysis

Activating Dynamic Analysis

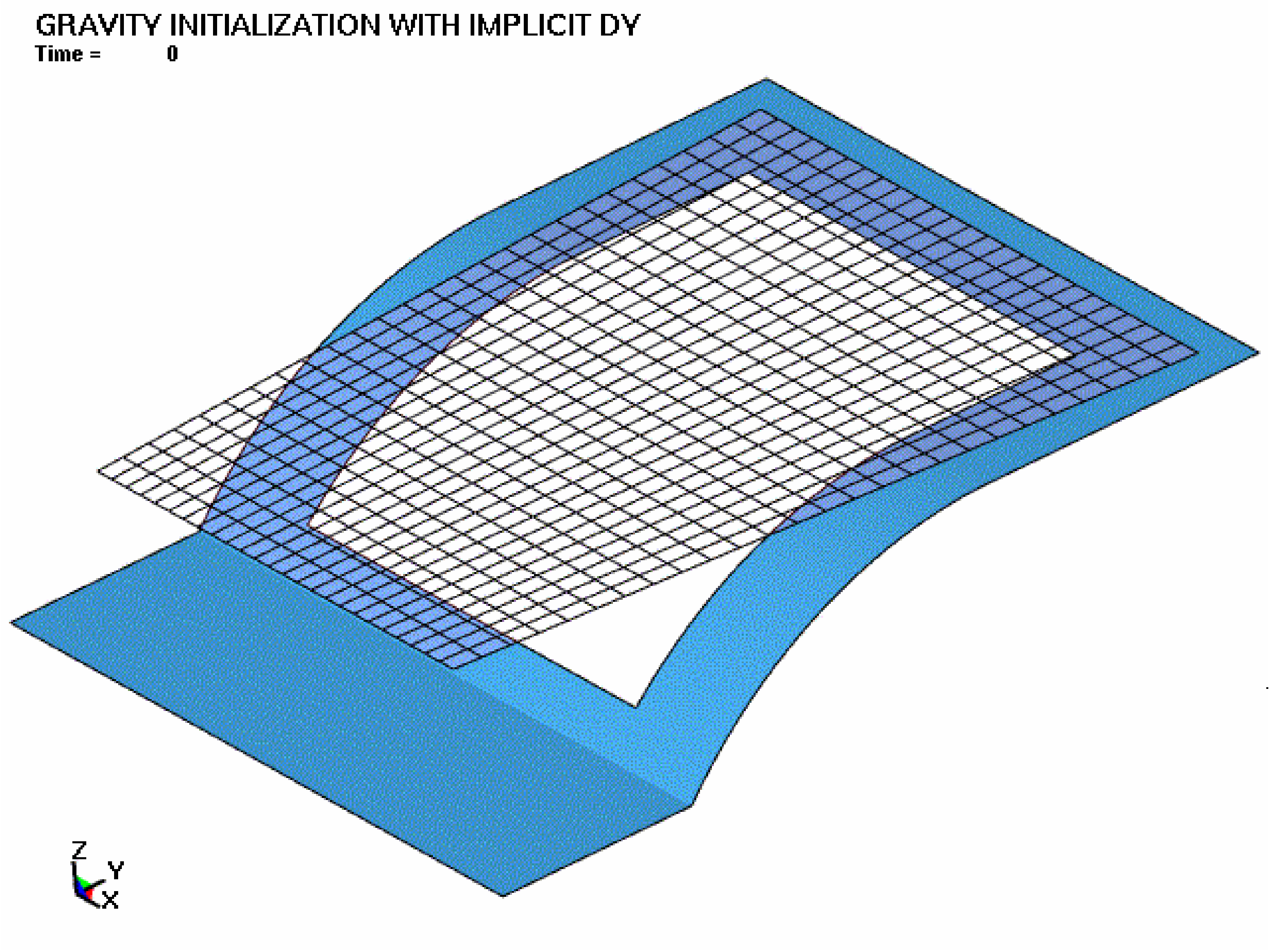
```
*CONTROL_IMPLICIT_DYNAMIC
$     imass      gamma       beta
          1          0          0
```

Implicit Dynamic Analysis may be linear or nonlinear

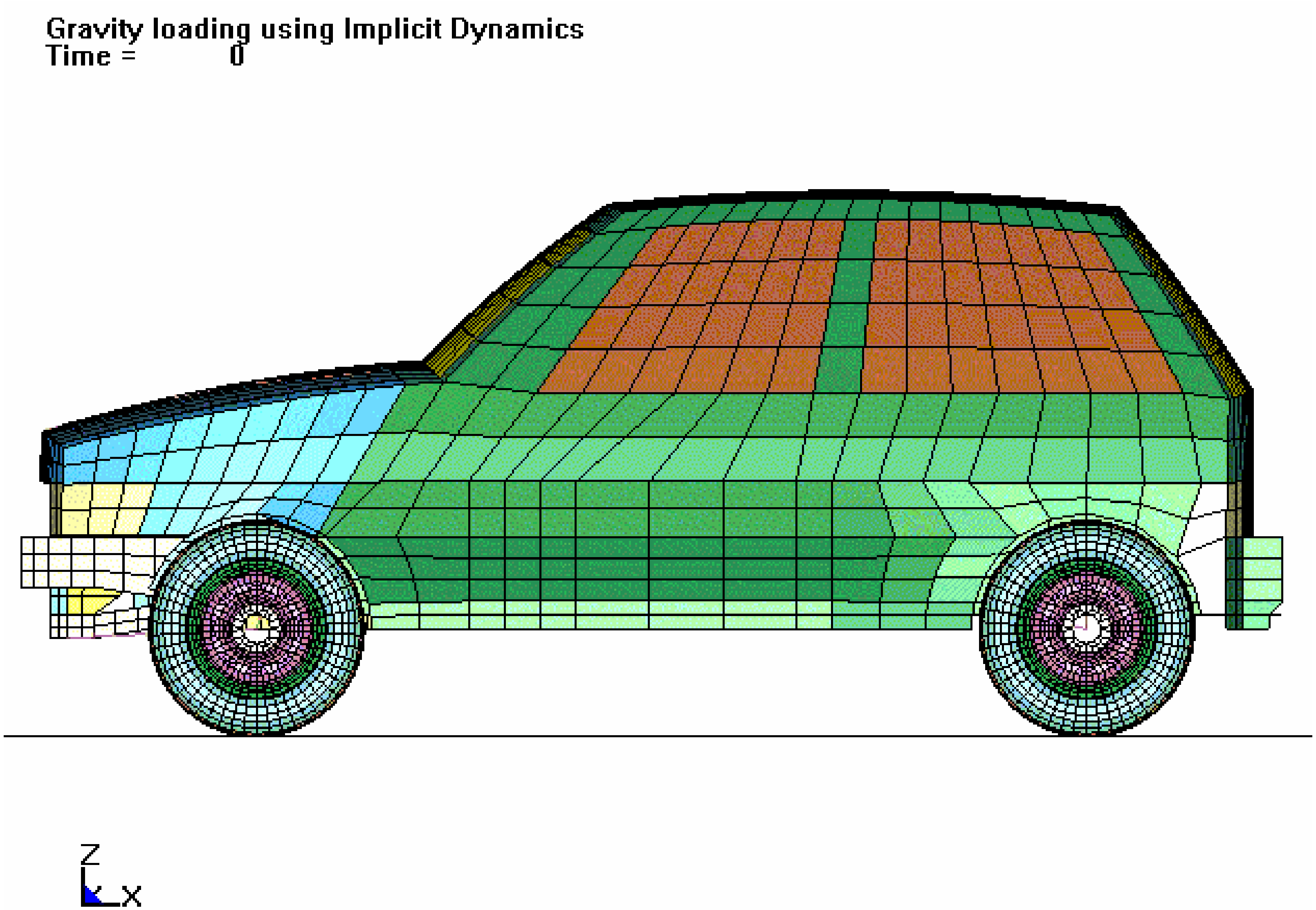
- inertia terms are simply added to stiffness matrix and residual vector
- very efficient when only one stiffness matrix factorization is performed
  - earthquake response analysis: long periods of nearly linear behavior
  - same stiffness matrix used for many nonlinear steps
- if time step size changes, a new stiffness will automatically be formed

$$\hat{\mathbf{K}} = \mathbf{K} + \left(\frac{1}{\beta \, \Delta t^2} \right) \mathbf{M}$$

Example: Sheet Metal Gravity Loading



Example: Car gravity loading



Eigenvalue Analysis

Compute Natural Frequencies and Mode Shapes

- linear analysis
- infinitesimal deformation (magnified for display)

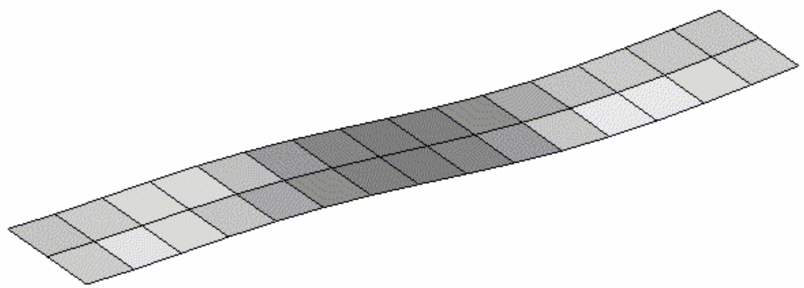
$$(\mathbf{K} - \lambda \mathbf{M})\,\boldsymbol{\varphi} = 0$$

Accuracy Requires Special Considerations

- linear elements (type 18, 20)
- double precision executable

Applications

- frequency analysis
- model integrity check
- extract modes for modal analysis



Activating Eigenvalue Analysis

Required Input Parameters

- non-zero termination time

  ***CONTROL_TERMINATION** `term=1.0`

- implicit analysis

  ***CONTROL_IMPLICIT_GENERAL** `isolvr=1, dt0=1.0`

- number of eigenvalues

  ***CONTROL_IMPLICIT_EIGENVALUE** `neigv=30`

Eigenvalue analysis with an existing implicit input deck

- just add one keyword, one input parameter:

```
*CONTROL_IMPLICIT_EIGENVALUE
$     neigv     center
         30        0.0
```

- LS-DYNA computes 30 lowest modes, terminates

Eigenvalue Input / Output

Input Options

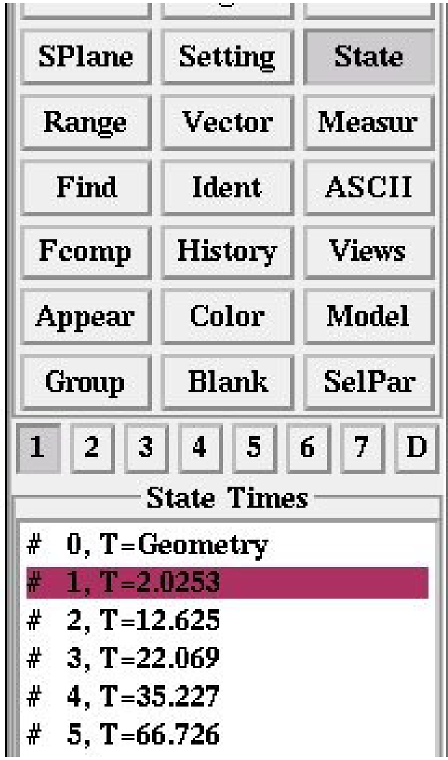
- number of eigenvalues/modes
- center frequency, frequency range
- eigenvalue extraction method: Lanczos eigensolver (default)

New output databases

- **d3eigv**: - binary plot database similar to d3plot
  - each state shows one mode shape
  - “State times” give circular frequencies f →
- **eigout**: - ASCII text file
  - summary of frequencies found

λ $\quad\quad$ $\omega = \sqrt{\lambda}$ $\quad\quad$ $f = \omega / 2\pi$ $\quad\quad$ $T = 1/f$

MODE	EIGENVALUE	RADIANS	CYCLES	PERIOD
1	1.619382E+02	1.272549E+01	2.025325E+00	4.937478E-01
2	6.292547E+03	7.932558E+01	1.262506E+01	7.920756E-02
3	1.922690E+04	1.386611E+02	2.206860E+01	4.531325E-02



Mode Extraction

*CONTROL_IMPLICIT_MODES

Modal Analysis

- approximate structural deformation using a set of modes
- modal amplitudes become the unknowns
- greatly reduced problem size
- superposition principle assumes linearity

Flexible Rigid Bodies

- large rigid body motion + superimposed modal deformation
- apply to a subset of parts, treat others as fully nonlinear

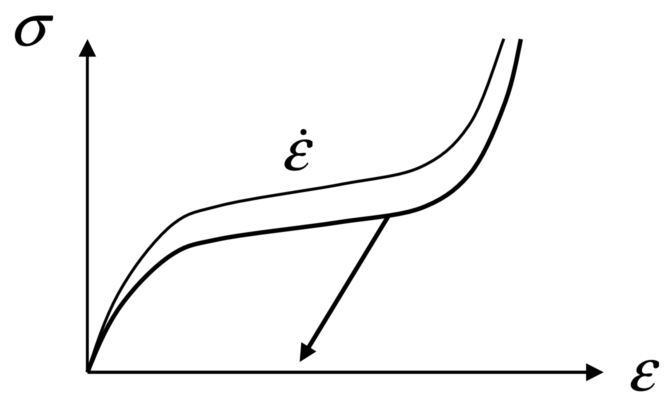
Analysis Procedure

1. compute modes for subset of parts (d3eigv, d3mode in v970)
2. rigidize, merge these parts
3. define *PART_MODES for master part
4. perform explicit transient dynamic analysis

Nonlinear Implicit Analysis

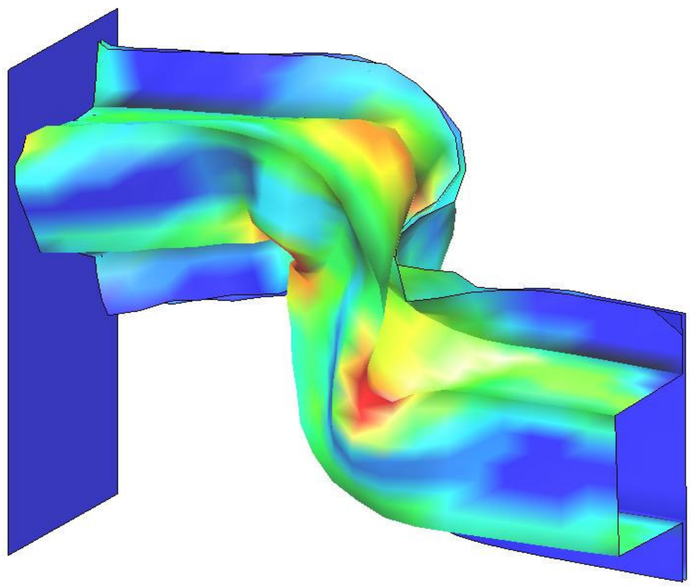
Material Nonlinearity

- plasticity, damage, failure
- rate dependence
- slope of stress-strain curve gives stiffness, should usually be monotonic



Geometric Nonlinearity

- large displacement, large rotation



Contact Nonlinearity

- normal force is sharply discontinuous
- frictional effects elastic-perfectly-plastic

Nonlinear Implicit Analysis

Implicit governing equations contain two problems to solve

$$\mathbf{M}\Delta\mathbf{a}_{n+1} + \mathbf{K}\Delta\mathbf{u}_{n+1} = \mathbf{f}^{ext}_{n+1} - \mathbf{f}^{\text{int}}_{n} - \mathbf{M}\mathbf{a}_{n}$$

Nonlinear Equilibrium Problem: *CONTROL_IMPLICIT_SOLUTION

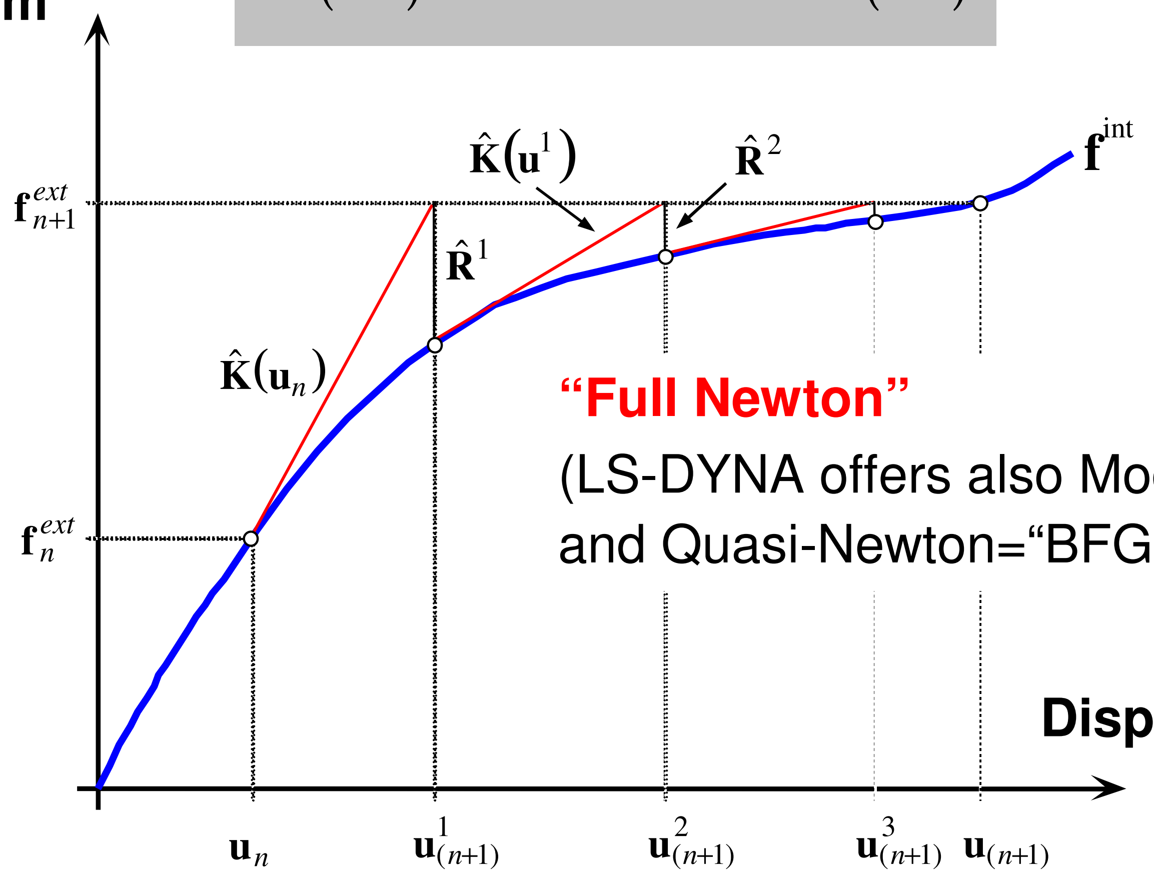
- find displacements **u** which satisfy equilibrium **f**ext=**f**int
- both **K**, **f**ext and **f**int can be nonlinear functions of ***u***
- iterative search employed using Newton-based method
- interactive switch "<ctrl-c> nlprint" toggles diagnostic output

Linear Algebra Problem: *CONTROL_IMPLICIT_SOLVER

- solve system of linear algebraic equations
- must solve during every nonlinear iteration
- great CPU and memory cost
- interactive switch "<ctrl-c> lprint" toggles diagnostic output

Nonlinear Equation Solver - Newton Method

$$\hat{\mathbf{K}}\left(\mathbf{u}^{(i)}\right)\Delta\mathbf{u} = \hat{\mathbf{R}}^{(i)} = \mathbf{f}^{ext} - \mathbf{f}^{\text{int}}\left(\mathbf{u}^{(i)}\right)$$



"Full Newton"

(LS-DYNA offers also Modified Newton and Quasi-Newton="BFGS")

Equilibrium is reached when iterations converge: $\left\|\Delta\mathbf{u}\right\| \to 0\,,\ \left\|\hat{\mathbf{R}}^{(i)}\right\| \to 0$

Input Parameters for Nonlinear Solver

NSOLVR: nonlinear solution method

- =1: linear approximation (no equilibrium iterations)
- =2: BFGS quasi-Newton method (DEFAULT)

ILIMIT: equilibrium iteration limit before re-evaluating $\hat{\mathbf{K}}$

- =1: new $\hat{\mathbf{K}}$ each iteration (“**Full Newton**" method)
- =11: use cheap BFGS update for 11 iterations, reform if not yet converged

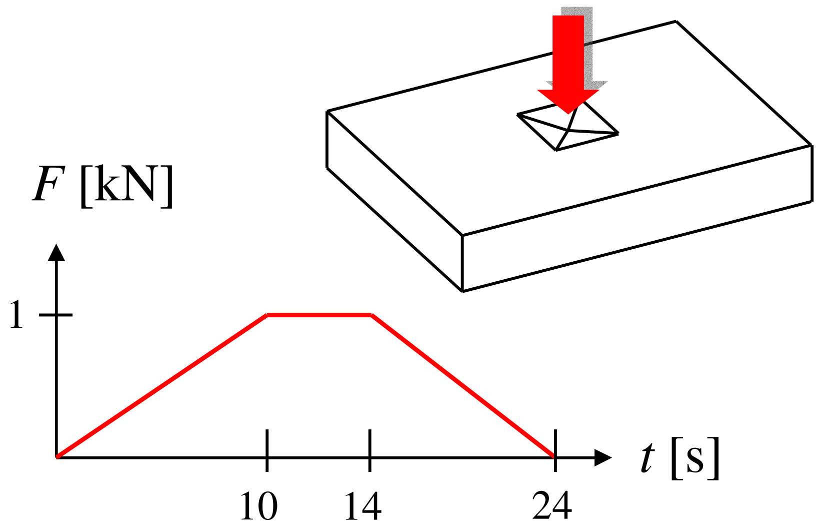
MAXREF: maximum reformation count before abandoning step

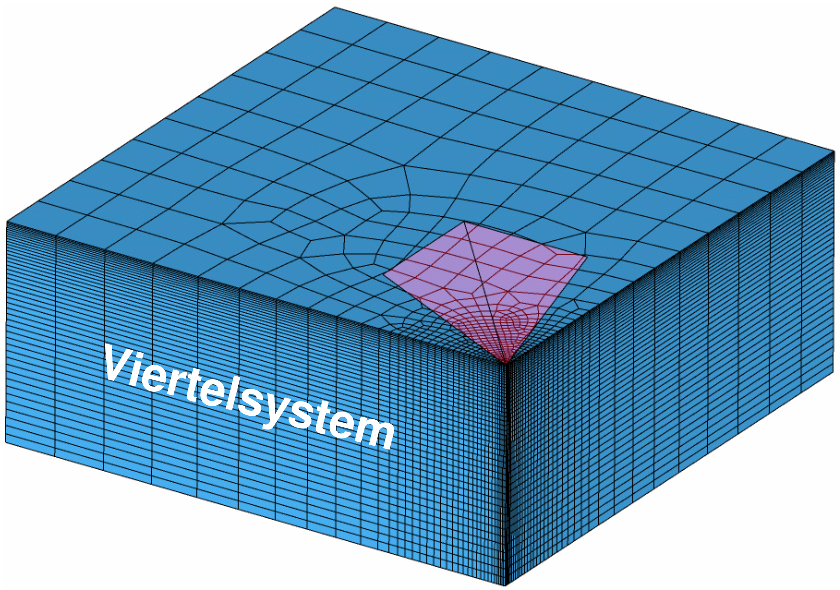
- if AUTO is active, dt will be reduced and step will be re-tried, so MAXREF can be smaller (~5)
- if AUTO not active, error termination occurs when MAXREF is reached so MAXREF should be larger (~15, default)

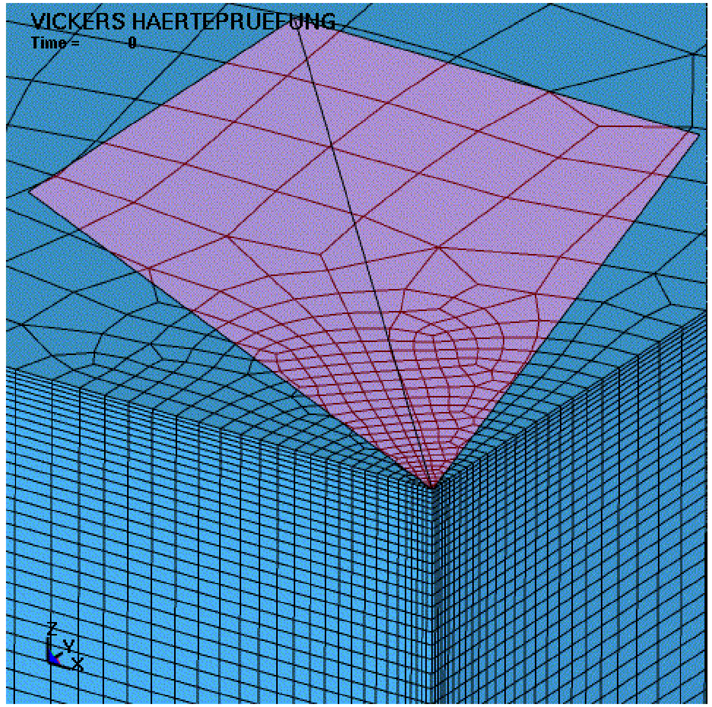
DCTOL, ECTOL: convergence tolerances

- use NLPRINT=1 or "<ctrl-c> nlprint" to monitor progress of iterations

Example: Vickers Hardness Test







implicit

1 cpu: 4 h

memory=140m (1.1 GB)

explicit

1 cpu: 270000 h (2.4E10 cycles)

Automatic Time Step Control

Automatic time step control adjusts stepsize during the simulation

- very persistent, reliable

***CONTROL_IMPLICIT_AUTO**

After successful steps

- compare iteration count to target value ITEOPT
- increase/decrease size of next step if difference exceeds window ITEWIN

After failed steps

- decrease step size
- back up, repeat failed step with new DT

Exponential algorithm for adjusting step size

- increase stepsize by 1/5 decade until DTMAX is reached
- decrease stepsize by 1/3 decade until DTMIN is reached
- error termination if convergence fails when DT=DTMIN

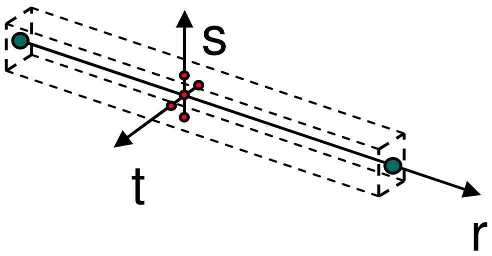
Implicit Capabilities: Element Types

Brick Elements: 1, 2, 3, 4, 10, 15, 16, 18

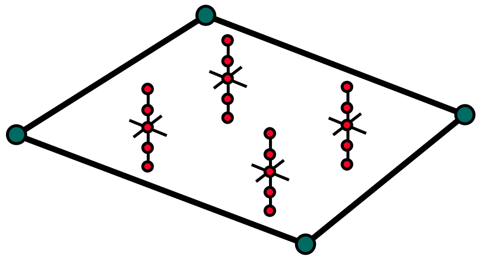
Beam (and 2D Shell) Elements: 1, 2, 3, 4, 5, 6, 7, 8, 9

Shell (and 2D Solid) Elements: 2, 4, 6, 10, 12, 13, 15, 16, 17, 18, 20, 21

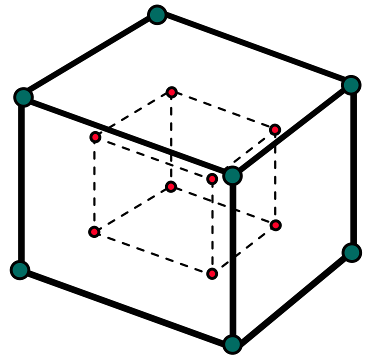
→ ***recommended*** **elements for implicit, e.g.**



Hughes-Liu:
type **1** (default)



fully integrated:
type **16** (not default!)



fully integrated S/R:
type **2** (not default!)

Alternate shell element formulations are substituted if requested elements are not available for implicit

Implicit Capabilities: Material Models

- stiffness matrix terms require extra evaluation of $\delta\sigma/\delta\varepsilon$
- some material models only available for selected element formulations

3D Solid Elements

- 1-7, 11, 12, 13, 14-17, 18, 19, 20, 21-23, 24, 26, 27, 30, 31, 33, 35, 36, 38, 41-50, 51-53, 57, 59, 60-62, 63, 64, 65, 70, 72, 73, 75-80, 83-85, 87-89, 91, 92, 96, 98, 100, 102, 103, 104, 105, 106, 107, 109-112, 115, 124, 126-129, 141-145, 161, 177, 178, 192, 193

Shell Elements

- 1-4, 6, 9, 18, 20, 21-23, 24, 27, 32, 36, 37, 41-50, 54, 55, 60, 76, 77, 91, 92, 98, 99, 103, 104, 106, 107, 116-118, 123

Beam Elements

- 1, 3, 4, 6, 9, 18, 20, 24, 41-50, 100

2D Solid Elements

- 1-7, 9, 12, 13, 18, 20, 24, 26, 41-50, 57, 63, 103, 104, 106, 107

Implicit Capabilities: Contact Interfaces

Several contact interfaces are available for implicit analysis

- *CONTACT_SURFACE_TO_SURFACE
- *CONTACT_NODES_TO_SURFACE
- *CONTACT_ONE_WAY_SURFACE_TO_SURFACE
- *CONTACT_FORMING_… (three variations)
- *CONTACT_AUTOMATIC_… (three variations)
- *CONTACT_TIED, …TIED_OFFSET
- *CONTACT_AUTOMATIC_SINGLE_SURFACE

All implicit contact interfaces use the penalty method, except TIED

Shooting node logic is automatically disabled for implicit

- SNLOG=1 on optional contact interface card "B"

Oriented normal vectors are recommended

Automatic contact types can fail due to large implicit stepsize

DYNA MORE

Nonlinear Convergence Problems

Convergence trouble is the most common problem

Error messages displayed by LS-DYNA, e.g.

- **iteration limit reached**

  displacement and energy tolerances were not satisfied, abandon step
- **divergence**

  out-of-balance force **R** is growing, reform **K** and continue iterations
- **negative eigenvalues**

  error from linear equation solver while computing $\mathbf{K}^{-1}$

Procedures for solving convergence problems

- determine reason for termination (examine error messages)
- activate print flags to get more information
- view deformed geometry during iteration process using "d3iter" database
- carefully inspect input deck
- see user’s manual: Appendix M

Current and Future Developments

MPP Implicit is nearing completion with all capabilities implemented

- Implicit simulation (statics and dynamics)
- Springback
- Vibration and Buckling analysis → **Version 971**
- Constraint and Attachment modes

Transition from dynamic to static

- e.g. gravity loading, roof crush

Implementation of still missing features, e.g.

- e.g. consistent tangent stiffness for more materials
- elements, e.g. type 13 tetrahedron for bulk forming
- contact types
- seatbelts
- airbags: fabric materials, inflator models, soft=2 - contact