January 2004

Leaders in Cutting Edge Technology
Software, Hardware and Services for The Engineering Community
Major Global Participants

Distributor and Consultancy Service Participants

LEAP **Australia**	**Altair** **Western Region US**	**DYNAMAX** **US**	**Cril Technology** **Simulation** **France**
GISSETA **India**	**MFAC** **Canada**	**DYNA*more*** **Germany**	**Flotrend** **Taiwan**
KOSTECH **Korea**	**ERAB** **Sweden**	**THEME** **Korea**	**ALTAIR** **Italy**
CAD-FEM **Germany**	**Numerica SRL** **Italy**	**ANSYS** **China**	**MSC.Software** **China**
SE&CS **US**			

Educational Participants and Contributing Authors

Dr. Ted. Belytschko **US**	**Dr. Bhavin V. Mehta** **US**	**Dr. Taylan Altan** **US**	**Dr. David Benson** **US**
Dr. Alexey L. **Borovkov** **Russia**	**Prof. Gennaro** **Monacelli** **Italy**	**Prof. Ala Tabiei** **US**	**Tony Taylor** **US**

Articles

LS-PREPOST FEATURES

- Full LS-DYNA 970 keyword support
- Subsystem is introduced for include files and imported model
- Extensive model manipulation features
- LS-DYNA data creation
- Airbag Folding
- Occupant positioning – improved capabilities
- Metal forming related features
- Node and Element Editing
- Element Generation
- Meshing capability
- Other features
    - Material database
    - 2D meshing sketch board
    - 201 Head impact positioning
    - SPH element generation

Keyword Input

- Each keyword has its own form for input and editing
- Keyword data that is present in the model
  will be highlighted with color
- There are over 800 keyword entities
- Comment card is available for each keyword input

Keyword Input Form

- Keyword input forms match LS-DYNA manual
- Each data field is identified by its name
- An explanation of the field is shown with a simple click in the field or the field name
- Simple selection button is used for the data field with pre-defined values
- A popup table can be used as an aid to transfer data to the selected field
- Link data can be viewed with a click on the name

Keyword reader and writer

- Keyword Reader support
    - LS-DYNA versions LS960, LS970, LS971
- Material keyword with name or number
    - *MAT_POWER_LAW_PLASTICITY or *MAT_018
- Keyword Writer support
    - LS-DYNA version LS960, LS970 or LS971 formats
    - More control on keyword card variable names output
    - More control and user defined on output keyword sequence

Subsystems

- The subsystem gives user better control over a model with many imported files
- A subsystem can consist of any keyword data
- Each include file will be treated as one subsystem
- Parts/Elements/Keyword data can be moved between different subsystems
- Subsystems can be output as one single files or many individual files

Model Manipulation

- A whole or portion of the model can be translated, rotated, scaled, transformed, reflected or projected
- New elements can also be created with each of these operations
- Shell element/segment normal check and reverse, auto reverse with seed element
- Move or copy elements from one part to another part
- Extensive element quality check
  - Feature angle, Warping, Aspect ratio, Characteristic length ,Internal angles, etc.
- Duplicated grid elimination (grids have the same coordinates)
- Free edges detection
- Referenced and un-referenced data check
  - Delete un-referenced data
  - Show referenced data
- ID renumbering interface
  - For all data (one single operation)
  - For each individual keyword data
    - Referenced data will be automatically updated

LS-DYNA data creation

- Set data (Beam, Shell, Node, Part, Segment, Discrete, etc.)
- Part data
- Mass element
- Nodal SPC data
- Initial Velocity
- Constrained Nodal Rigid Body
- Spot weld data
- General weld data
- Rigid walls

Airbag Folding

- The Airbag folding menu is designed to make airbag folding simple and straight forward.
- The folding procedure leads to a list of fold instructions which can be saved and reloaded later.
- Thin, Thick, Tuck and Spiral Folds can be defined.
- Folds can be examined via the Section Plane menu for good shape and freedom from nodal intrusions.
- Whole folding procedure can be stepped through and animated

Occupant Positioning

- Multiple dummies can be imported into one single model
- Move a dummy in a model to the desired location
- Use a tree file to define joints and limbs
- Rotate limbs of the dummies about their joints
- All related keyword data will be transformed
- Manipulation of the dummy model is now recorded and can be reset to its original position
- Final positioning can be saved as a LS-DYNA keyword input deck

Metal Forming Related Features

- Creation of new parts by offsetting elements along the element normal direction
- Separation measure between parts can be displayed as fringe plot
- Part travel distance to another part before contact can be calculated
- Multiple section cuts for different states or locations
- Curves creation, break, join, import and export
- Draw bead showing and creation
- Binder wall and Guide pin creation
- Part trimming

Node Edit

- Create nodes
    - Create one and more by coordinate or on element
- Replace node
    - Join two nodes to one by node1, node2 or midpoint
- Align nodes
    - Defined a line and project selected node onto the line
- Modify nodes
    - Pick a node by mouse and drag it to new position or input coordinate
    - Element Quality Check when moving node

Element Editing

- Element quality check
- Create element
    - Create new Beam,Tria, Quad, Tetra, Penta and Hexa by picking nodes on mesh
- Split element
    - Split elements by ten different methods   constrained adaptivity will be automatically created when needed
- Delete element
    - Un-referenced nodes can also be deleted

Element Generation

- Element generation from other elements

- o Beam
  - From Shell Edges
  - From Curves
- Shell
  - o From Solid faces and surfaces
  - o Edge drag or spin
  - o Curve drag or spin
- Solid
  - o From Shell drag
  - o From Shell spin
  - o From Shell offset
  - o 4 node tetrahedron to 10 node tetrahedron

Meshing Capability

- Incorporated ETA's Meshers
- Tool mesher – for metal stamping application
  - o Gives very good mesh quality for curves surfaces
  - o Gives large element size for flat area
- Topology mesher – for general crash application
  - o With Auto repair feature
- IGES or VDA file input
- Mesh quality controls

Other Features

- 201 Head Impact Positioning
  - o Multiple head can be positioned in the same model
  - o Head can be tilted vertically or rotated horizontally interactively
  - o Configuration file can be setup to have head model loaded automatically
  - o Multiple LS-DYNA keyword files can be output for different head positions
- SPH Element Generation
  - o SPH elements can be generated in simple geometries such as boxes, spheres and cylinders
  - o Material models for SPH can be automatically setup or can be picked by the users
- Material database
  - o Save user frequently used material data in keyword format
  - o Allow users to share material data with other
  - o Have private and public copy of material data
  - o Load curves (stress stain curves) will be saved along with the material data
  - o Local coordinates systems will also be saved with the material data
- 2D meshing sketch board
- Create points, lines, and arcs
  - o Use MAZE's Magic mesher for mesh generation
  - o For 2D applications

With every form of life on Earth dependent on water, it's difficult to imagine research more essential than deciphering the future of the planet's oceans. But global-scale research requires global-scale observations and models, criteria that rapidly thin the field of potential leaders in ocean studies.

One such leader enjoys a particular edge. "Using the vantage point of space, NASA gains an understanding of our home planet that we could never achieve were we bound to the Earth's surface," notes Dr. Ghassem R. Asrar, associate administrator of NASA's Earth Science Enterprise. NASA's remarkable 45-year history and vast scientific and engineering resources have helped the agency launch numerous research missions to understand and protect planet Earth.
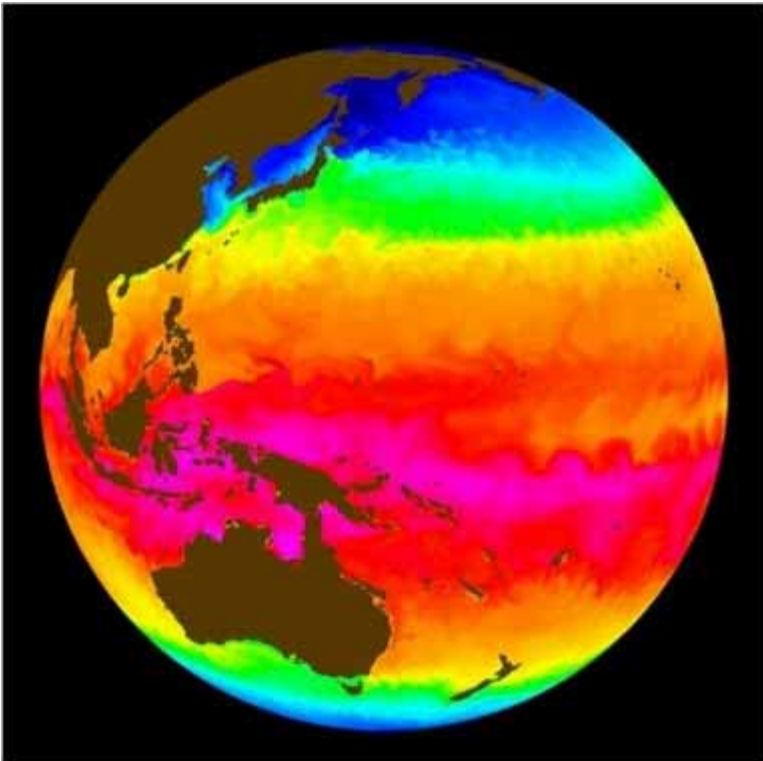
This is an ECCO simulation of ocean conditions in early 1993. Ocean surface temperatures range from -1.9 degrees Centigrade (in dark blue) to 37 degrees Centigrade (in pink/white).

One such mission is underway at NASA Ames Research Center in Mountain View, Calif. Ames deploys some of the world's most powerful computing systems to achieve things that, quite literally, have never been done before. Recently, Earth sciences researchers from Ames and NASA's Jet Propulsion Laboratory (JPL) did it again.

Just weeks after attaining record levels of sustained performance and scalability on a 256-processor global shared-memory SGI® Altix™ 3000 system, the team at NASA Ames doubled the size of its Altix™ system-achieving 512 processors in a single image, by far the largest supercomputer ever to run on the Linux® operating system. (NASA announced its technical feat at the SC2003 supercomputing conference.) NASA's effort is part an intra-agency collaborative research program between NASA Ames, JPL and NASA's Goddard Space Flight Center to accelerate the science return for large-scale earth modeling problems.

The group's current focus is to apply its 512-processor Altix system toward scaling the performance of ECCO (Estimating the Circulation and Climate of the Ocean). An application collaboratively developed by the Massachusetts Institute of Technology, JPL, and the Scripps Institution of Oceanography, ECCO is designed to improve the understanding of large-scale ocean dynamics by assimilating real-time ocean conditions as measured by satellites and onsite observation. With the more powerful Altix system, NASA researchers can progress to higher resolution models that better match with satellite observation.

Recently, NASA Earth sciences researchers doubled the size of the agency's SGI Altix™ system - achieving 512 processors in a single image, by far the largest supercomputer ever to run on the Linux® operating system

" **We want to know** how the Earth system is changing, and how those changes affect life on Earth. " – Dr. Walt Brooks, NASA Ames Advanced Supercomputing Division Chief

With its 256-processor Altix system, NASA achieved the fastest ECCO results ever recorded, with an ECCO ocean circulation model executing a one-quarter-degree global problem at a rate of 1.4 simulated years per wall clock day. Researchers now are aiming for a near-linear advance in performance on the larger single system image. While such a goal can be daunting, if not impossible, for traditional computer systems, NASA is able to take full advantage of its Altix system's 512 Intel® Itanium® 2 processors by leveraging the latest Intel® compilers, a robust and proven 64-bit Linux operating environment, and the high-bandwidth SGI® NUMAflex™ global shared-memory architecture.

"Large-scale problems require an extremely low-latency interface," notes Bob Ciotti, lead for the Terascale Application group at NASA Ames. "That determines how well you'll scale on these tightly coupled problems. And by having a single system image, we benefit from a more efficient and simpler programming development environment and a more robust I/O architecture that's a good match for applications where we're pushing lots of data."

" **This effort will pave the way** for us to make full use of satellite observations by bringing satellite data directly into the model. And that just wasn't possible before. "
– Dr. Ichiro Fukumori, ECCO Project Scientist from JPL

"Lots of data" might be something of an understatement. All told, NASA Ames' Earth sciences and aerospace research has generated more than a petabyte (or 1,024 terabytes) of data. With the acquisition of the Altix system-first 128 processors, then another 128, and finally another 256-NASA also invested in a total of 28 terabytes of SGI® InfiniteStorage capacity. The Fibre Channel storage installation keeps simulation and observational data accessible to ECCO ocean circulation models-an advantage that will prove even more important as ECCO models increase in resolution and produce even larger data sets.

With applications designed to run on hundreds more processors than they do today, NASA researchers have made specific efforts to optimize the return on their Altix investment. Running their codes on the record 512-processor single system image has allowed NASA researchers to see immediate scalability benefits with little or no porting or tuning. The first Linux OS-based supercomputers to enable global shared-memory, SGI Altix systems are powered by the third-generation NUMAflex supercomputing architecture. With NUMAflex, high-performance computing (HPC) innovators like NASA can analyze

data sets as whole entities, without breaking them up into smaller segments to be handled by individual processors.

"Technically, this effort will pave the way for us to make full use of satellite observations by bringing satellite data directly into the model," says Dr. Ichiro Fukumori, ECCO Project Scientist from JPL. "We'll move toward a much more data-driven model. Our models will include real-world data that will constantly refine and constrain our results."

Today for instance, ECCO models achieve one-quarter-degree resolution, which is roughly equivalent to 25 kilometers. But eventually, NASA wants to drive the level of detail to one-tenth of a degree. "This will make our results much more accurate," adds Fukumori. "And that just wasn't possible before."

With code capable of scaling beyond 512 processors, NASA will continue to push the limits of computing as it seeks answers to some of the most important questions facing humankind today. In the end, SGI and Altix are helping NASA uncover insights that may well touch the lives of every living thing on the planet.

"The Earth is complex and fragile, and the only known harbor of life in the solar system," notes Dr. Walt Brooks, NASA Ames Advanced Supercomputing Division chief. "We want to know how the Earth system is changing, and how those changes affect life on Earth. That's the work we do here."

FEA Information Participants
Commercial, Contributing, and Educational

Headquarters	Company	
Australia	Leading Engineering Analysis Providers	www.leapaust.com.au
Canada	Metal Forming Analysis Corp.	www.mfac.com
China	ANSYS – China	www.ansys.com.cn
China	MSC.Software – China	www.mscsoftware.com.cn
France	Cril Technology Simulation	www.criltechnology.com
Germany	DYNA*more*	www.dynamore.de
Germany	CAD-FEM	www.cadfem.de
India	GissEta	www.gisseta.com
Italy	Altair Engineering srl	www.altairtorino.it
Italy	Numerica srl	www.numerica-srl.it
Japan	The Japan Research Institute, Ltd	www.jri.co.jp
Japan	Fujitsu Ltd.	www.fujitsu.com
Japan	NEC	www.nec.com
Korea	THEME Engineering	www.lsdyna.co.kr
Korea	Korean Simulation Technologies	www.kostech.co.kr
Russia	State Unitary Enterprise - STRELA	www.ls-dynarussia.com
Sweden	Engineering Research AB	www.erab.se
Taiwan	Flotrend Corporation	www.flotrend.com
UK	OASYS, Ltd	www.arup.com/dyna
USA	IBM	www-1.bim.com/servers/deepcomputing
USA	INTEL	www.intel.com
USA	Livermore Software Technology	www.lstc.com
USA	Engineering Technology Associates	www.eta.com
USA	ANSYS, Inc	www.ansys.com
USA	Hewlett Packard	www.hp.com
USA	SGI	www.sgi.com

USA	MSC.Software	www.mscsoftware.com
USA	DYNAMAX	www.dynamax-inc.com
USA	AMD	www.amd.com
USA	SE&CS	www.sonic.net/lschwer/SECS/index.htm
Educational Participants & Contributing Authors		
USA	Dr. T. Belytschko	Northwestern University
USA	Dr. D. Benson	Univ. California – San Diego
USA	Dr. Bhavin V. Mehta	Ohio University
USA	Dr. Taylan Altan	The Ohio State U – ERC/NSM
USA	Prof. Ala Tabiei	University of Cincinnati
USA	Tony Taylor	Irvine Aerospace Inc.
Russia	Dr. Alexey I. Borovkov	St. Petersburg State Tech. University
Italy	Prof. Gennaro Monacelli	Prode – Elasis & Univ. of Napoli, Federico II

*Keyword

KEYWORD_*OPTION
- An option, *ID*, is available to assign a prefix to all output and scratch filenames, i.e., the file name for D3PLOT becomes:

File_name_prefix.D3PLOT

- The prefix can also be assigned by the command option, jobid on the execute line:

ls971 I=input.k   jobid=*File_name_prefix*

- Multiple jobs can now be executed in the same directory

Local coordinate systems:

- *CONSTRAINED_LOCAL (new)
  - Like *CONSTRAINED_GLOBAL but in a local coordinate system
- *DEFINE_COORDINATE_VECTOR
  - Can now be attached to a nodal point and transformed to follow the rotations of the attached nodal point.
- *DEFINE_VECTOR
  - The x, y, and z components of a vector are defined in a local system
- *LOAD_BODY_…
  - Body forces now can be applied in a local system
- *PART_MOVE
  - Parts can be moved in a local system

*Constrained_spline

- A cubic spline interpolation element
  - Displacements and slopes are matched at endpoints
    - Based on beam theory
    - Widely used in NASTRAN
- Provides a way of connecting regions of different mesh density
- Works explicitly and implicitly
- Implemented for NASTRAN compatibility

*Control_implicit_inertia_relief

- New feature for implicit computations to allow analyses of models with rigid body modes, e.g., aircraft in flight
- Computes the rigid body modes and uses these rigid modes to constrain the motion
- Works for linear statics, both single and multi-step
- Input requires threshold eigenvalue for identifying the rigid body modes. Default=0.001hz

*Define_curve_function

- Can be referenced just like any other curve
- Arbitrary analytic expressions of any complexity
    - Read in as ASCII FORTRAN expression
- Can reference other curves, either tabulated or analytic
    - For complete generality, a dependency tree is created so curves can reference curves that reference curves, etc.
- Examples of analytic expressions:
    - 42.5*sin(time*pi/20.)
    - Max(LC10,sqrt(LC122*5.)), LC10 and LC122 are load curve ID's
- Expressions can be functions of time, displacements, velocities, relative displacements, etc.

10-node tetrahedron elements

- Implemented for MPP and SMP
- Isoparametric formulation in 970
- composite formulation in 971
- Cost
    - Isoparametric form is the same cost per element cycle as SRI solid element. Δt is smaller due to high frequencies
    - Composite form is twice the cost
- Contact treated automatically by 4 triangles for each face
- Available for both implicit and explicit calculations
- 4 or 5 integration points, constant pressure for nonlinear materials

Composite Tetrahedron

- Ten-node tetrahedron divided into 12 sub-tetrahedrons
- Linear displacement in sub-tetrahedron
- Assumed linear strain, or constant volumetric and linear deviatoric strain over entire tetrahedron
- Strain operators obtained from compatibility condition

$$\int_{\Omega_e} \delta\overline{\sigma} : (\nabla_s \mathbf{u} - \overline{\varepsilon})d\Omega = 0$$
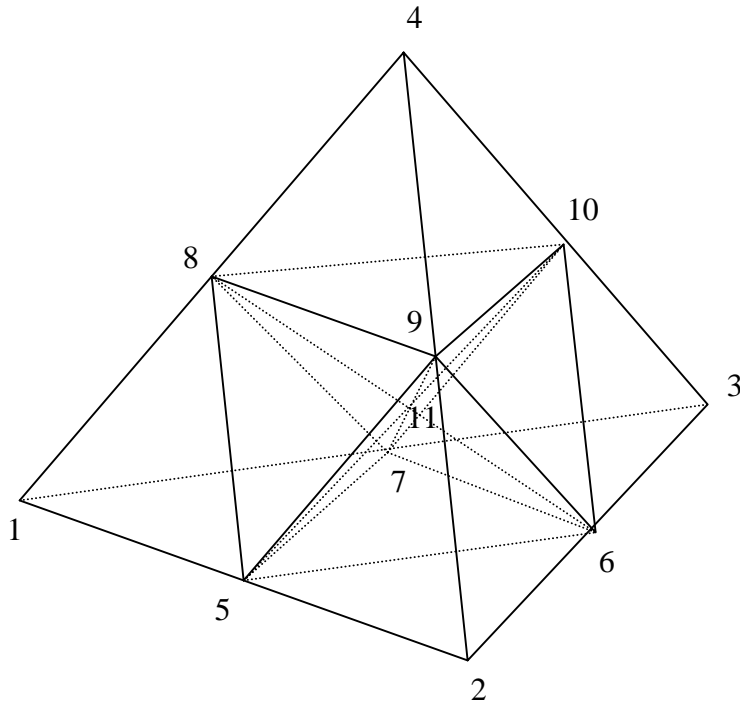
■Advantages:
　　■Optimal lumped mass matrix
　　■More flexible and more accurate than constant-strain tetrahedral element
　　■Easy implementation of contact algorithms
■Disadvantages:
　　■Speed: slightly slower than the quadratic isoparametric tetrahedron

New material models

■　*MAT_BARLAT_YLD2000　(Version 971 and next release of 970)

The primary application of this material model is in aluminum sheet metal stamping and forming simulations.　The yield condition for this material can be written

$$f(\boldsymbol{\sigma}, \varepsilon_p) = \sigma_{\text{eff}}(\sigma_{xx}, \sigma_{yy}, \sigma_{xy}) - \sigma_Y(\varepsilon_p) \leq 0$$

where

$$\sigma_{\text{eff}}(\sigma_{xx}, \sigma_{yy}, \sigma_{xy}) = \left(\frac{1}{2}(\phi' + \phi'')\right)^{1/a}$$

$$\phi' = \left|X'_1 - X'_2\right|^a$$

$$\phi'' = \left|2X''_1 + X''_2\right|^a + \left|X''_1 + 2X''_2\right|^a$$

The X'_i and X''_i $\sigma_y = a - be^{-c\varepsilon^p}$ are eigenvalues of X'_{ij} and X''_{ij} and are given by

$$X'_1 = \frac{1}{2}\left(X'_{11} + X'_{22} + \sqrt{(X'_{11} - X'_{22})^2 + 4X'^2_{12}}\right)$$

$$X'_2 = \frac{1}{2}\left(X'_{11} + X'_{22} - \sqrt{(X'_{11} - X'_{22})^2 + 4X'^2_{12}}\right)$$

and

$$X''_1 = \frac{1}{2}\left(X''_{11} + X''_{22} + \sqrt{(X''_{11} - X''_{22})^2 + 4X''^2_{12}}\right)$$

$$X''_2 = \frac{1}{2}\left(X''_{11} + X''_{22} - \sqrt{(X''_{11} - X''_{22})^2 + 4X''^2_{12}}\right)$$

respectively. The X'_{ij} and X''_{ij} are given by

$$\begin{pmatrix} X'_{11} \\ X'_{22} \\ X'_{12} \end{pmatrix} = \begin{pmatrix} L'_{11} & L'_{12} & 0 \\ L'_{21} & L'_{22} & 0 \\ 0 & 0 & L'_{33} \end{pmatrix} \begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{pmatrix} \qquad \begin{pmatrix} X''_{11} \\ X''_{22} \\ X''_{12} \end{pmatrix} = \begin{pmatrix} L''_{11} & L''_{12} & 0 \\ L''_{21} & L''_{22} & 0 \\ 0 & 0 & L''_{33} \end{pmatrix} \begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{pmatrix}$$

where

$$\begin{pmatrix} L'_{11} \\ L'_{12} \\ L'_{21} \\ L'_{22} \\ L'_{33} \end{pmatrix} = \frac{1}{3}\begin{pmatrix} 2 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_7 \end{pmatrix} \qquad \begin{pmatrix} L''_{11} \\ L''_{12} \\ L''_{21} \\ L''_{22} \\ L''_{33} \end{pmatrix} = \frac{1}{9}\begin{pmatrix} -2 & 2 & 8 & -2 & 0 \\ 1 & -4 & -4 & 4 & 0 \\ 4 & -4 & -4 & 1 & 0 \\ -2 & 8 & 2 & -2 & 0 \\ 0 & 0 & 0 & 0 & 9 \end{pmatrix} \begin{pmatrix} \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \\ \alpha_8 \end{pmatrix}$$

The parameters α_1 to α_8 are the parameters that determines the shape of the yield surface.

■*MAT_RATE_SENSITIVE_COMPOSITE_FABRIC (Version 971 and next release of 970)

Depending on the type of failure surface, this model may be used to model rate sensitive composite materials with unidirectional layers, complete laminates, and woven fabrics. A viscous stress tensor, based on an isotropic Maxwell model with up to six terms in the Prony series expansion, is superimposed on the rate independent stress tensor of the composite fabric. The viscous stress tensor approach should work reasonably well if the stress increases due to rate affects are up to 15% of the total stress. This model is implemented for both shell and thick shell elements. The viscous stress tensor is effective at eliminating spurious stress oscillations.

■*MAT_CSCM (Version 971)

This is a smooth or continuous surface cap model and is available for solid elements in LS-DYNA developed by Aptek for the FHWA and added to LS-DYNA by Yvonne Murray of Aptek. The following is the documentation provided by Aptek. In this model user has the option of inputting the material properties, or requesting default material properties for normal strength concrete.

Model Formulation and Input Parameters

This is a cap model with a smooth intersection between the shear yield surface and hardening cap, as shown in the Figure. The initial damage surface coincides with the yield surface. Rate effects are modeled with viscoplasticity. A manual containing a complete theoretical description, with references and example problems, is under development and will be available in December 2004.
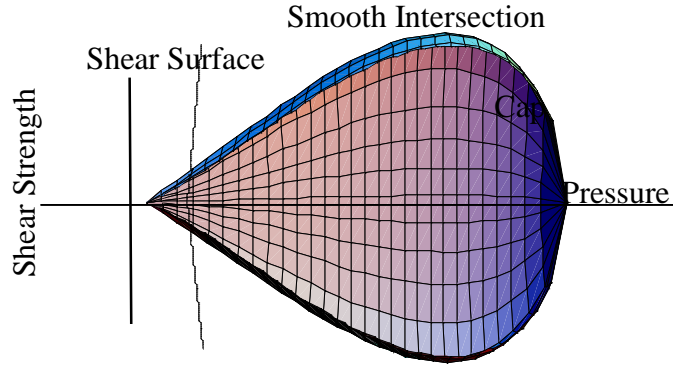
General shape of the concrete model yield surface in two-dimensions.

Stress Invariants. The yield surface is formulated in terms of three stress invariants: J_1 is the first invariant of the stress tensor, J_2' is the second invariant of the deviatoric stress tensor, and J_3' is the third invariant of the deviatoric stress tensor. The invariants are defined in terms of the deviatoric stress tensor, S_{ij} and pressure, P, as follows:

$$J_1 = 3P$$

$$J_2' = \frac{1}{2} S_{ij} S_{ij}$$

$$J_3' = \frac{1}{3} S_{ij} S_{jk} S_{ki}$$

Plasticity Surface. The three invariant yield function is based on these three invariants, and the cap hardening parameter, κ, as follows:

$$f(J_1, J_2', J_3', \kappa) = J_2' - \Re^2 F_f^2 F_c$$

Here F_f is the shear failure surface, F_c is the hardening cap, and $\Re$ is the Rubin three-invariant reduction factor. The cap hardening parameter κ is the value of the pressure invariant at the intersection of the cap and shear surfaces.

Trial elastic stress invariants are temporarily updated via the trial elastic stress tensor, σ^T. These are denoted J_1^T, $J_2'^T$, and $J_3'^T$. Elastic stress states are modeled when $f(J_1^T, J_2'^T, J_3'^T, \kappa^T) \leq 0$. Elastic-plastic stress states are modeled when $f(J_1^T, J_2'^T, J_3'^T, \kappa^T) > 0$. In this case, the plasticity algorithm returns the stress state to the yield surface such that $f(J_1^P, J_2'^P, J_3'^P, \kappa^P) = 0$. This is accomplished by enforcing the plastic consistency condition with associated flow.

Shear Failure Surface. The strength of concrete is modeled by the shear surface in the tensile and low confining pressure regimes:

$$F_f(J_1) = \alpha - \lambda \exp^{-\beta J_1} + \theta J_1$$

17

Here the values of $\alpha, \beta, \lambda,$ and θ are selected by fitting the model surface to strength measurements from triaxial compression (TXC) tests conducted on plain concrete cylinders.

Rubin Scaling Function. Concrete fails at lower values of $\sqrt{3J_2'}$ (principal stress difference) for triaxial extension (TXE) and torsion (TOR) tests than it does for TXC tests conducted at the same pressure. The Rubin scaling function $\mathfrak{R}$ determines the strength of concrete for any state of stress relative to the strength for TXC, via $\mathfrak{R}F_f$. Strength in torsion is modeled as $Q_1 F_f$. Strength in TXE is modeled as $Q_2 F_f$, where:

$$Q_1 = \alpha_1 - \lambda_1 exp^{-\beta_1 J_1} + \theta_1 J_1$$

$$Q_2 = \alpha_2 - \lambda_2 exp^{-\beta_2 J_1} + \theta_2 J_1$$

Cap Hardening Surface. The strength of concrete is modeled by a combination of the cap and shear surfaces in the low to high confining pressure regimes. The cap is used to model plastic volume change related to pore collapse (although the pores are not explicitly modeled). The isotropic hardening cap is a two-part function that is either unity or an ellipse:

$$F_c(J_1, \kappa) = 1 - \frac{[J_1 - L(\kappa)] \left[|J_1 - L(\kappa)| + J_1 - L(\kappa) \right]}{2 \left[X(\kappa) - L(\kappa) \right]^2}$$

where $L(\kappa)$ is defined as:

$$L(\kappa) = \begin{cases} \kappa & if \quad \kappa > \kappa_0 \\ \kappa_0 & otherwise \end{cases}$$

The equation for F_c is equal to unity for $J_1 \leq L(\kappa)$. It describes the ellipse for $J_1 > L(\kappa)$. The intersection of the shear surface and the cap is at $J_1 = \kappa$. κ_0 is the value of J_1 at the *initial* intersection of the cap and shear surfaces before hardening is engaged (before the cap moves). The equation for $L(\kappa)$ restrains the cap from retracting past its initial location at κ_0.

The intersection of the cap with the J_1 axis is at $J_1 = X(\kappa)$. This intersection depends upon the cap ellipticity ratio R, where R is the ratio of its major to minor axes:

$$X(\kappa) = L(\kappa) + RF_f(L(\kappa))$$

The cap moves to simulate plastic volume change. The cap expands ($X(\kappa)$ and κ increase) to simulate plastic volume compaction. The cap contracts ($X(\kappa)$ and κ decrease) to simulate plastic volume expansion, called dilation. The motion (expansion and contraction) of the cap is based upon the hardening rule:

$$\varepsilon_v^p = W(1 - \exp^{-D_1(X - X_0) - D_2(X - X_0)^2})$$

Here ε_v^p the plastic volume strain, W is the maximum plastic volume strain, and D_1 and D_2 are model input parameters. X_0 is the initial location of the cap when $\kappa = \kappa_0$.

The five input parameters (X_0, W, D_1, D_2, and R) are obtained from fits to the pressure-volumetric strain curves in isotropic compression and uniaxial strain. X_0 determines the pressure at which compaction initiates in isotropic compression. R, combined with X_0, determines the pressure at which compaction initiates in uniaxial strain. D_1, and D_2 determine the shape of the pressure-volumetric strain curves. W determines the maximum plastic volume compaction.

Shear Hardening Surface. In unconfined compression, the stress-strain behavior of concrete exhibits nonlinearity and dilation prior to the peak. Such behavior is be modeled with an initial shear yield surface, $N_H F_f$, which hardens until it coincides with the ultimate shear yield surface, F_f. Two input parameters are required. One parameter, N_H, initiates hardening by setting the location of the initial yield surface. A second parameter, C_H, determines the rate of hardening (amount of nonlinearity).

__Damage.__ Concrete exhibits softening in the tensile and low to moderate compressive regimes.

$$\sigma_{ij}^d = (1-d)\sigma_{ij}^{vp}$$

A scalar damage parameter, d, transforms the viscoplastic stress tensor without damage, denoted σ^{vp}, into the stress tensor with damage, denoted σ^d. Damage accumulation is based upon two distinct formulations, which we call brittle damage and ductile damage. The initial damage threshold is coincident with the shear plasticity surface, so the threshold does not have to be specified by the user.

Ductile Damage. Ductile damage accumulates when the pressure (P) is compressive and an energy-type term, τ_c, exceeds the damage threshold, τ_{0c}. Ductile damage accumulation depends upon the total strain components, ε_{ij}, as follows:

$$\tau_c = \sqrt{\frac{1}{2}\sigma_{ij}\varepsilon_{ij}}$$

The stress components σ_{ij} are the elasto-plastic stresses (with kinematic hardening) calculated before application of damage and rate effects.

Brittle Damage. Brittle damage accumulates when the pressure is tensile and an energy-type term, τ_t, exceeds the damage threshold, τ_{0t}. Brittle damage accumulation depends upon the maximum principal strain, ε_{max}, as follows:

$$\tau_t = \sqrt{E\varepsilon_{max}^2}$$

Softening Function. As damage accumulates, the damage parameter d increases from an initial value of zero, towards a maximum value of one, via the following formulations:

Brittle Damage
$$d(\tau_t) = \frac{0.999}{D}\left[\frac{1+D}{1+D\exp^{-C(\tau_t - \tau_{0t})}} - 1\right]$$

Ductile Damage
$$d(\tau_c) = \frac{dmax}{B}\left[\frac{1+B}{1+B\exp^{-A(\tau_c - \tau_{0c})}} - 1\right]$$

The damage parameter that is applied to the six stresses is equal to the current maximum of the brittle or ductile damage parameter. The parameters A and B or C and D set the shape of the softening curve plotted as stress-displacement or stress-strain. The parameter *dmax* is the maximum damage level that can be attained. It is calculated internally calculated and is less than one at moderate confining pressures. The compressive softening parameter, A, may also be reduced with confinement, using the input parameter *pmod*, as follows:

$$A = A(dmax + 0.001)^{pmod}$$

Regulating Mesh Size Sensitivity. The concrete model maintains constant fracture energy, regardless of element size. The fracture energy is defined here as the area under the stress-displacement curve from peak strength to zero strength. This is done by internally formulating the softening parameters A and C in

terms of the element length , l (cube root of the element volume), the fracture energy, G_f, the initial damage threshold, τ_{0t} or τ_{0c}, and the softening shape parameters, D or B.

The fracture energy is calculated from up to five user-specified input parameters (G_{fc}, G_{ft}, G_{fs}, $pwrc$, $pwrc$). The user specifes three distinct fracture energy values. These are the fracture energy in uniaxial tensile stress, G_{ft}, pure shear stress, G_{fs}, and uniaxial compressive stress, G_{fc}. The model internally selects the fracture energy from equations which interpolate between the three fracture energy values as a function of the stress state (expressed via two stress invariants). The interpolation equations depend upon the user-specified input powers $pwrc$ and $pwrt$, as follows.

$$\text{if the pressure is tensile} \qquad G_f = G_{fs} + trans(G_{ft} - G_{fs}) \quad where \quad trans = \left(\frac{-J_1}{\sqrt{3J_2'}} \right)^{pwrt}$$

$$\text{if the pressure is compressive} \quad G_f = G_{fs} + trans(G_{fc} - G_{fs}) \quad where \quad trans = \left(\frac{J_1}{\sqrt{3J_2'}} \right)^{pwrc}$$

The internal parameter *trans* is limited to range between 0 and 1.

Element Erosion. An element losses all strength and stiffness as $d \to 1$. To prevent computational difficulties with very low stiffness, element erosion is available as a user option. An element erodes when $d > 0.99$ and the maximum principal strain is greater than a user supplied input value, 1-ERODE.

Viscoplastic Rate Effects. At each time step, the viscoplastic algorithm interpolates between the elastic trial stress, σ_{ij}^T, and the inviscid stress (without rate effects), σ_{ij}^p, to set the viscoplastic stress (with rate effects), σ_{ij}^{vp} :

$$\sigma_{ij}^{vp} = (1-\gamma)\sigma_{ij}^T + \gamma\sigma_{ij}^p \qquad with \qquad \gamma = \frac{\Delta t / \eta}{1 + \Delta t / \eta}$$

This interpolation depends upon the effective fluidity coefficient, η, and the time step, Δt. The effective fluidity coefficient is internally calculated from five user-supplied input parameters and interpolation equations:

$$\text{if the pressure is tensile} \qquad \eta = \eta_s + trans(\eta_t - \eta_s) \qquad trans = \left(\frac{-J_1}{\sqrt{3J_2'}} \right)^{pwrt}$$

$$\text{if the pressure is compressive} \quad \eta = \eta_s + trans(\eta_c - \eta_s) \qquad trans = \left(\frac{J_1}{\sqrt{3J_2'}} \right)^{pwrc}$$

$$\eta_t = \frac{\eta_{0t}}{\dot{\varepsilon}^{N_t}} \qquad \eta_c = \frac{\eta_{0c}}{\dot{\varepsilon}^{N_c}} \qquad \eta_s = Srate\,\eta_t$$

The input parameters are η_{0t} and N_t for fitting uniaxial tensile stress data, η_{0c} and N_c for fitting the uniaxial compressive stress data, and *Srate* for fitting shear stress data. The effective strain rate is $\dot{\varepsilon}$.

This viscoplastic model may predict substantial rate effects at high strain rates ($\dot{\varepsilon}$ >100). To limit rate effects at high strain rates, the user may input overstress limits in tension (*overt*) and compression (*overc*). These input parameters limit calculation of the fluidity parameter, as follows:

$$if \ E\dot{\varepsilon}\eta > over \quad then \quad \eta = \frac{over}{E\dot{\varepsilon}}$$

where *over* = *overt* when the pressure is tensile, and *over* = *overc* when the pressure is compressive.

The user has the option of increasing the fracture energy as a function of effective strain rate via the *repow* input parameter, as follows:

$$G_f^{rate} = G_f \left(1 + \frac{E\dot{\varepsilon}\eta}{f'} \right)^{repow}$$

Here G_f^{rate} is the fracture energy enhanced by rate effects, and f' is the yield strength before application of rate effects (which is calculated internally by the model). The term in brackets is greater than, or equal to one, and is the approximate ratio of the dynamic to static strength. \

■*MAT_BRAIN_LINEAR_VISCOELASTIC (Version 971 and next release of 970)

This is a special material model for brain tissue based on a linear viscoelastic behavior. The shear relaxation behavior is described for the Maxwell model by:

$$G(t) = G + (G_0 - G_\infty)e^{-\beta t}$$

A Jaumann rate formulation is used

$$\overset{\nabla}{\sigma'_{ij}} = 2\int_0^t G(t - \tau) \, D'_{ij}(\tau)dt$$

where the prime denotes the deviatoric part of the stress rate, $\overset{\nabla}{\sigma}_{ij}$, and the strain rate D_{ij} . For the Kelvin model the stress evolution equation is defined as:

$$\dot{s}_{ij} + \frac{1}{\tau} s_{ij} = \left(1 + \delta_{ij}\right) G_0 \, \dot{e}_{ij} + \left(1 + \delta_{ij}\right) \frac{G_\infty}{\tau} \dot{e}_{ij}$$

Strain data, such as the maximum principal values, are written to an LS-DYNA database and may be used to predict damage.

■*MAT_PLASTICITY_WITH_DAMAGE_ORTHO_RCDC (Version 971 and next release of 970)

This is an elasto-visco-plastic material with an arbitrary stress versus strain curve and an arbitrary strain rate dependency. Damage is considered before rupture occurs using an orthotropic damage model that is combined with the RCDC damage model developed by Wilkins at LLNL. It is implemented for shell elements with multiple through thickness integration points. Directional damage begins after a

defined failure strain is reached in tension and continues to evolve until a tensile rupture strain is reached in either one of the two orthogonal directions. After rupture is detected at all integration points, the element is deleted. The purpose of this model is to model the failure of aluminum components in crash simulations.

The Rc-Dc model is defined as the following:

The damage D is given by

$$D = \int \omega_1 \omega_2 d\varepsilon^p$$

where ε^p is the equivalent plastic strain,

$$\omega_1 = \left(\frac{1}{1 - \gamma \sigma_m} \right)^\alpha$$

is a triaxial stress weighting term and

$$\omega_2 = \left(2 - A_D \right)^\beta$$

is a asymmetric strain weighting term. In the above σ_m is the mean stress and

$$A_D = \min \left(\left| \frac{S_2}{S_3} \right|, \left| \frac{S_3}{S_2} \right| \right)$$

Fracture is initiated when the accumulation of damage is

$$\frac{D}{D_c} > 1$$

where D_c is the a critical damage given by

$$D_c = D_0 \left(1 + b \left| \nabla D \right|^\lambda \right)$$

A fracture fraction,

$$F = \frac{D - D_c}{D_s}$$

defines the degradations of the material by the Rc-Dc model.

■ ***MAT_SIMPLIFIED_RUBBER/FOAM** (Version 971 and next release of 970)

The simplified rubber model has now been extended to highly compressible foams by including the additional term in the Hill strain energy density function related to compressibility. This material model provides a rubber and foam model defined by a single uniaxial load curve and, in order to include strain rate effects, by a family of uniaxial curves at discrete strain rates. The foam formulation is triggered by defining a Poisson's ratio which is assumed to be constant in order to simplify the implementation.

The Hill strain energy density function in terms of the principle stretches for this highly compressible foam is given by:

$$W = \sum_{j=1}^{m} \frac{C_j}{b_j} \left[\lambda_1^{b_j} + \lambda_2^{b_j} + \lambda_3^{b_j} - 3 + \frac{1}{n} \left(J^{-nb_j} - 1 \right) \right]$$

where C_j, b_j, and n are material constants and $J = \lambda_1 \lambda_2 \lambda_3$ represents the ratio of the deformed to the undeformed state. The principal Cauchy stresses are then give by:

$$t_i = \sum_{j=1}^{m} \frac{C_j}{J} \left[\lambda_i^{b_j} - J^{-nb_j} \right] \quad i = 1, 2, 3$$

This simplified model provides an approximation to these stress values without the need to fit the coefficients of the strain energy density function to the defined load curve.

Special Announcements and Highlights of News Pages

Personal Websites of Interest

Marsha Victory **FEA Information Inc.**	**horse rescue**	**www.livermorehorses.com**
Len Schwer **SE&CS**	**travel diaries -photos**	**www.schwer.net/LenSchwer/**
Ray Jurevicius **Jurevicius Engineering, Inc**	**Miscellaneous details about binoculars made from two 8" f/6.3 Newtonian telescopes.**	**www.j-engineering.com/ATM**

Posted on FEA Information and archived one month on the News Page

December 01	**SGI**	**Linux**
	ETA	**FEMB**
	CRIL	**Distributor: France**
December 08	**IBM**	**RS/6000 SP System**
	OASYS	**Primer, D3Plot**
	HP	**HP9000**
December 15	**Intel**	**PC Mobility**
	FUJITSU	**PRIMERGY**
	FLOTREND	**Distributor: Taiwan**
December 22	**AMD**	**Processor Based Server**
	NEC	**NEAX2400 IPX**
	STRELA	**Distributor – Russia**
December 29	**ANSYS**	**ANSYS 8.0**
	LSTC	**LS-OPT**
	KOSTECH	**Distributor: Korea**

EVENTS:

2004	
Feb 19	**LS-DYNA Users Conference** – China – hosted by MSC.Software
May 2-3	**8th International LS-DYNA Users** conference will again be held at the Hyatt Regency Dearborn, Fairlane Town Center, Dearborn, MI hosted by LSTC and ETA
May 10-12	**OPTECH04**, Optimization Technology Meeting 2004
May 24-26	**2004 ANSYS Users Conference and Exhibition** to be held in Pittsburgh, Pennsylvania, U.S.A.

A Quantitative Approach for Determining the Communication and Computation Costs in MPP LS-DYNA Simulations

Yih-Yih Lin
Hewlett-Packard Company
MR01-3
200 Forest Street
Marlborough, MA 01752
USA

Tel. 1-404-774-5278
Email: yih-yih.lin@hp.com

Keywords:
Communication cost, computation cost, speedup

ABSTRACT

As MPP LS-DYNA uses the message-passing paradigm to obtain parallelism, the elapsed time of an MPP LS-DYNA simulation comprises of two parts: computation cost and communication cost. A quantitative approach for determining the communication cost and, hence, the computation cost and the speedup of an

MPP LS-DYNA simulation is presented. Elapsed times, characteristics of interconnect networks—latency and bandwidth—and message patterns are first measured, and then the method of least square errors is applied to estimate the two costs. This approach allows one to predict the performance or the speedup of MPP LS-DYNA simulations with any interconnect networks whose characteristics are known.

INTRODUCTION—Aggregate Latency and Aggregate Bandwidth of Interconnects in MPP-LSDYNA Simulations

To run an N-processor MPP LS-DYNA simulation, or job, an interconnect network, or called simply as an interconnect, must first be established to connect the N processors; the collection of the N processors and the interconnect is called an N-processor cluster. In this paper, we will consider only the case that the N processors are of the same kind. For such a job, MPP LS-DYNA starts by decomposing the geometrical configuration of the model into N sub-domains. Each of the N processors is assigned to perform computation on one of the sub-domains; meanwhile, messages are passed among all those processors so that necessary physical conditions, such as force conditions, can be enforced. Let T^1_{comput}, T^2_{comput}, …, T^N_{comput} be each processor's computation cost, and let T^1_{comm}, T^2_{comm}, …, T^N_{comm} be each processor's communication cost. Define the computation cost T_{comput} as max $(T^1_{comput}, T^2_{comput}, …, T^N_{comput})$ and the communication cost T_{comm} as max$(T^1_{comm}, T^2_{comm}, …, T^N_{comm})$, respectively. Then the job's elapsed time can be described as:

$$T_{elapsed} = T_{comput} + T_{comm} \qquad (1)$$

For a given decomposition, the computation cost T_{comput} is fixed. In contrast, the communication cost T_{comm} varies with the characteristics of interconnects used. The term "speedup" is defined as the ratio $T_{elapsed, 1-processor} / T_{elapsed, N-processor}$. In general, speedups are smaller than N. Since for the 1-processor job the communication cost T_{comm} is zero, the perfect speedup, N, can be realized only under the unrealistic conditions of zero communication cost, i.e., $T_{comm} = 0$ and perfectly balanced decomposition, which renders $T^1_{comput} = T^2_{comput} = …= T^N_{comput}$.

Because of the assumption that the N processors are of the same kind, the variation of T^1_{comput}, T^2_{comput}, …, T^N_{comput} must arise out of the unbalanced decomposition of the N sub-domains. It is extremely difficult to find a universal algorithm to decompose a model with a balanced decomposition. MPP LS-DYNA does provide features, as documented in *pfile in parallel specific options*, for users to provide hints to get a more balanced decomposition than the default.

There are typically a large number of messages of various sizes transacting in an MPP LS-DYNA simulation. The communication cost T_{comm} is the sum of the communication costs of each message in the processor that obtains the maximal communication cost (called the "maximal" processor). The communication cost of a message depends solely on the two factors, latency and bandwidth, of the interconnect [1]:

Communication cost of a message = Latency + Message Size / Bandwidth

The latency is the sum of sender overhead, receiver overhead and time of flight; and the bandwidth refers to the maximum rate at which the interconnect can propagate information once the message enters the network. Messages of MPP LS-DYNA comprises of various different types, such as point-to-point communication and collective operations. In general, for a given interconnect, latency varies with message types, and bandwidth varies with message types and lengths. All the messages can be divided into m groups with the same latency, the same bandwidth and the same length. Considering messages of the "maximal" processor, let n_i, t^{lan}_i, t^{bw}_i and s_i be the ith group's number of messages, latency, bandwidth and message size, respectively. Then the job's communication cost can be described as follows:

$$T_{comm} = \Sigma^m_{i=1} \ n_i (t^{lan}_i + s_i / t^{bw}_i) \qquad (2)$$

It is well known that the most basic operation for message passing is the point-to-point, or so called ping-pong, communication. Let t^{lan} and t^{bw} be the latency and bandwidth of the ping-pong communication, and let α_i be the ratio t^{lan}_i / t^{lan} and β_i be the ratio t^{bw} / t^{bw}_i, respectively. Then formula (2) becomes

$$T_{comm} = (\Sigma^m_{i=1} n_i\alpha_i) t^{lan} + (\Sigma^m_{i=1} n_i\beta_is_i) / t^{bw} \qquad (3)$$

Further, let M be the number of messages, i.e., $M = \Sigma^m_{i=1}n_i$, and let s be the average message size. Setting

$$M\alpha = \Sigma^m_{i=1} n_i\alpha_i \quad\text{and}\quad M\beta s = \Sigma^m_{i=1} n_i\beta_is_i \qquad (4)$$

we have the following formula

$$T_{comm} = M(\alpha t^{lan} + \beta s / t^{bw}) \qquad (5)$$

Numbers α and β are called the latency constant and the bandwidth constant, respectively. The interconnect now can be viewed as having an *aggregate* latency of αt^{lan} and an *aggregate* bandwidth of t^{bw}/β. For a given cluster, its ping-pong latency and bandwidth, t^{lan} and t^{bw}, can be measured. The number of messages and the average message size in each of the N processors can also be measured. For a relatively balanced N-processor job, as discussed here in this paper, the number of messages, M, and the average message size, s, in the "maximal" processor can be approximated by averaging numbers of messages and average message sizes of the N processors. If the latency and bandwidth constants, α and β, or equivalently the *aggregate* latency and the *aggregate* bandwidth, can be determined, then formula (5) will allow one to obtain the communication cost T_{comm}.

To determine the two numbers α and β, assume all jobs are done on two different clusters, which comprise of the same number and the same kind of processors, but of two different interconnects, *a* and *b*. The two clusters are named as clusters *a* and *b*, respectively; their ping-pong latencies are denoted as t^{lan}_a and t^{lan}_b, respectively; and so are their ping-pong bandwidths as t^{bw}_a and t^{bw}_b. With such two clusters, then it can be conjectured that the two numbers, α and β, in formula (4) remain the same, from runs to runs, with different numbers of processors and the same clusters *a* and *b*. The conjecture is good because all decompositions and hence message patterns are similar. With this conjecture, the two numbers, α and β, can then be determined by the method of least square errors.

Because two jobs, with clusters *a* and *b*, of the same number of processors and precision have identical message patterns, they have the same number of messages and the same average message size. Let the number of messages and the average message size be denoted as M_n and s_n, respectively. Since the decompositions of the two jobs are identical, their computation costs T_{comput} are equal. Denoting further the elapsed times with clusters *a* and *b* as $T^a_{elapsed}$ and $T^b_{elapsed}$, respectively, we have, from formulas (1) and (5),

$$M_n(t^{lan}_a - t^{lan}_b)\alpha + M_ns_n(1/ t^{bw}_a - 1/ t^{bw}_b)\beta = T^a_{elapsed} - T^b_{elapsed} \qquad (6)$$

When applying to measured data, formula (6) is only approximately correct and forms the base for obtaining the least square errors. In formula (6), let the two elapsed times on the right-hand side be substituted with the measured ones, and let the error be defined as the difference between the right-hand side and the left-hand side. Furthermore, let several pairs of same number-of-processor jobs, with the number of processors, n, varying, be measured. Each pair of such jobs produces an error. The sum of squares of those errors is a quadratic function of the two variables, α and β, and the solution that minimizes the quadratic function, which can be easily solved, is known to be the best approximation under the criterion of least square errors.

MODEL, MACHINE, INTERCONNECTS, MEASURED DATA

Model, Machine, and OS
In this paper, the well-known car crash model, refined Neon, of 535 thousand elements and with simulation time of 30 milliseconds, is used. Both single- and double-precision 960.1647 versions of MPP LS-DYNA, with the default decomposition, are used. A 32-processor cluster, consisted of 16 machines of HP's 900MHz rx2600 with HP-UX 11.22, is used. The rx2600 is a 2-CPU Itanium2 machine.

Interconnects and Their Characteristics
Two interconnects are used: the Gigabit Ethernet (GigE) and HP's Hyperfabric 2 (HF2). Its ping-pong latency and bandwidth have been measured and are shown in Table 1.

Elapsed times

Table 2 and Figure 1 show elapsed times, actually measured, for jobs with numbers of processors 1, 2, 4, 8, 16, and 32; and each with the four cases: single precision, GigE; single precision, HF2; double precision, GigE; double precision, HF2.

Message Patterns

Table 3 shows the measured average numbers of messages and average message sizes per processor, with numbers of processors 4, 8, 16, and 32, and with single and double precisions. Furthermore, it has been found that messages for all those jobs are concentrated within a small range of small message sizes. Figures 2 and 3 show such a concentration of small messages for the 32-processor, single-precision job. Such a concentration makes formula (6) a better approximation, because its formulation would be accurate if all messages were of the same type and the same size.

	GigE	HF2
Latency	43 μsec	22 μsec
Bandwidth	112 MB/sec	216 MB/sec

Table 1. Ping-pong latency and bandwidth of Gigabit Ethernet and HF2

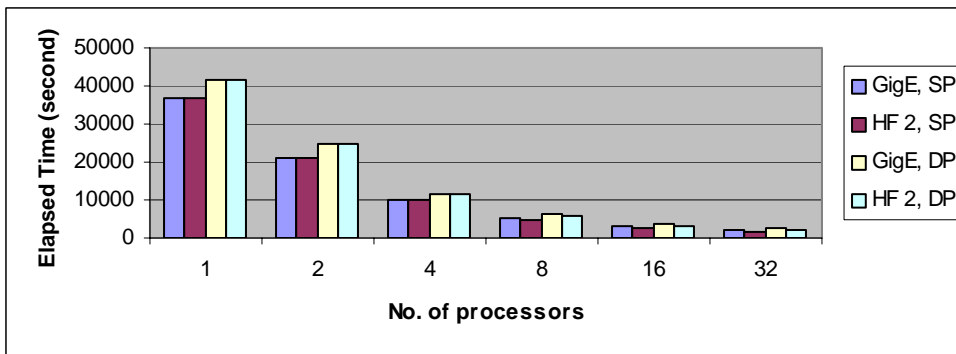| No. of processors / Interconnect, Precision	1	2	4	8	16	32
GigE, SP	37010	21065	9926	5108	2963	2094
HF 2, SP	37010	21065	9913	4998	2800	1799
GigE, DP	41407	24484	11827	6215	3582	2441
HF 2, DP	41407	24484	11703	6024	3332	2119

Table 2. Elapsed times, in seconds, measured

Figure 1. Graph for table 2

No. of Processors	4	8	16	32
Ave. No. of Messages, SP	1232174	1760433	2419095	3684285
Ave. No. of Messages, DP	1231635	1760515	2419646	3683544
Ave. Message Size in Bytes, SP	1707	1044	703	445
Ave. Message Size in Bytes, DP	3360	2042	1368	866

Table 3. Average numbers of messages per processor and averages message sizes for single-precision and double-precision jobs with different numbers of processors.
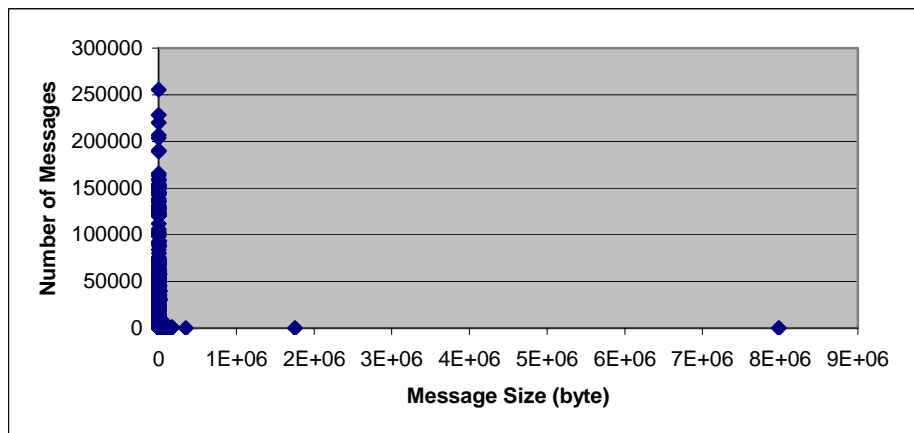
Figure 2. Distributions of all message sizes in the 32-processor, single-precision job

ESTIMATION OF COMMUNICATION COSTS

Aggregate Latency and Aggregate Bandwidth
To estimate α and β, call the cluster with GigE as cluster *a* and the one with HF2 as cluster *b*. Then, two jobs—one from cluster *a*, the other from cluster *b*—with the same number of processors and the same arithmetic precision form a pair of jobs, as described in the **INTRODUCTION** section. With numbers of processors being 4, 8, 16, and 32, and with arithmetic precisions being single and double, there are 8 such pairs of jobs. The 8 errors, as derived from formula (6), for these 8 pairs of jobs, can then be obtained with the ping-pong latency and bandwidth in Table 1, the elapsed time data in Table 2, and the message data in Table 3. The sum of squares of these 8 errors is a quadratic function of α and β. The minimum of the quadratic function occurs when its partial derivatives with respect to α and β are equal to zero, which in turn forms two linear equations of the two unknowns α and β. A computer program based on this approach has been written to obtain:

$$\alpha = 3.55 \text{ and } \beta = 1.90$$

This means that the *aggregate* latency of a given interconnect for the Neon model is 3.55 times its ping-pong latency, and its *aggregate* bandwidth is 0.526, or 1/1.90, times its ping-pong bandwidth.
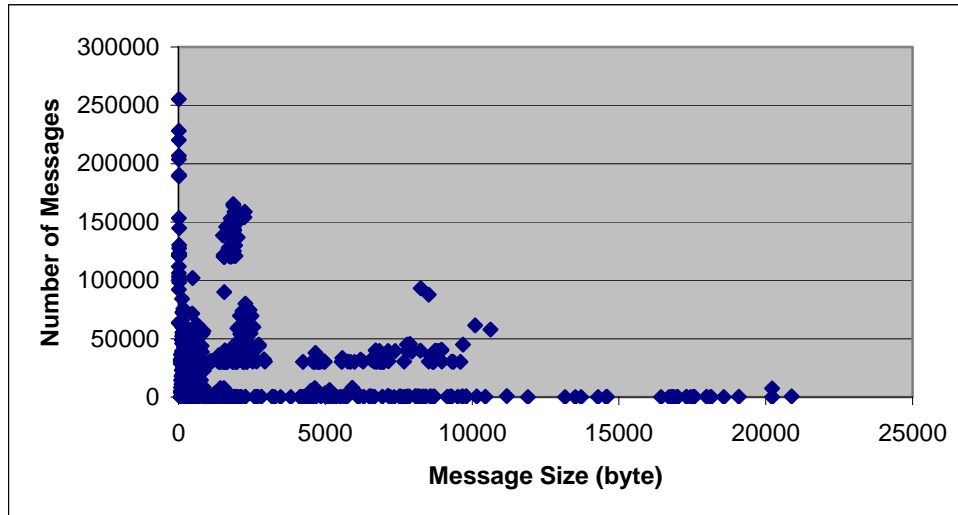
Figure 3. Distribution of message sizes, in the range of 0 to 25,000 bytes, in the same job as Figure 2

Number of Processors	4	8	16	32
Single Precision				
Computation Cost	9750	4826	2580	1500
Communication Cost/GigE	176	282	383	594
Communication Cost/HF2	163	172	220	299
Double Precision				
Computation Cost	11570	5870	3135	1814
Communication Cost/GigE	257	345	447	627
Communication Cost/HF2	133	154	197	305

Table 4. Estimated computation and communication costs in seconds

Number of Processors	4	8	16	32
Single Precision				
Latency/GigE	84%	90%	93%	95%
Bandwidth/GigE	16%	10%	7%	5%
Latency/HF2	84%	89%	93%	95%
Bandwidth/HF2	16%	11%	7%	5%
Double Precision				
Latency/GigE	73%	81%	87%	91%
Bandwidth/GigE	27%	19%	13%	9%
Latency/HF2	73%	81%	87%	91%
Bandwidth/HF2	27%	19%	13%	9%

Table 5. Percentages of latency and bandwidth costs in estimated communication costs
Estimates of Communication and Computation Costs
With the latency constant α and the bandwidth constant β determined, we can then use formula (5) to estimate the communication cost T_{comm}, and hence the computation cost T_{comput}, using formula (1); the result is shown in Table 4, and the percentages of costs due to latency and bandwidth in the estimated communication costs are shown in Table 5. Notice that while the cost percentage of bandwidth decreases,

the cost percentage of latency increases, as the number of processors increases. This result is consistent with the fact the number of messages in MPP LS-DYNA increases with the number of processors.

Estimates of Elapsed Times for Various Cases
Further, we now have the capability to predict the elapsed times with interconnects of known characteristics. For example, shown in Table 6 and Figure 4 are estimated elapsed times for the 5 single-precision cases:

1. An interconnect of infinite speed, i.e., zero latency and infinite bandwidth
2. An interconnect with the same latency as that of HF2 and with infinite bandwidth
3. An interconnect with the same latency as that of HF2 and with bandwidth doubled
4. An interconnect with the same bandwidth as that of the HF2 and zero latency
5. An interconnect with the same bandwidth as that of the HF2 and latency halved

And shown in Table 7 and Figure 5 are the 5 corresponding double-precision cases. The maximal speedups are obtained with an interconnect of infinite speed, i.e., with Case 1. Table 8 shows these maximal speedups for both single and double precision; for example, it predicts that the maximal speedup for the single, precision, 32-processor job is 24.67.

Number of Processors	4	8	16	32
Measured	9913	4998	2800	1799
Infinite Speed	9750	4826	2580	1500
Infinite Bandwidth	9846	4964	2769	1788
Bandwidth Doubled	9856	4972	2777	1795
Zero Latency	9769	4842	2595	1515
Latency Halved	9817	4911	2690	1659

Table 6. Measured elapsed times and estimated elapsed times in seconds for the 5 single-precision cases: infinite-speed interconnect, HF2 with infinite bandwidth, HF2 with bandwidth doubled, HF2 with zero latency, HF2 with latency halved.
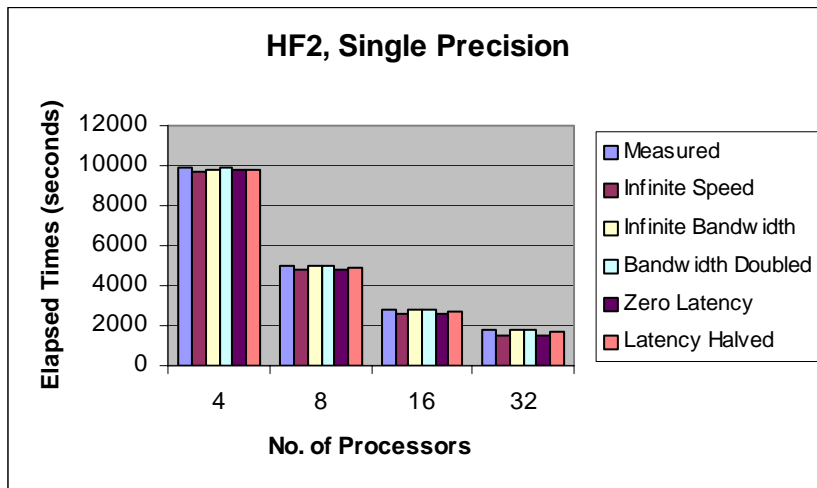

Figure 4. Graph for Table 6

Number of Processors	4	8	16	32
Measured	11703	6024	3332	2119
Infinite Speed	11570	5870	3135	1814
Infinite Bandwidth	11666	6007	3324	2102
Bandwidth Doubled	11684	6023	3339	2116
Zero Latency	11606	5902	3164	1842
Latency Halved	11654	5970	3259	1986

Table 7. Measured elapsed times and estimated elapsed times in seconds for the 5 double –precision cases: infinite-speed interconnect, HF2 with infinite bandwidth, HF2 with bandwidth doubled, HF2 with zero latency, HF2 with latency halved.
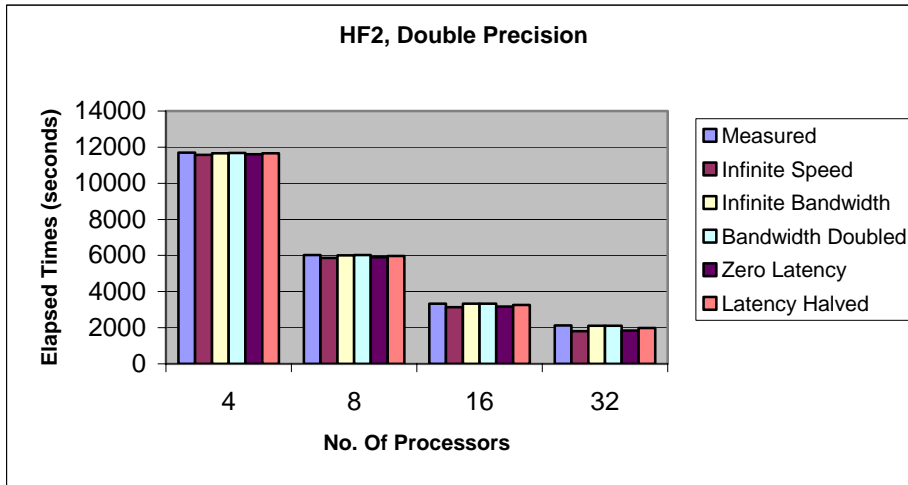

Figure 5. Graph for Table 7

Number of Processors	4	8	16	32
Maximal Speedup, SP	3.80	7.67	14.34	24.67
Maximal Speedup, DP	3.58	7.05	13.21	22.83

Table 8. Maximal speedups for single-precision and double precision jobs

SUMMARY AND CONCLUSIONS

In this paper, a quantitative approach to estimate the communication and the computation costs of an MPP LS-DYNA simulation is presented. The knowledge of the two costs will provide the MPP LS-DYNA user, the software developer and the hardware designer a deep insight into factors that affect the performance of MPP LS-DYNA.

REFERENCES

1. Hennessy, J. L., Patterson, D. A., *Computer Architecture: A Quantitative Approach*, 2nd Edition, 1996, Morgan Kaufmann Publishers, Inc., pp. 565-571.