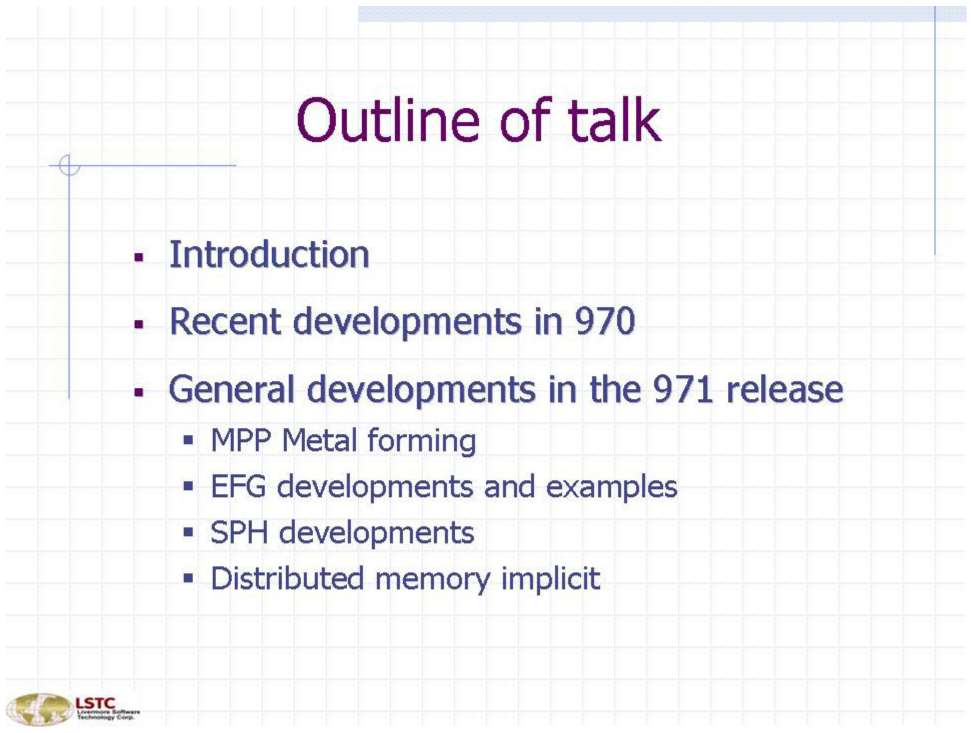
Current and Future Developments of LS-DYNA

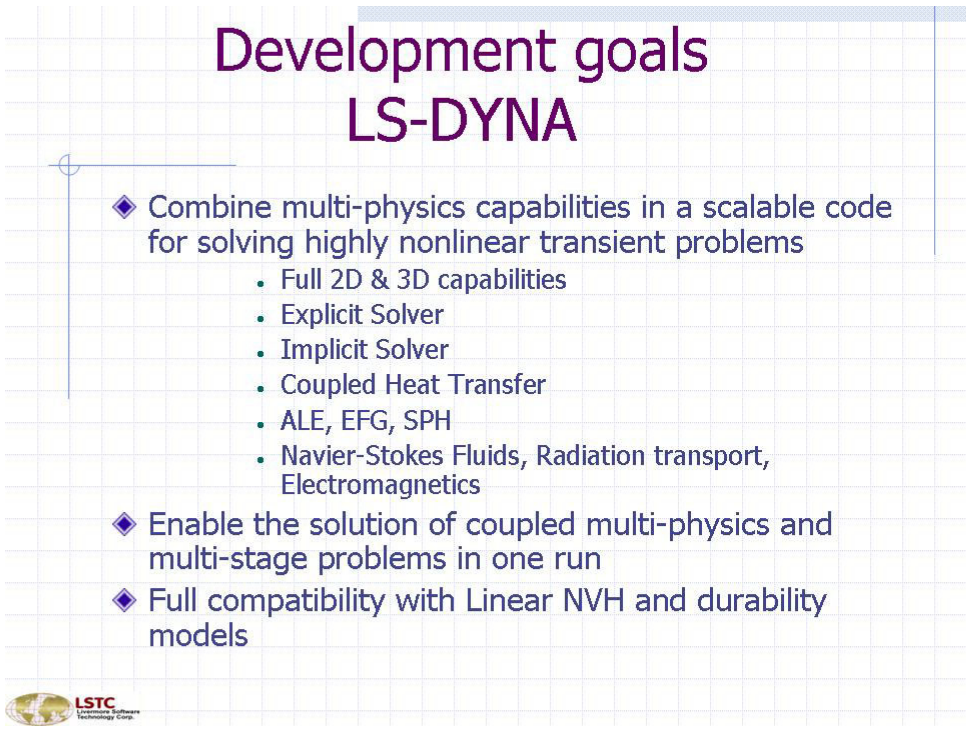
Authors:

Dr J. O. Hallquist
Livermore Software Technology Corporation



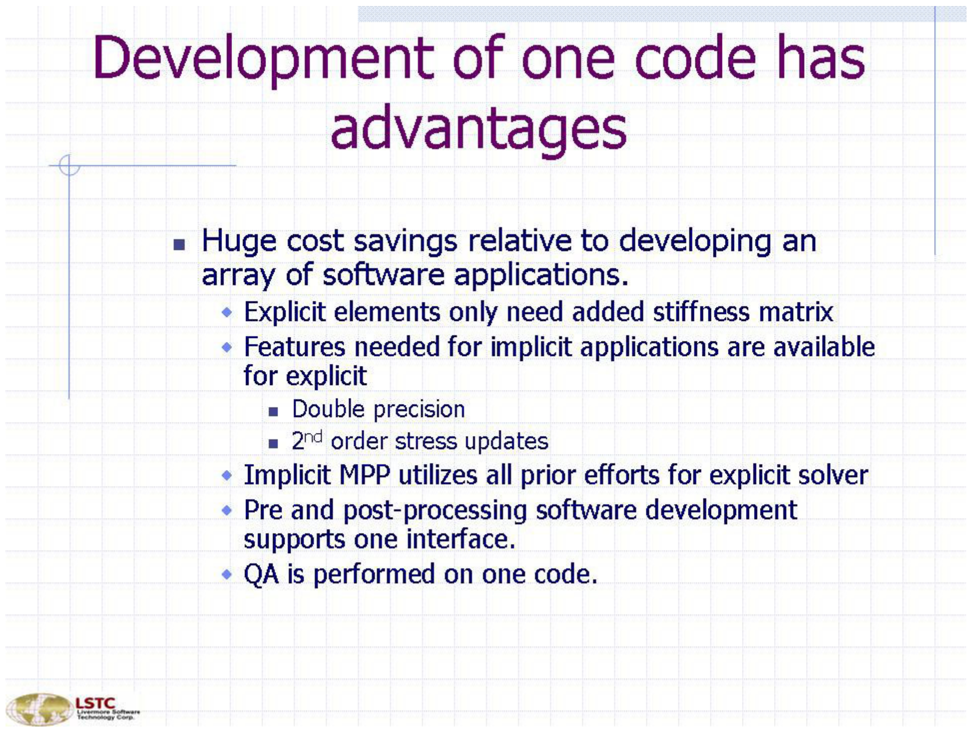
Outline of talk

- Introduction
- Recent developments in 970
- General developments in the 971 release
  - MPP Metal forming
  - EFG developments and examples
  - SPH developments
  - Distributed memory implicit



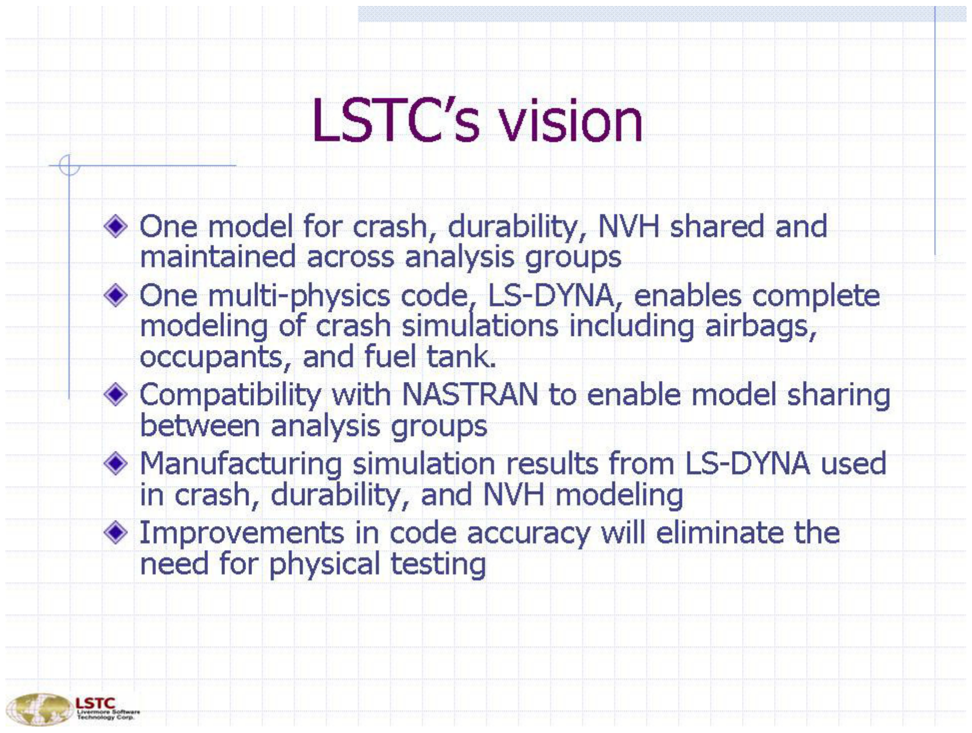
Development goals LS-DYNA

- Combine multi-physics capabilities in a scalable code for solving highly nonlinear transient problems
  - Full 2D & 3D capabilities
  - Explicit Solver
  - Implicit Solver
  - Coupled Heat Transfer
  - ALE, EFG, SPH
  - Navier-Stokes Fluids, Radiation transport, Electromagnetics
- Enable the solution of coupled multi-physics and multi-stage problems in one run
- Full compatibility with Linear NVH and durability models



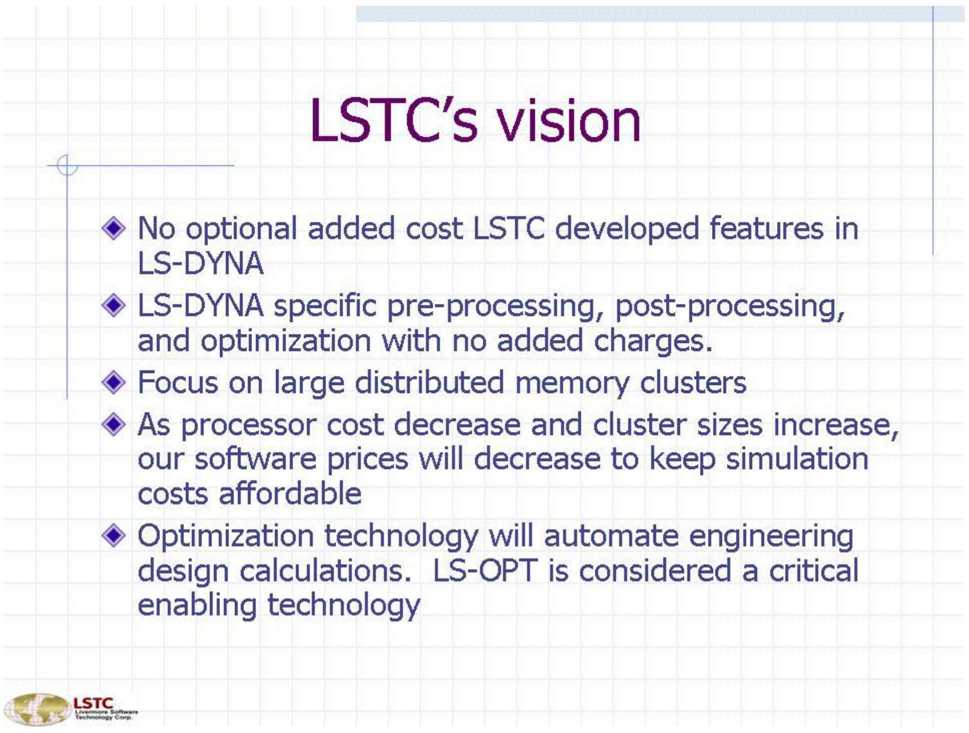
Development of one code has advantages

- Huge cost savings relative to developing an array of software applications.
  - Explicit elements only need added stiffness matrix
  - Features needed for implicit applications are available for explicit
    - Double precision
    - 2nd order stress updates
  - Implicit MPP utilizes all prior efforts for explicit solver
  - Pre and post-processing software development supports one interface.
  - QA is performed on one code.



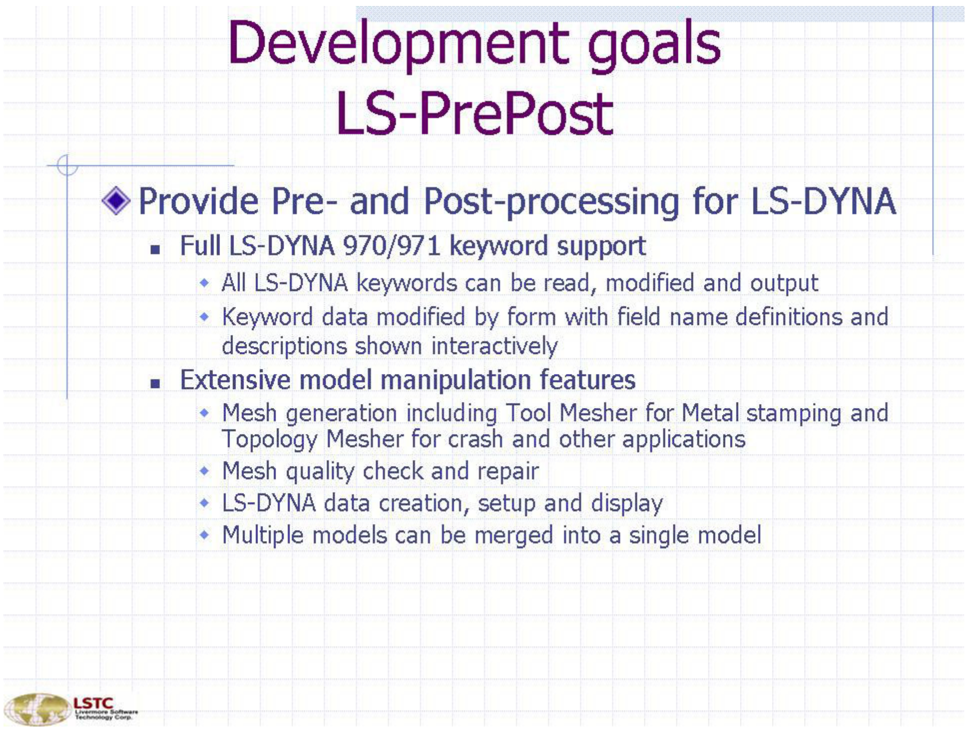
LSTC's vision

- One model for crash, durability, NVH shared and maintained across analysis groups
- One multi-physics code, LS-DYNA, enables complete modeling of crash simulations including airbags, occupants, and fuel tank.
- Compatibility with NASTRAN to enable model sharing between analysis groups
- Manufacturing simulation results from LS-DYNA used in crash, durability, and NVH modeling
- Improvements in code accuracy will eliminate the need for physical testing



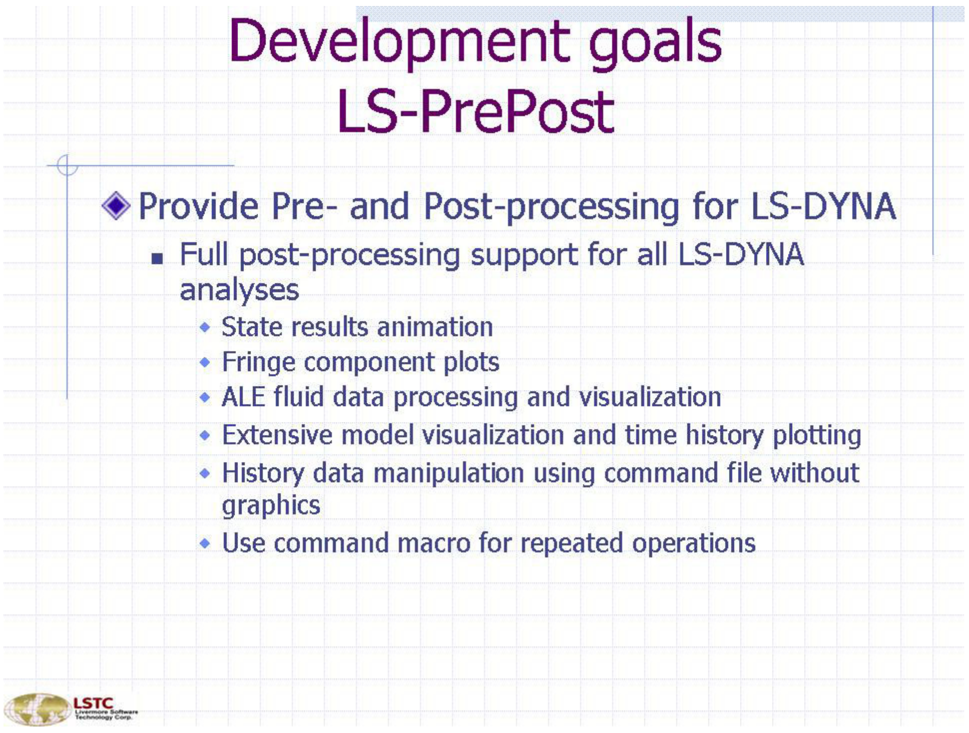
LSTC's vision

- No optional added cost LSTC developed features in LS-DYNA
- LS-DYNA specific pre-processing, post-processing, and optimization with no added charges.
- Focus on large distributed memory clusters
- As processor cost decrease and cluster sizes increase, our software prices will decrease to keep simulation costs affordable
- Optimization technology will automate engineering design calculations. LS-OPT is considered a critical enabling technology



Development goals LS-PrePost

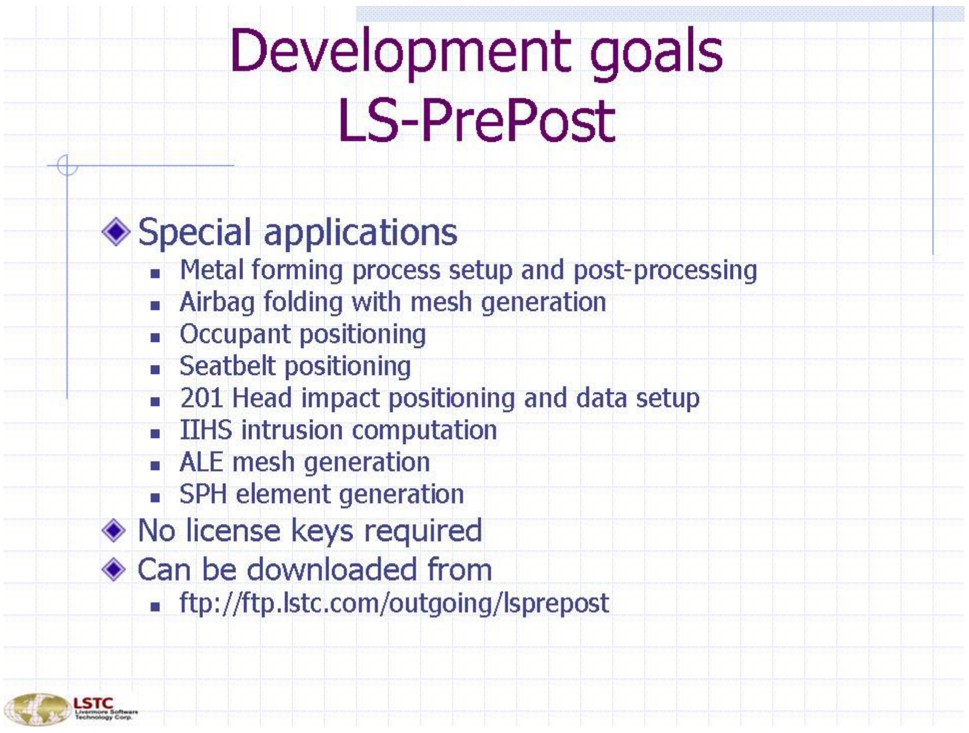
- Provide Pre- and Post-processing for LS-DYNA
  - Full LS-DYNA 970/971 keyword support
    - All LS-DYNA keywords can be read, modified and output
    - Keyword data modified by form with field name definitions and descriptions shown interactively
  - Extensive model manipulation features
    - Mesh generation including Tool Mesher for Metal stamping and Topology Mesher for crash and other applications
    - Mesh quality check and repair
    - LS-DYNA data creation, setup and display
    - Multiple models can be merged into a single model



Development goals LS-PrePost

- Provide Pre- and Post-processing for LS-DYNA
  - Full post-processing support for all LS-DYNA analyses
    - State results animation
    - Fringe component plots
    - ALE fluid data processing and visualization
    - Extensive model visualization and time history plotting
    - History data manipulation using command file without graphics
    - Use command macro for repeated operations

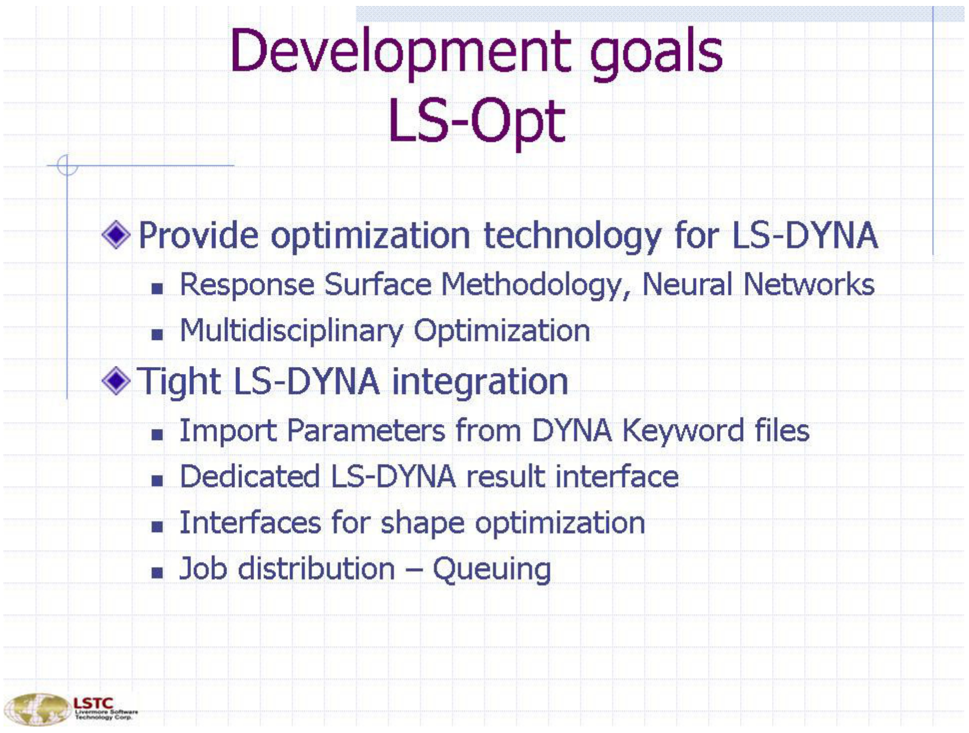
LSTC Livermore Software Technology Corp.



Development goals LS-PrePost

- Special applications
  - Metal forming process setup and post-processing
  - Airbag folding with mesh generation
  - Occupant positioning
  - Seatbelt positioning
  - 201 Head impact positioning and data setup
  - IIHS intrusion computation
  - ALE mesh generation
  - SPH element generation
- No license keys required
- Can be downloaded from
  - ftp://ftp.lstc.com/outgoing/lsprepost

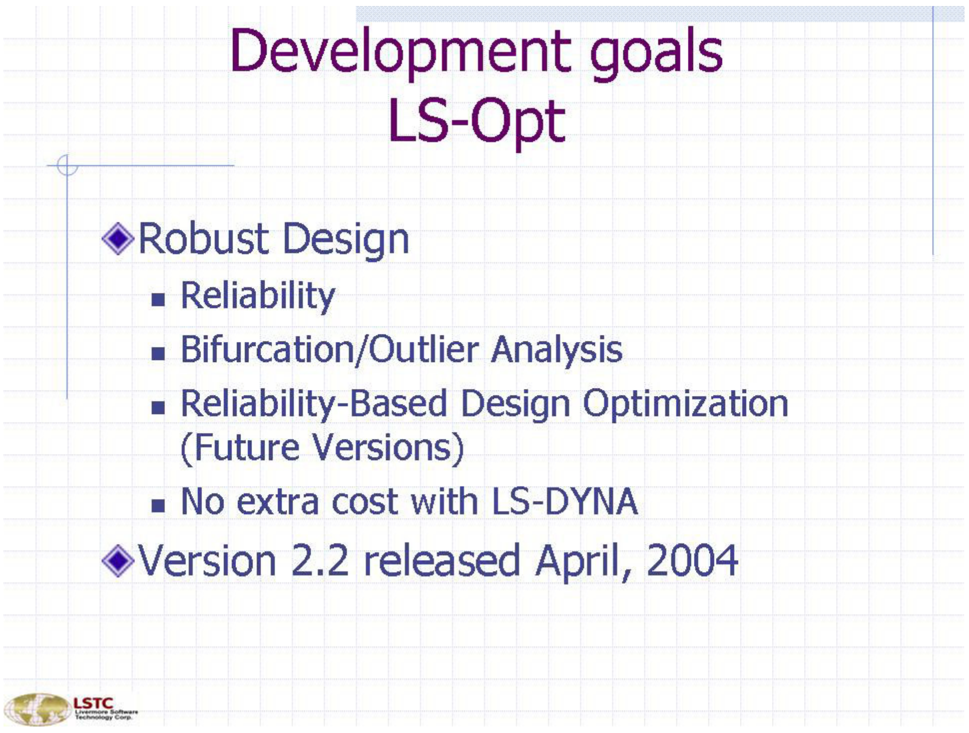
LSTC Livermore Software Technology Corp.



Development goals LS-Opt

- Provide optimization technology for LS-DYNA
  - Response Surface Methodology, Neural Networks
  - Multidisciplinary Optimization
- Tight LS-DYNA integration
  - Import Parameters from DYNA Keyword files
  - Dedicated LS-DYNA result interface
  - Interfaces for shape optimization
  - Job distribution – Queuing

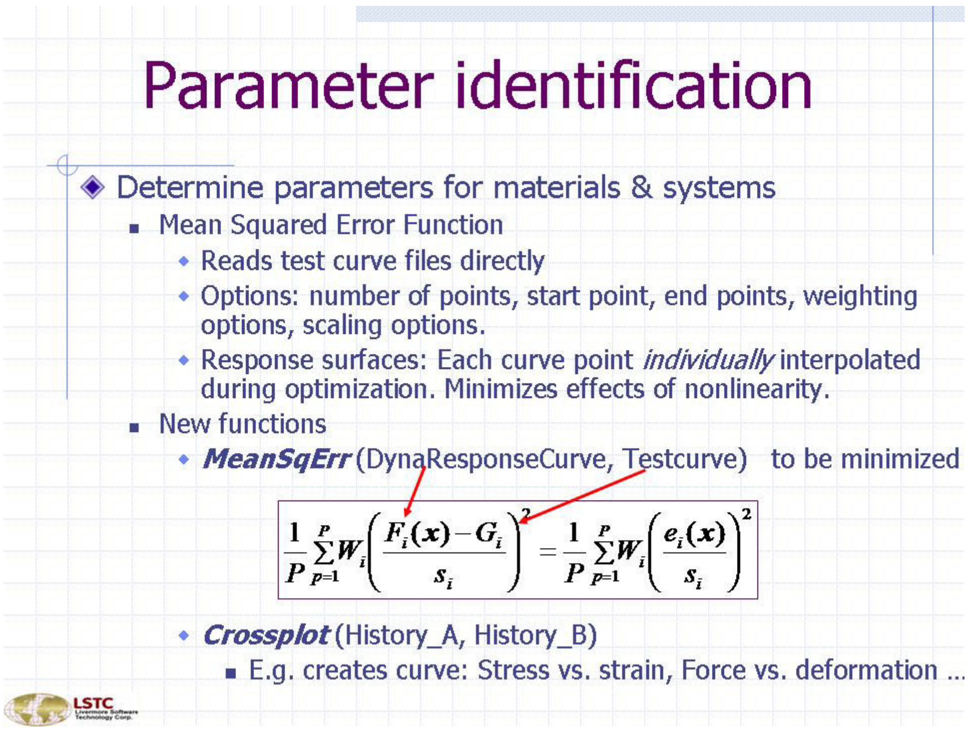
LSTC Livermore Software Technology Corp.



Development goals LS-Opt

- Robust Design
  - Reliability
  - Bifurcation/Outlier Analysis
  - Reliability-Based Design Optimization (Future Versions)
  - No extra cost with LS-DYNA
- Version 2.2 released April, 2004

LSTC Livermore Software Technology Corp.

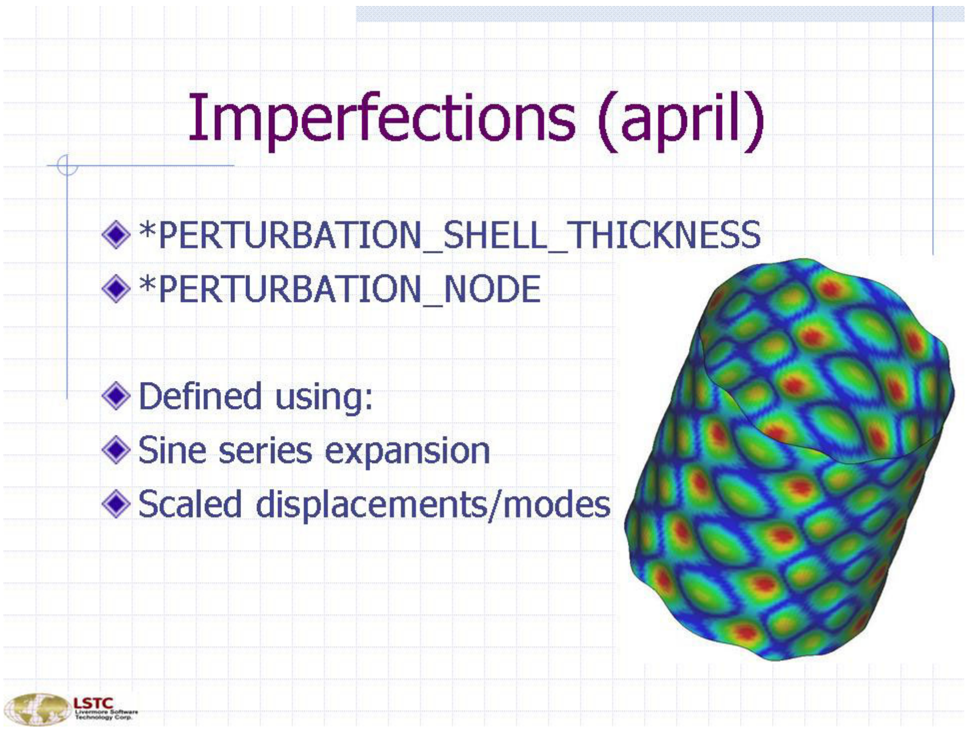


Parameter identification

- Determine parameters for materials & systems
  - Mean Squared Error Function
    - Reads test curve files directly
    - Options: number of points, start point, end points, weighting options, scaling options.
    - Response surfaces: Each curve point *individually* interpolated during optimization. Minimizes effects of nonlinearity.
  - New functions
    - ***MeanSqErr*** (DynaResponseCurve, Testcurve) to be minimized

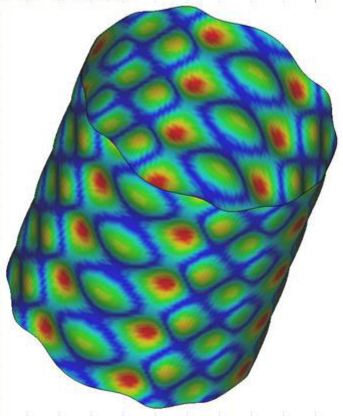
$$\boxed{\frac{1}{P}\sum_{p=1}^{P} W_i\left(\frac{F_i(x)-G_i}{s_i}\right)^2 = \frac{1}{P}\sum_{p=1}^{P} W_i\left(\frac{e_i(x)}{s_i}\right)^2}$$

    - ***Crossplot*** (History_A, History_B)
      - E.g. creates curve: Stress vs. strain, Force vs. deformation …



Imperfections (april)

- *PERTURBATION_SHELL_THICKNESS
- *PERTURBATION_NODE
- Defined using:
- Sine series expansion
- Scaled displacements/modes

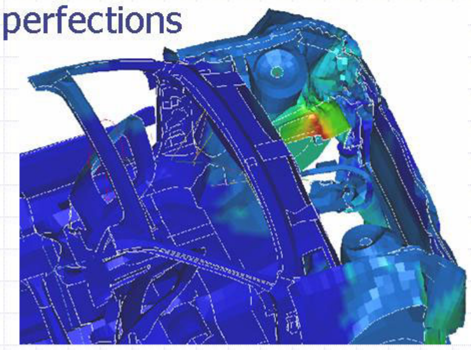


Reliability visualization (fall)

- View reliability information in LS-PREPOST
- Effect of a design variable (DSA)
- Most important design variable
- Zones sensitive to imperfections
- For example:

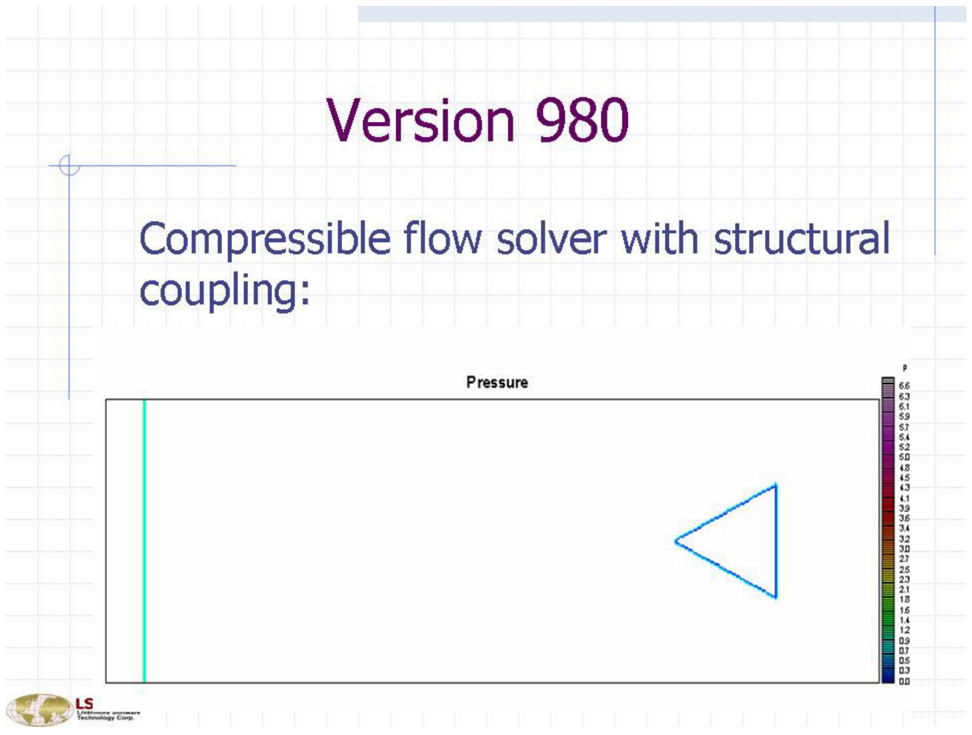
$$std\_dev(\mathrm{x\_disp})$$

$$\frac{d\,\mathrm{FLD}}{d\,x_1}$$



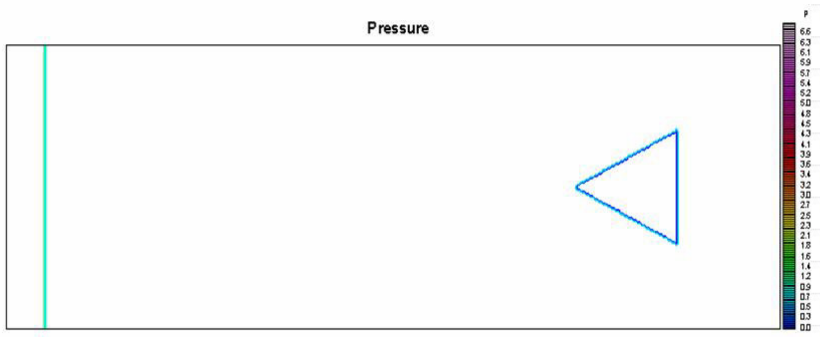
Version 980

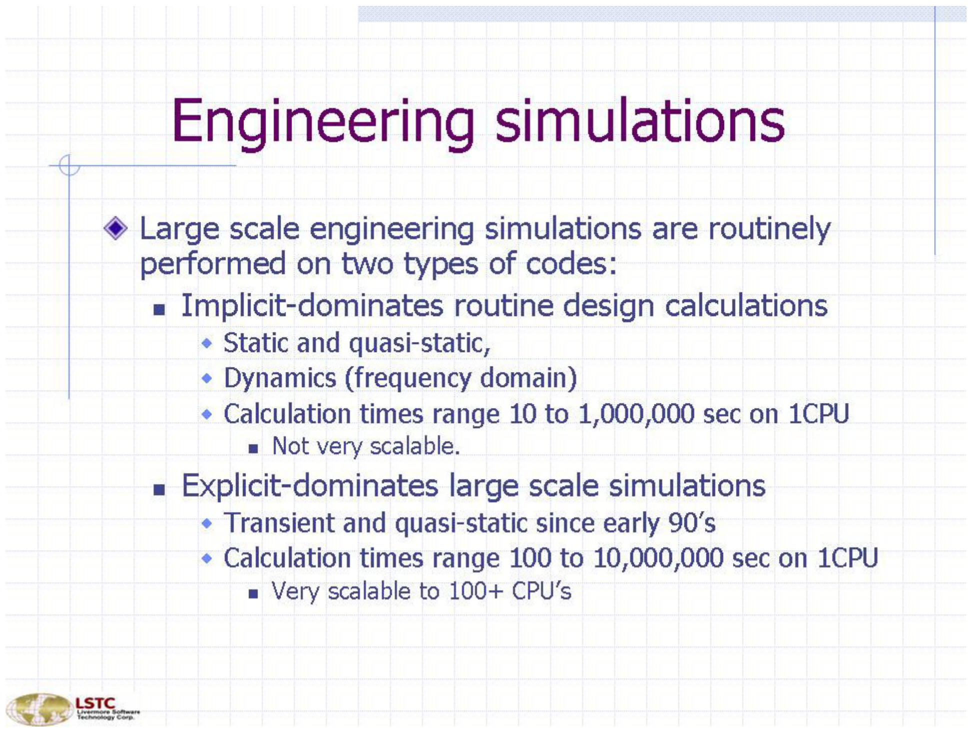
- Version 980 introduces a new code framework to accommodate the addition of electromagnetic-structure coupling and fluid-structure coupling with embedded, scalable CFD solvers.
- New physics:
  - An explicit compressible flow solver based upon the Conservation Element/Solution Element (CE/SE) Method
  - Radiative heat transport through participating media, as well as using exchange factors initially coupled to the incompressible flow solver
  - Solid-fluid heat flow coupling for the incompressible flow solver
  - Electromagnetic FEA coupled to solids for forming applications
- Release anticipated in 2006 or 2007



Version 980

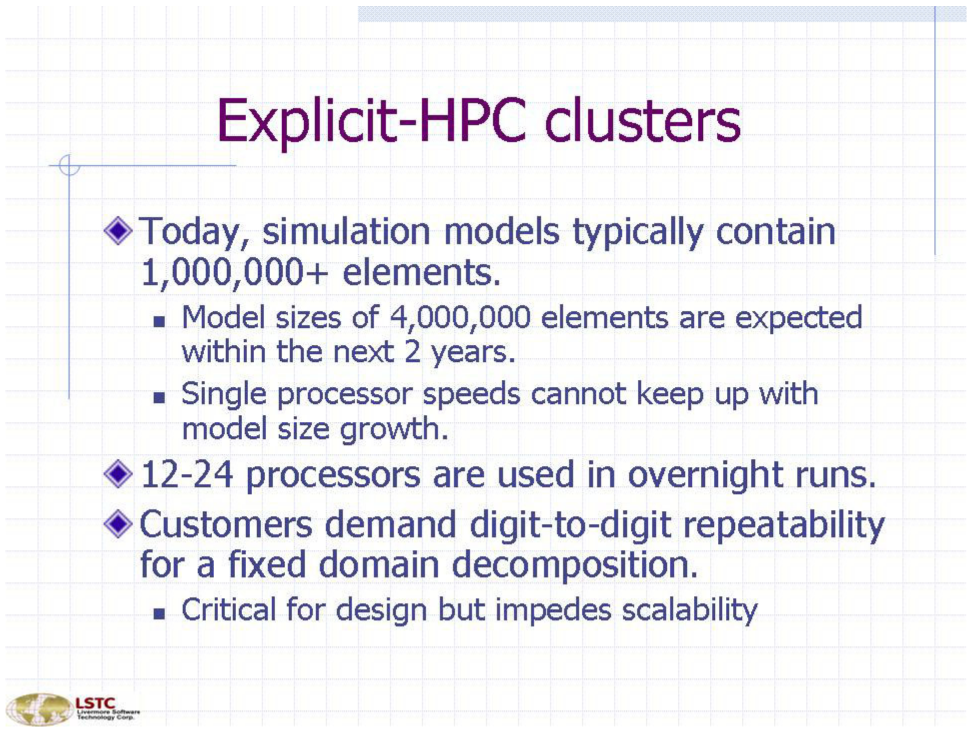
Compressible flow solver with structural coupling:





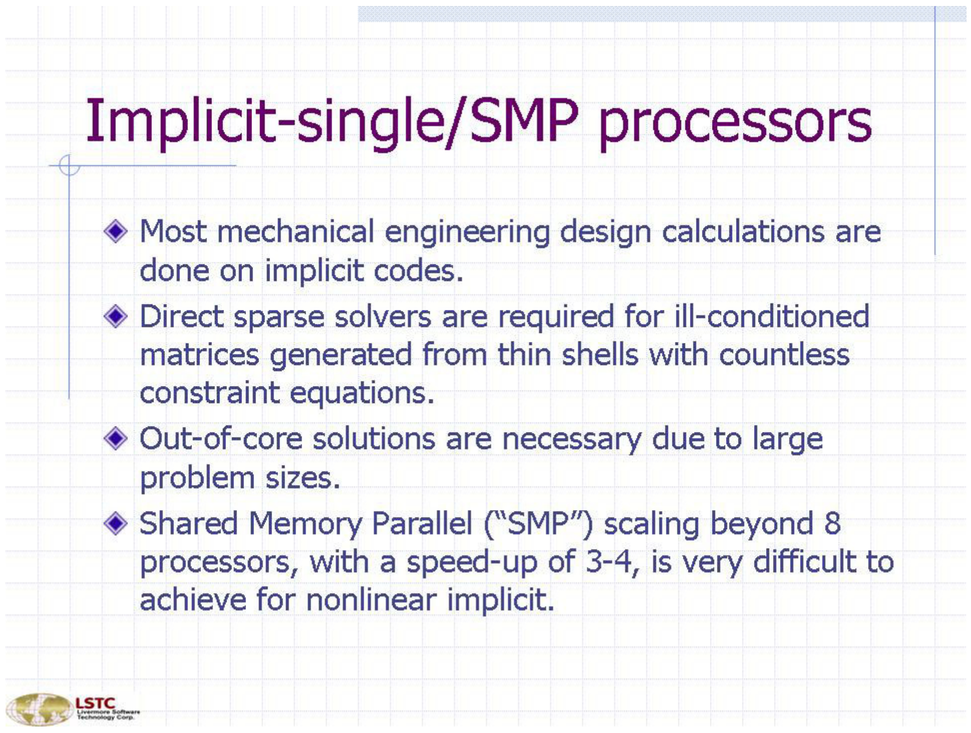
Engineering simulations

- Large scale engineering simulations are routinely performed on two types of codes:
  - Implicit-dominates routine design calculations
    - Static and quasi-static,
    - Dynamics (frequency domain)
    - Calculation times range 10 to 1,000,000 sec on 1CPU
      - Not very scalable.
  - Explicit-dominates large scale simulations
    - Transient and quasi-static since early 90's
    - Calculation times range 100 to 10,000,000 sec on 1CPU
      - Very scalable to 100+ CPU's



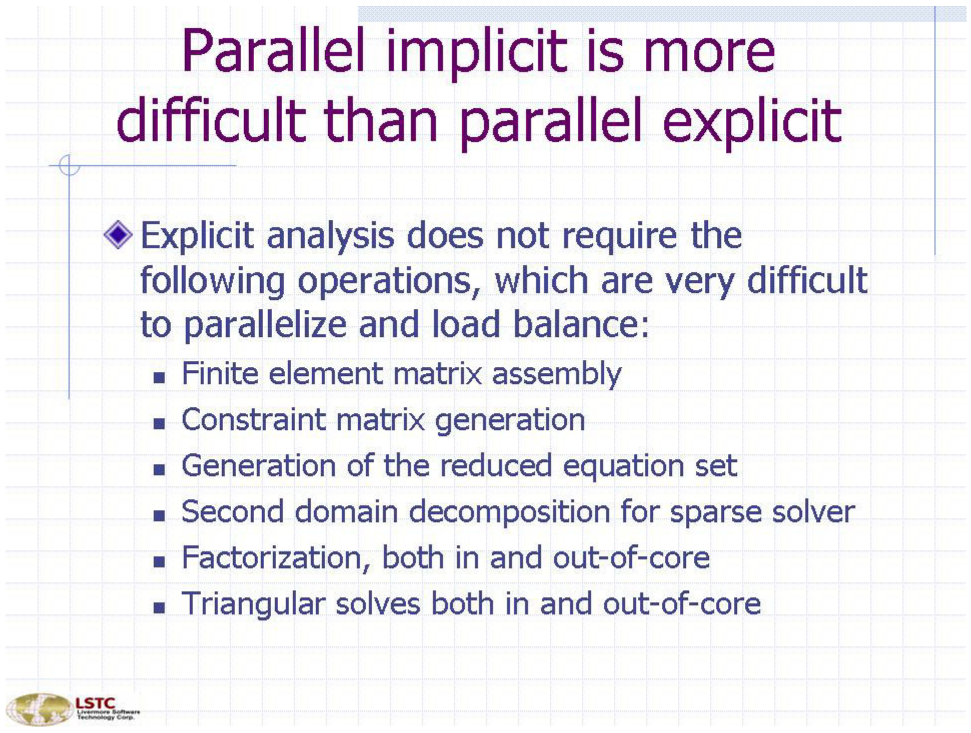
Explicit-HPC clusters

- Today, simulation models typically contain 1,000,000+ elements.
  - Model sizes of 4,000,000 elements are expected within the next 2 years.
  - Single processor speeds cannot keep up with model size growth.
- 12-24 processors are used in overnight runs.
- Customers demand digit-to-digit repeatability for a fixed domain decomposition.
  - Critical for design but impedes scalability



Implicit-single/SMP processors

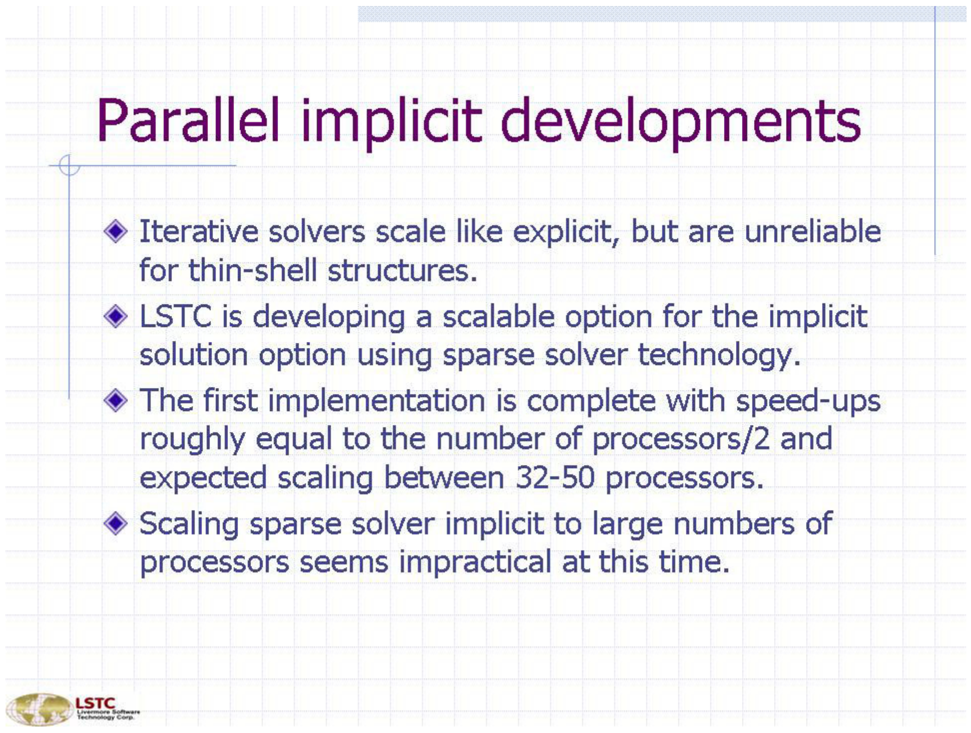
- Most mechanical engineering design calculations are done on implicit codes.
- Direct sparse solvers are required for ill-conditioned matrices generated from thin shells with countless constraint equations.
- Out-of-core solutions are necessary due to large problem sizes.
- Shared Memory Parallel ("SMP") scaling beyond 8 processors, with a speed-up of 3-4, is very difficult to achieve for nonlinear implicit.



Parallel implicit is more difficult than parallel explicit

- Explicit analysis does not require the following operations, which are very difficult to parallelize and load balance:
  - Finite element matrix assembly
  - Constraint matrix generation
  - Generation of the reduced equation set
  - Second domain decomposition for sparse solver
  - Factorization, both in and out-of-core
  - Triangular solves both in and out-of-core

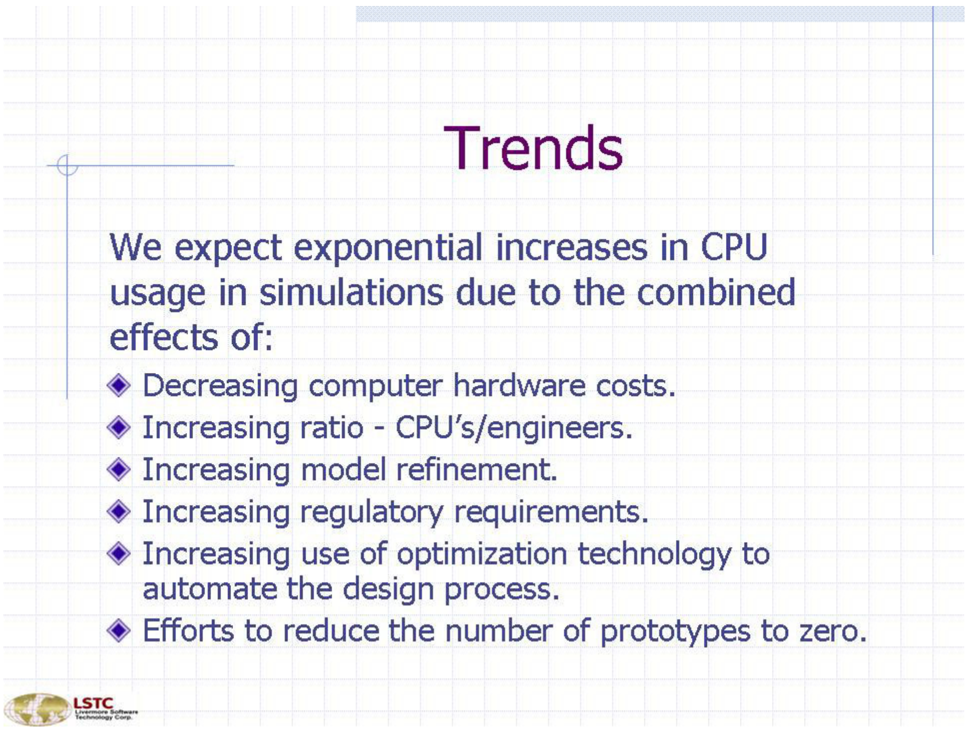
LSTC Livermore Software Technology Corp.



Parallel implicit developments

- Iterative solvers scale like explicit, but are unreliable for thin-shell structures.
- LSTC is developing a scalable option for the implicit solution option using sparse solver technology.
- The first implementation is complete with speed-ups roughly equal to the number of processors/2 and expected scaling between 32-50 processors.
- Scaling sparse solver implicit to large numbers of processors seems impractical at this time.

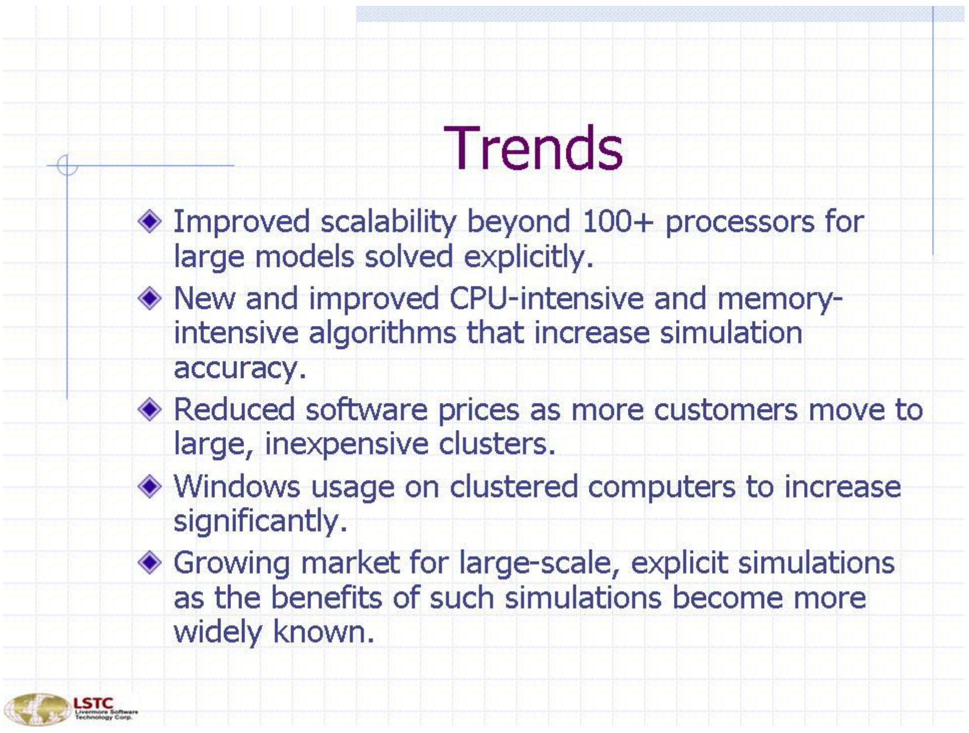
LSTC Livermore Software Technology Corp.



Trends

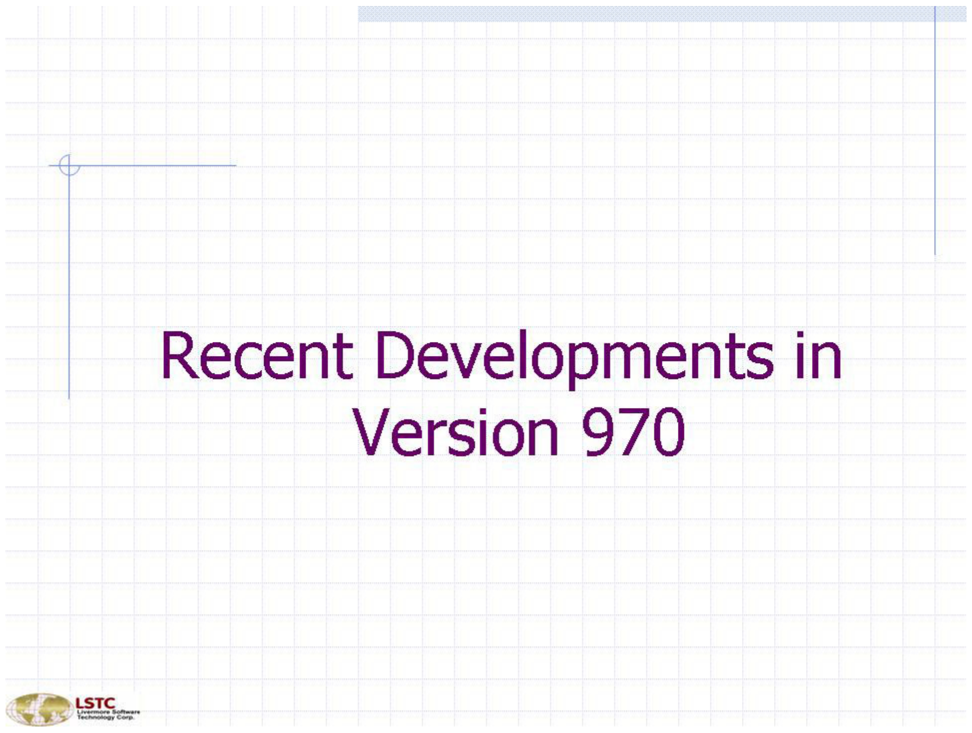
We expect exponential increases in CPU usage in simulations due to the combined effects of:

- Decreasing computer hardware costs.
- Increasing ratio - CPU's/engineers.
- Increasing model refinement.
- Increasing regulatory requirements.
- Increasing use of optimization technology to automate the design process.
- Efforts to reduce the number of prototypes to zero.

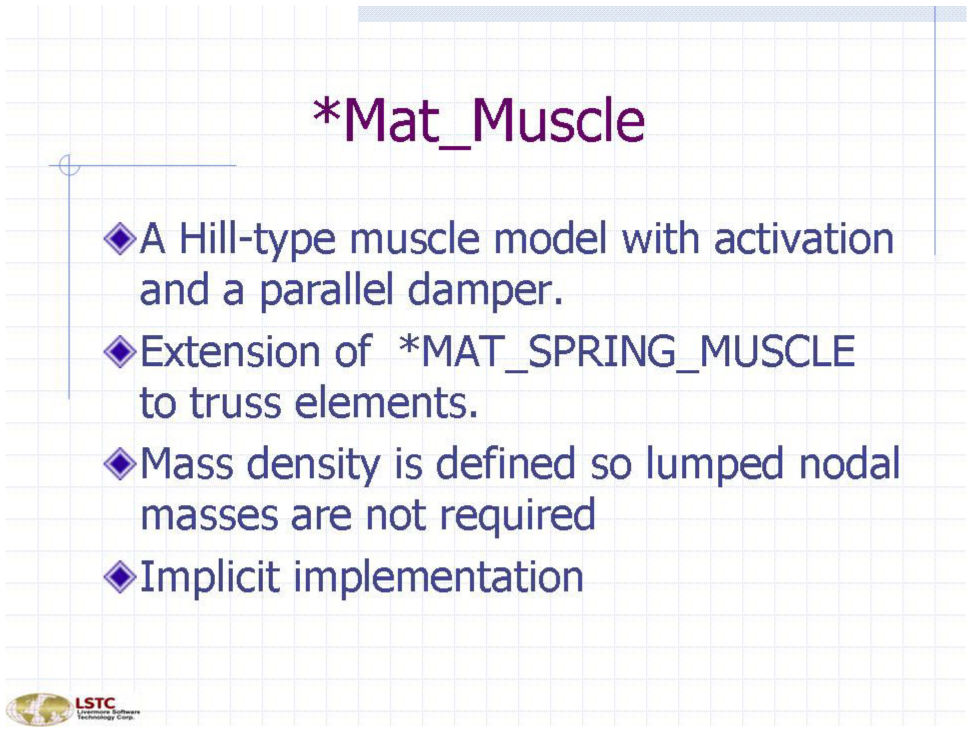


Trends

- Improved scalability beyond 100+ processors for large models solved explicitly.
- New and improved CPU-intensive and memory-intensive algorithms that increase simulation accuracy.
- Reduced software prices as more customers move to large, inexpensive clusters.
- Windows usage on clustered computers to increase significantly.
- Growing market for large-scale, explicit simulations as the benefits of such simulations become more widely known.



Recent Developments in Version 970



*Mat_Muscle

- A Hill-type muscle model with activation and a parallel damper.
- Extension of *MAT_SPRING_MUSCLE to truss elements.
- Mass density is defined so lumped nodal masses are not required
- Implicit implementation

*Mat_simplified_rubber/foam_ with_failure

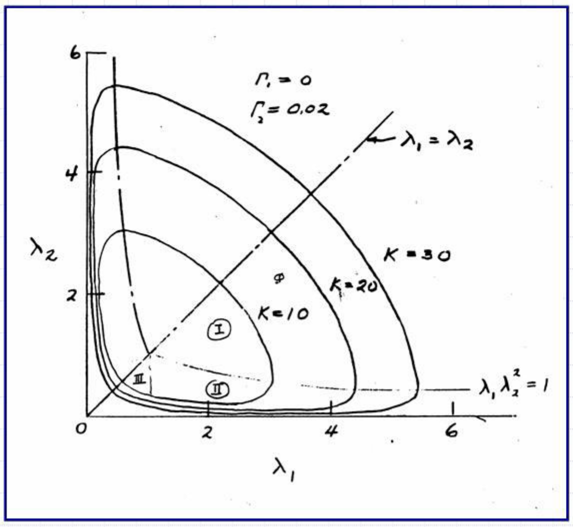
- Failure criterion is defined in terms of the invariants of the right Cauchy-Green deformation tensor:

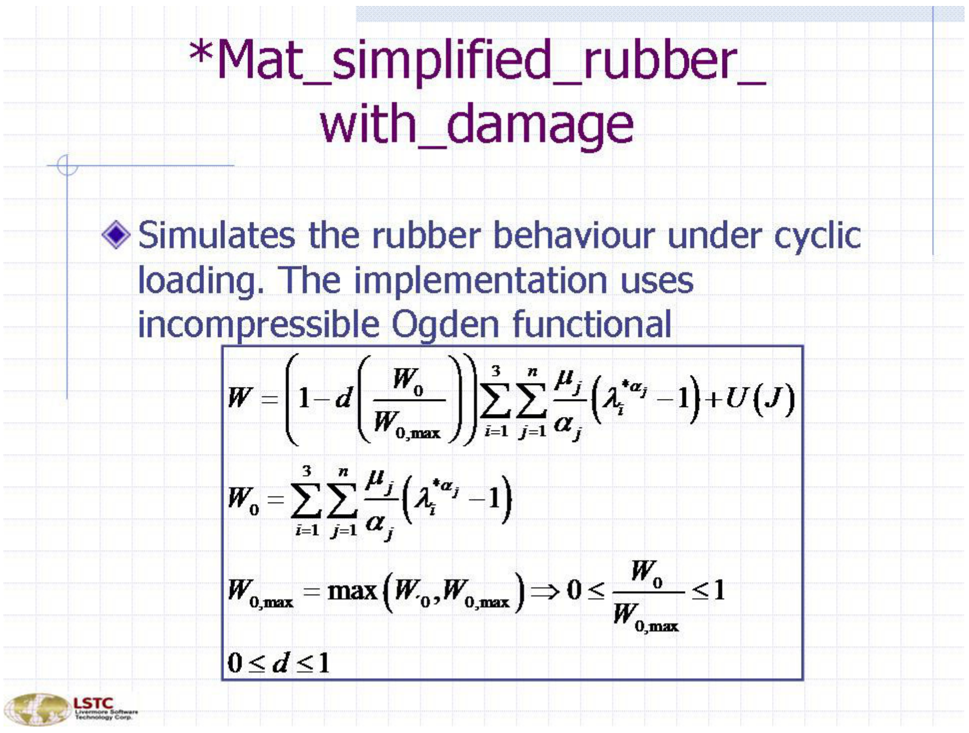
$$f(I_1, I_2, I_3) = (I_1 - 3) + \Gamma_1 (I_1 - 3)^2 + \Gamma_2 (I_2 - 3) = K$$

where *K* is a material parameter which controls the size enclosed by the failure surface

- Works with shell elements

*Mat_simplified_rubber/foam_ with_failure





*Mat_simplified_rubber_ with_damage

- Simulates the rubber behaviour under cyclic loading. The implementation uses incompressible Ogden functional

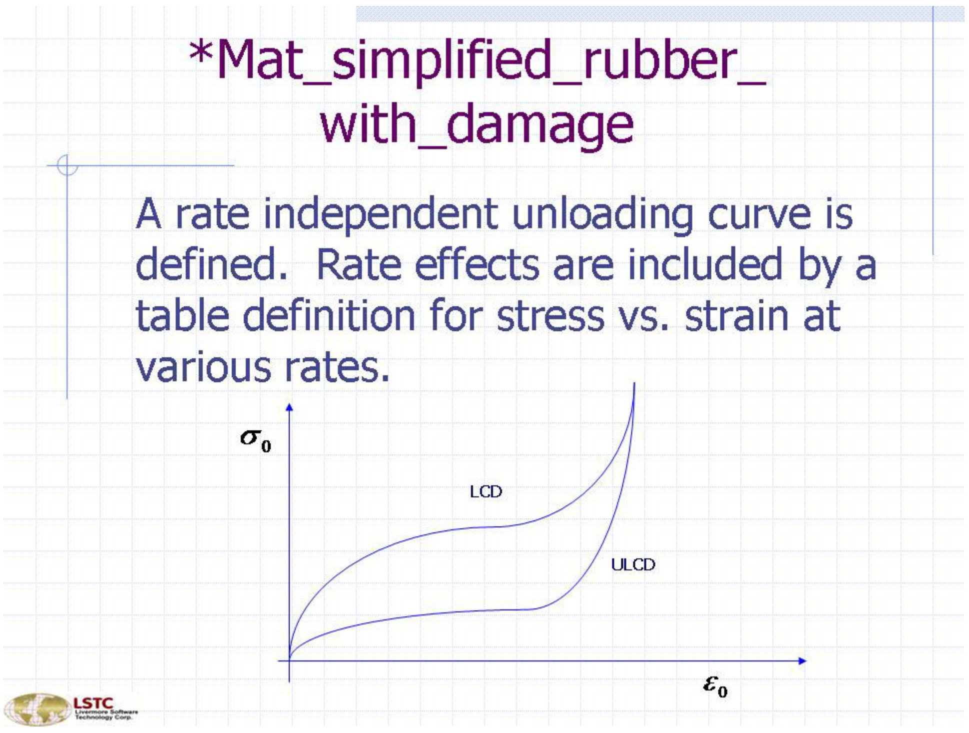
$$W=\left(1-d\left(\frac{W_0}{W_{0,\max}}\right)\right)\sum_{i=1}^{3}\sum_{j=1}^{n}\frac{\mu_j}{\alpha_j}\left(\lambda_i^{*\alpha_j}-1\right)+U(J)$$

$$W_0=\sum_{i=1}^{3}\sum_{j=1}^{n}\frac{\mu_j}{\alpha_j}\left(\lambda_i^{*\alpha_j}-1\right)$$

$$W_{0,\max}=\max\left(W_0,W_{0,\max}\right)\Rightarrow 0\le\frac{W_0}{W_{0,\max}}\le 1$$

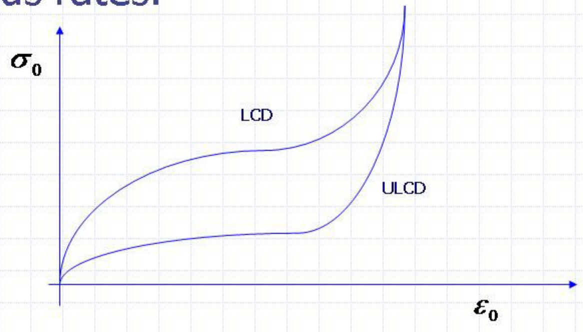
$$0\le d\le 1$$

LSTC Livermore Software Technology Corp.

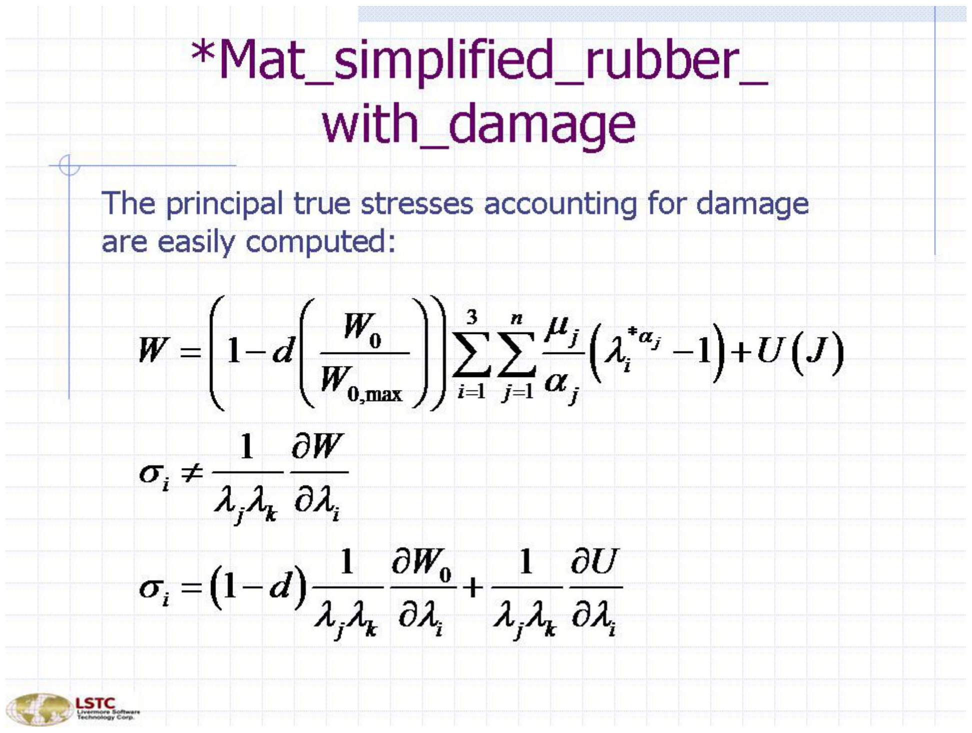


*Mat_simplified_rubber_ with_damage

A rate independent unloading curve is defined. Rate effects are included by a table definition for stress vs. strain at various rates.



LSTC Livermore Software Technology Corp.



*Mat_simplified_rubber_with_damage

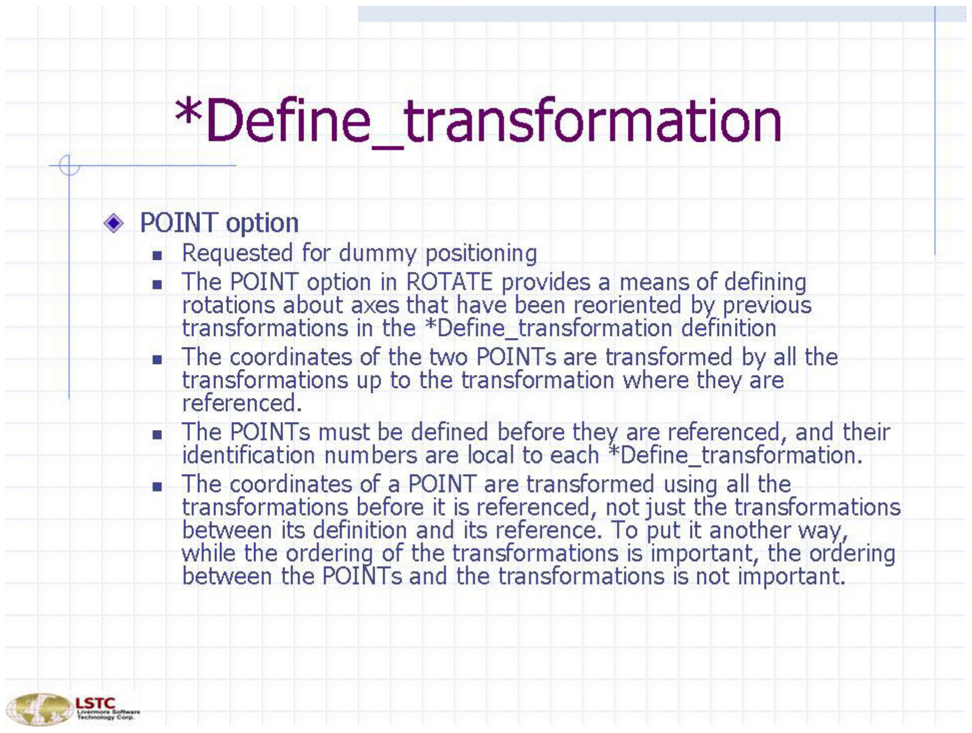
The principal true stresses accounting for damage are easily computed:

$$W=\left(1-d\left(\frac{W_0}{W_{0,\max}}\right)\right)\sum_{i=1}^{3}\sum_{j=1}^{n}\frac{\mu_j}{\alpha_j}\left(\lambda_i^{*\alpha_j}-1\right)+U(J)$$

$$\sigma_i\neq\frac{1}{\lambda_j\lambda_k}\frac{\partial W}{\partial\lambda_i}$$

$$\sigma_i=(1-d)\frac{1}{\lambda_j\lambda_k}\frac{\partial W_0}{\partial\lambda_i}+\frac{1}{\lambda_j\lambda_k}\frac{\partial U}{\partial\lambda_i}$$

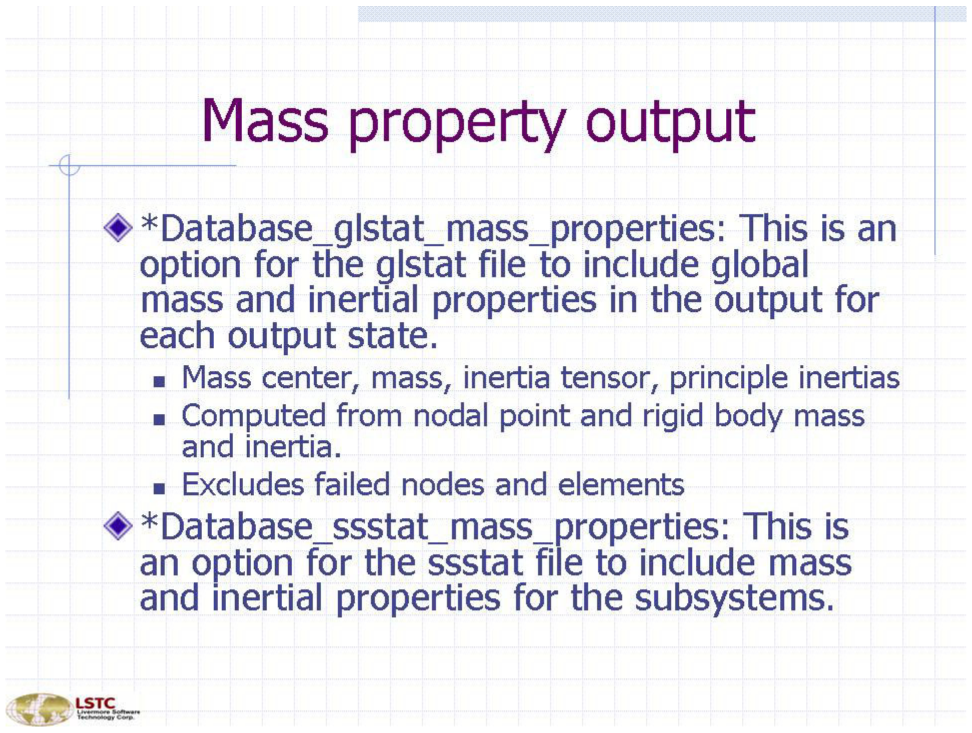
LSTC Livermore Software Technology Corp.



*Define_transformation

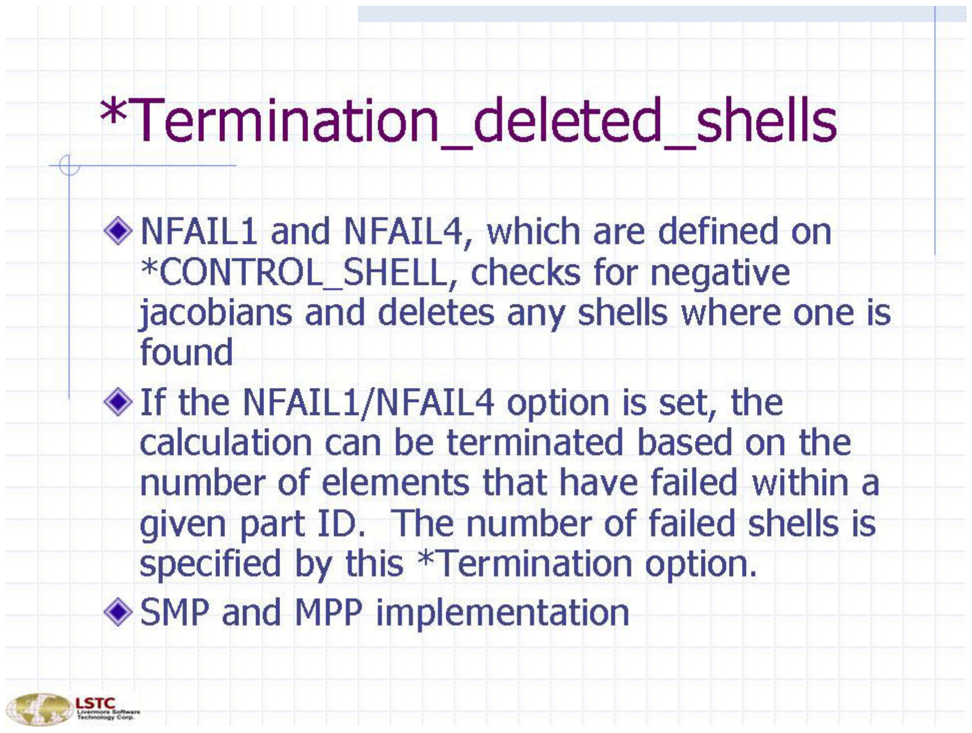
- POINT option
  - Requested for dummy positioning
  - The POINT option in ROTATE provides a means of defining rotations about axes that have been reoriented by previous transformations in the *Define_transformation definition
  - The coordinates of the two POINTs are transformed by all the transformations 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.

LSTC Livermore Software Technology Corp.



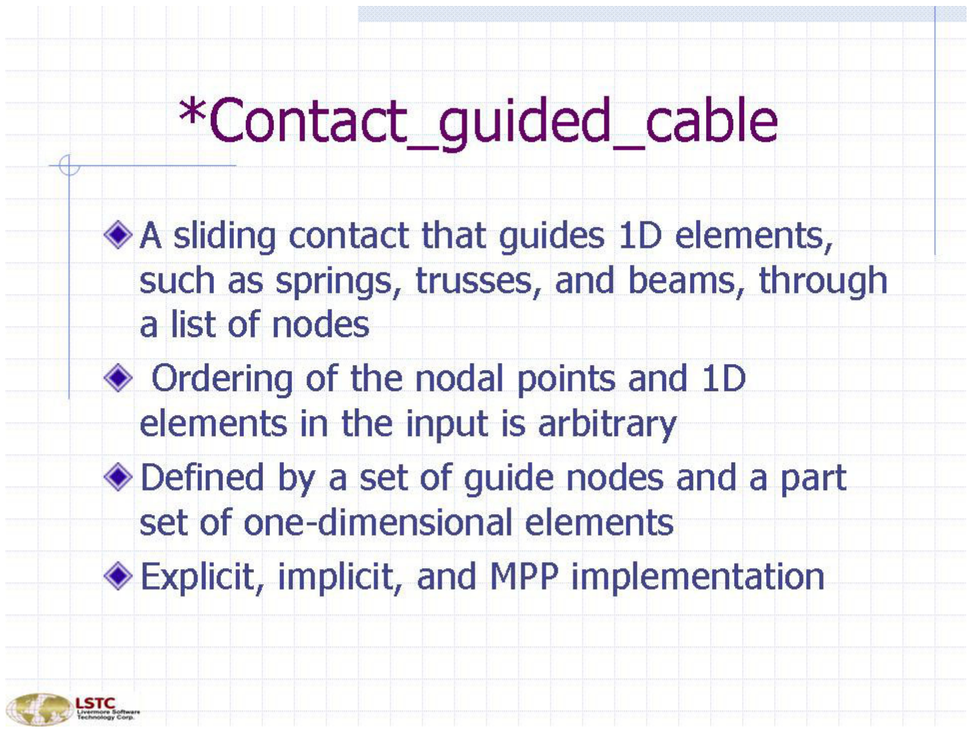
Mass property output

- *Database_glstat_mass_properties: This is an option for the glstat file to include global mass and inertial properties in the output for each output state.
  - Mass center, mass, inertia tensor, principle inertias
  - Computed from nodal point and rigid body mass and inertia.
  - Excludes failed nodes and elements
- *Database_ssstat_mass_properties: This is an option for the ssstat file to include mass and inertial properties for the subsystems.



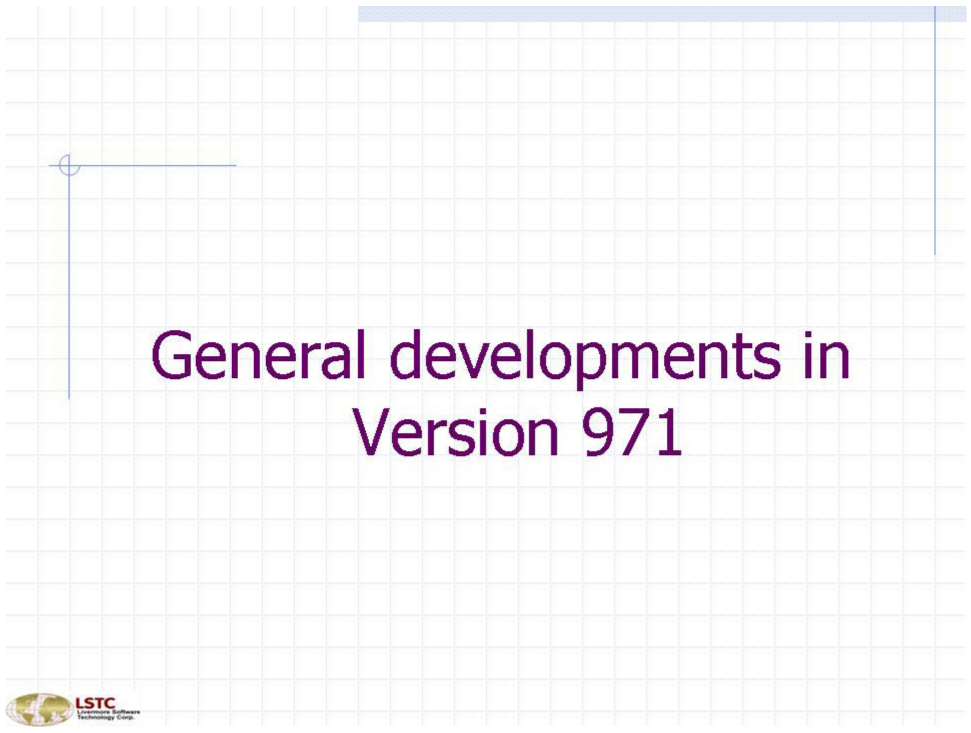
*Termination_deleted_shells

- NFAIL1 and NFAIL4, which are defined on *CONTROL_SHELL, checks for negative jacobians and deletes any shells where one is found
- If the NFAIL1/NFAIL4 option is set, the calculation can be terminated based on the number of elements that have failed within a given part ID. The number of failed shells is specified by this *Termination option.
- SMP and MPP implementation

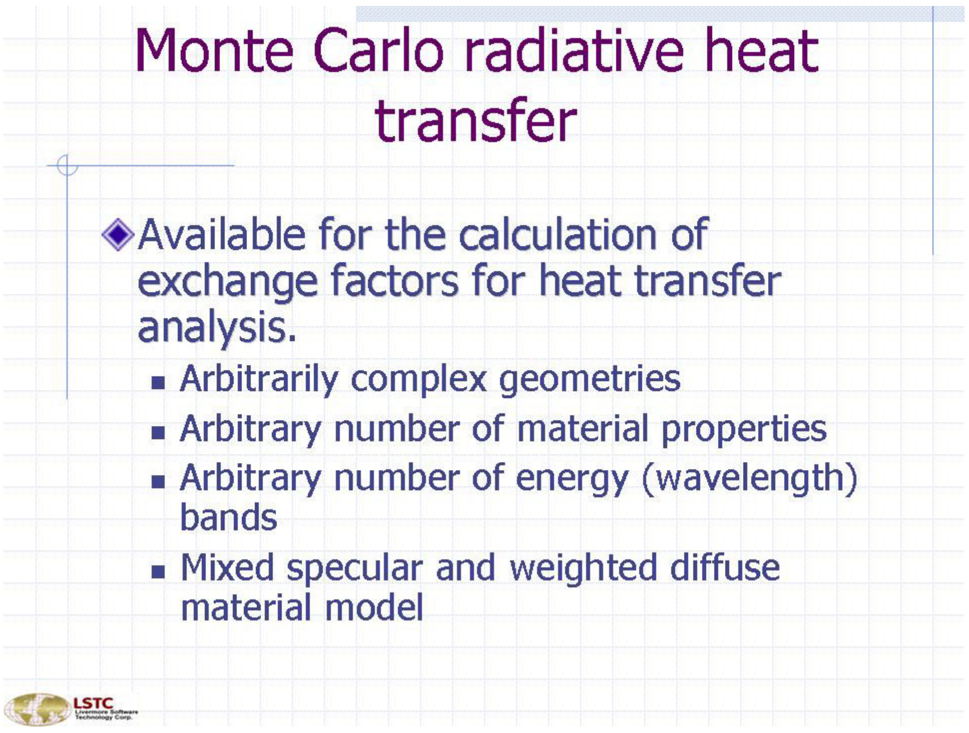


*Contact_guided_cable

- A sliding contact that guides 1D elements, such as springs, trusses, and beams, through a list of nodes
- Ordering of the nodal points and 1D elements in the input is arbitrary
- Defined by a set of guide nodes and a part set of one-dimensional elements
- Explicit, implicit, and MPP implementation



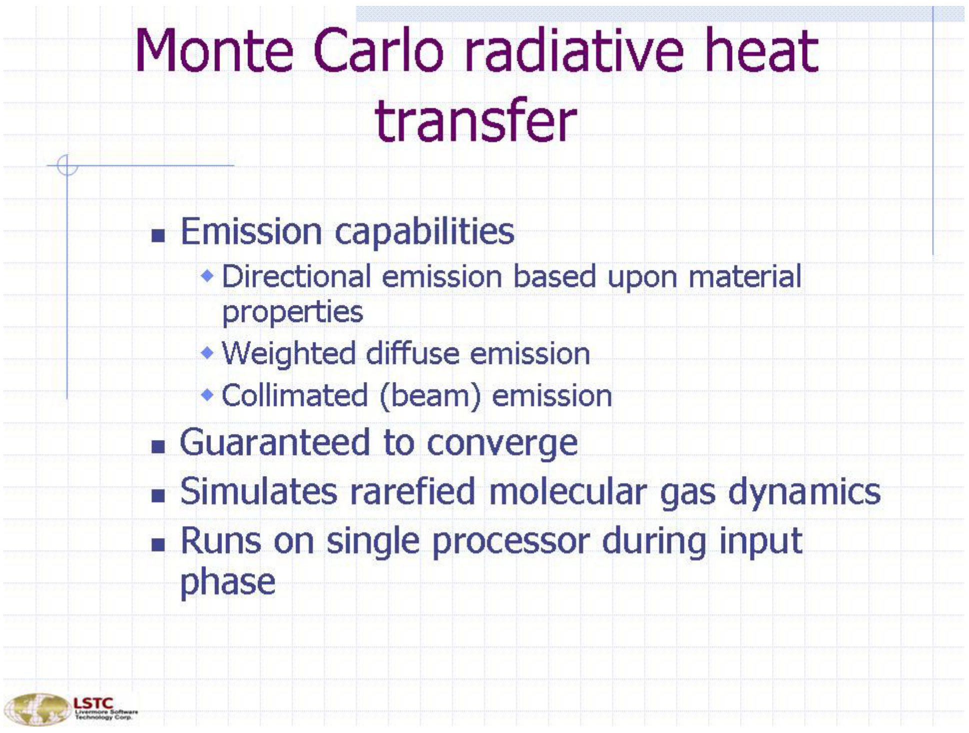
General developments in Version 971



Monte Carlo radiative heat transfer

- Available for the calculation of exchange factors for heat transfer analysis.
  - Arbitrarily complex geometries
  - Arbitrary number of material properties
  - Arbitrary number of energy (wavelength) bands
  - Mixed specular and weighted diffuse material model

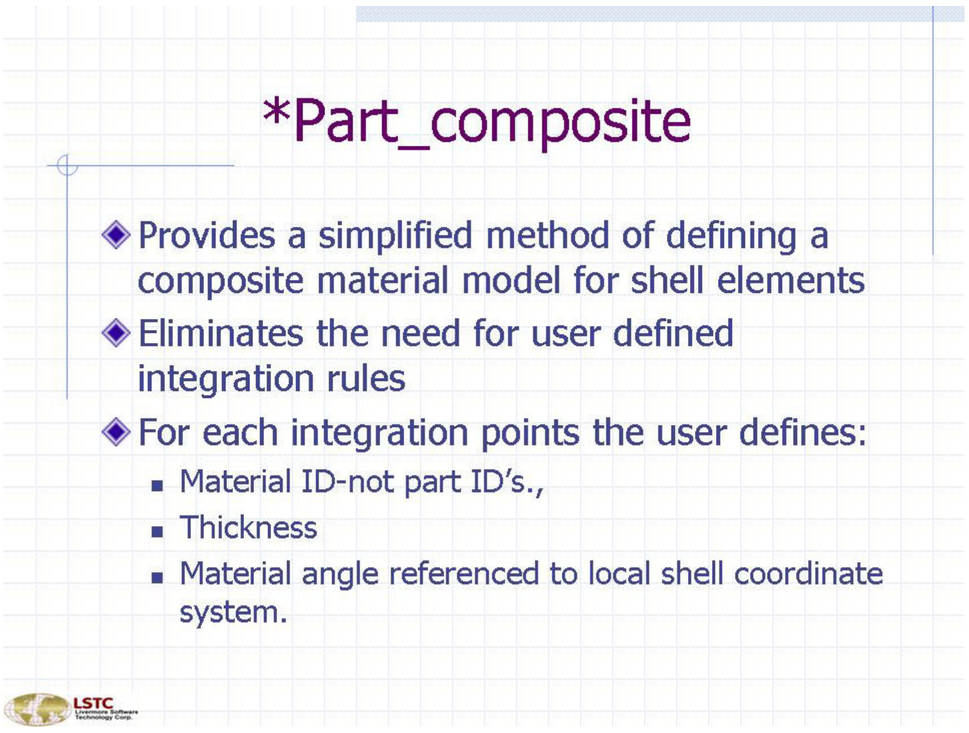
LSTC Livermore Software Technology Corp.



Monte Carlo radiative heat transfer

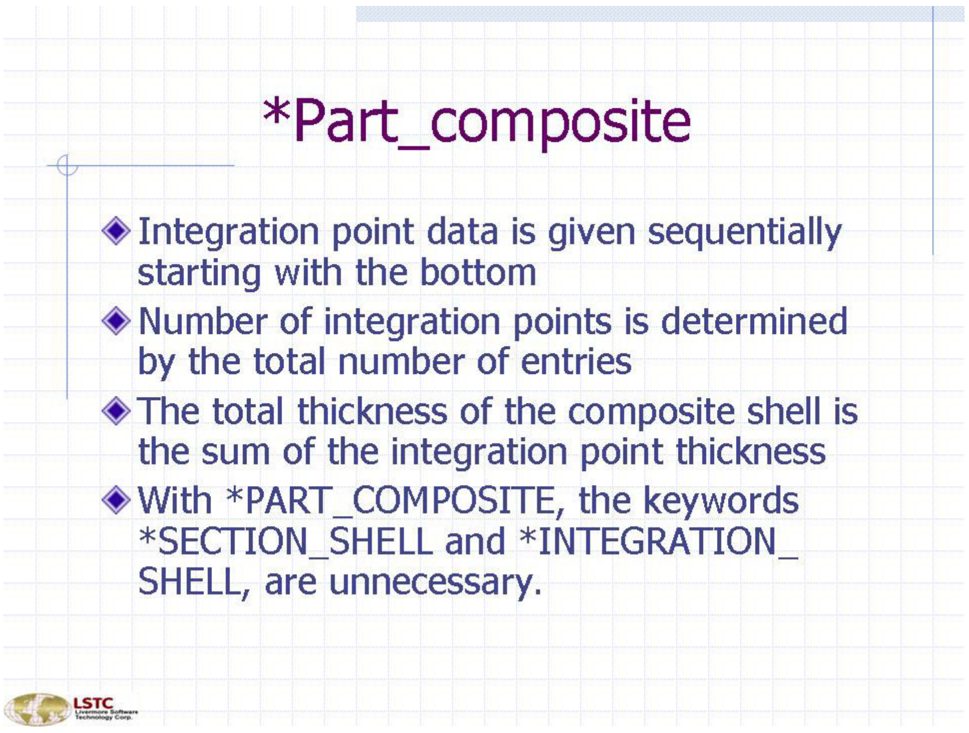
- Emission capabilities
  - Directional emission based upon material properties
  - Weighted diffuse emission
  - Collimated (beam) emission
- Guaranteed to converge
- Simulates rarefied molecular gas dynamics
- Runs on single processor during input phase

LSTC Livermore Software Technology Corp.



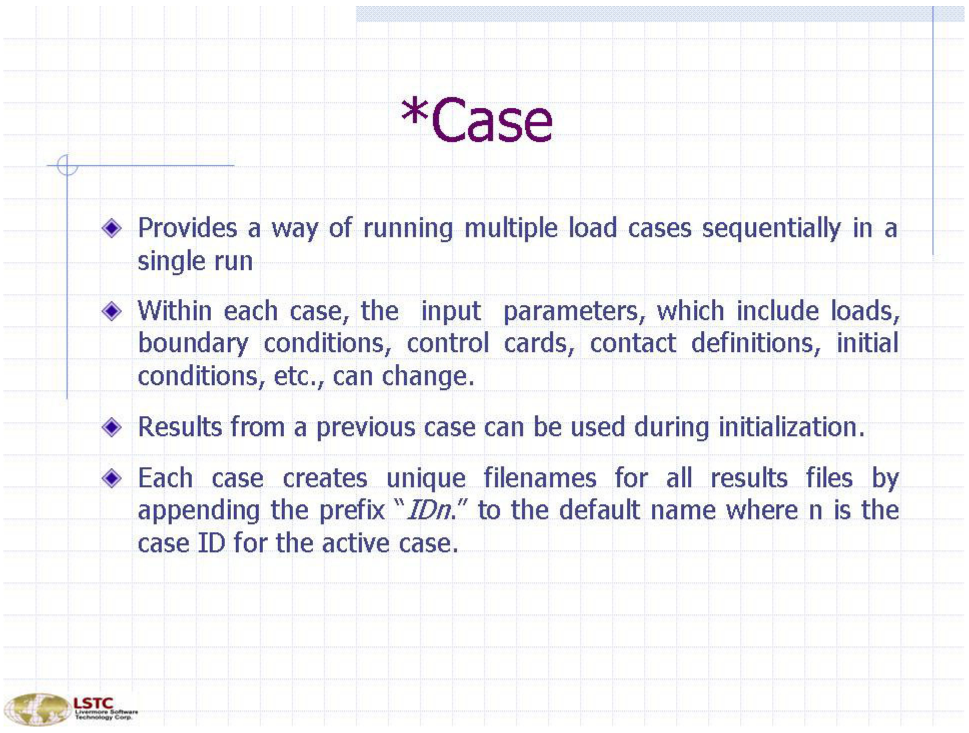
*Part_composite

- Provides a simplified method of defining a composite material model for shell elements
- Eliminates the need for user defined integration rules
- For each integration points the user defines:
  - Material ID-not part ID's.,
  - Thickness
  - Material angle referenced to local shell coordinate system.



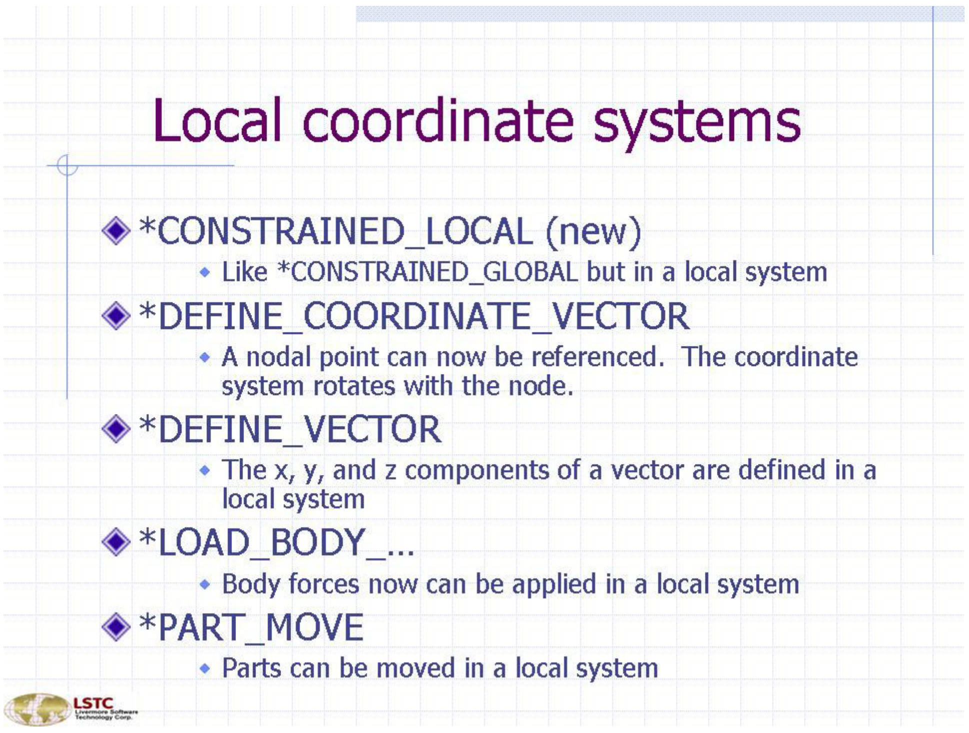
*Part_composite

- Integration point data is given sequentially starting with the bottom
- Number of integration points is determined by the total number of entries
- The total thickness of the composite shell is the sum of the integration point thickness
- With *PART_COMPOSITE, the keywords *SECTION_SHELL and *INTEGRATION_ SHELL, are unnecessary.



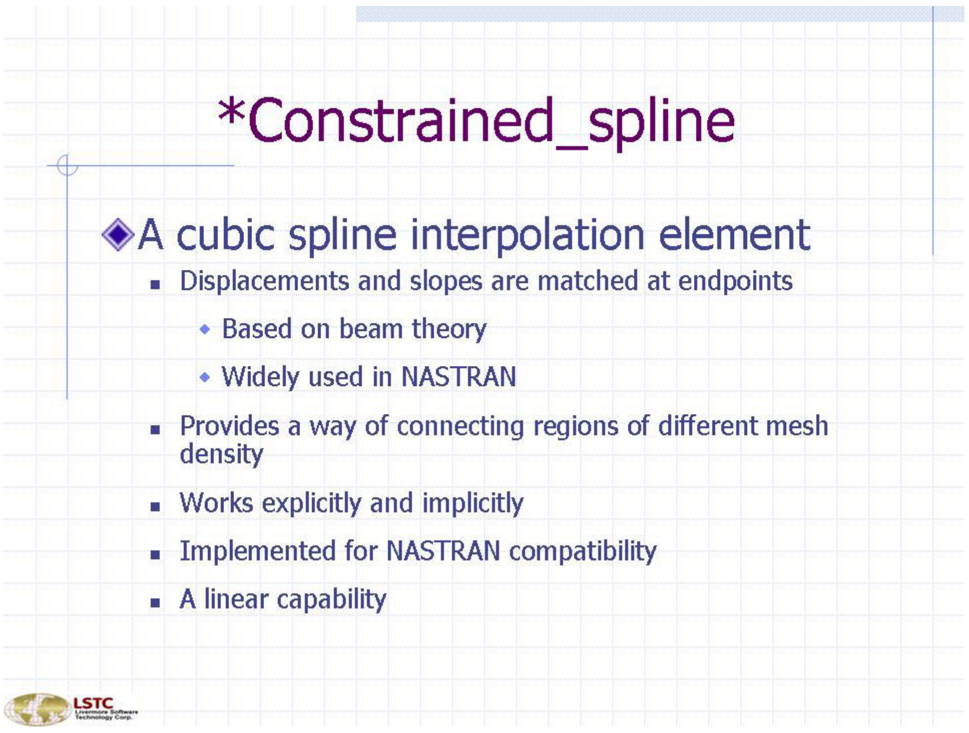
*Case

- Provides a way of running multiple load cases sequentially in a single run
- Within each case, the input parameters, which include loads, boundary conditions, control cards, contact definitions, initial conditions, etc., can change.
- Results from a previous case can be used during initialization.
- Each case creates unique filenames for all results files by appending the prefix "*IDn.*" to the default name where n is the case ID for the active case.



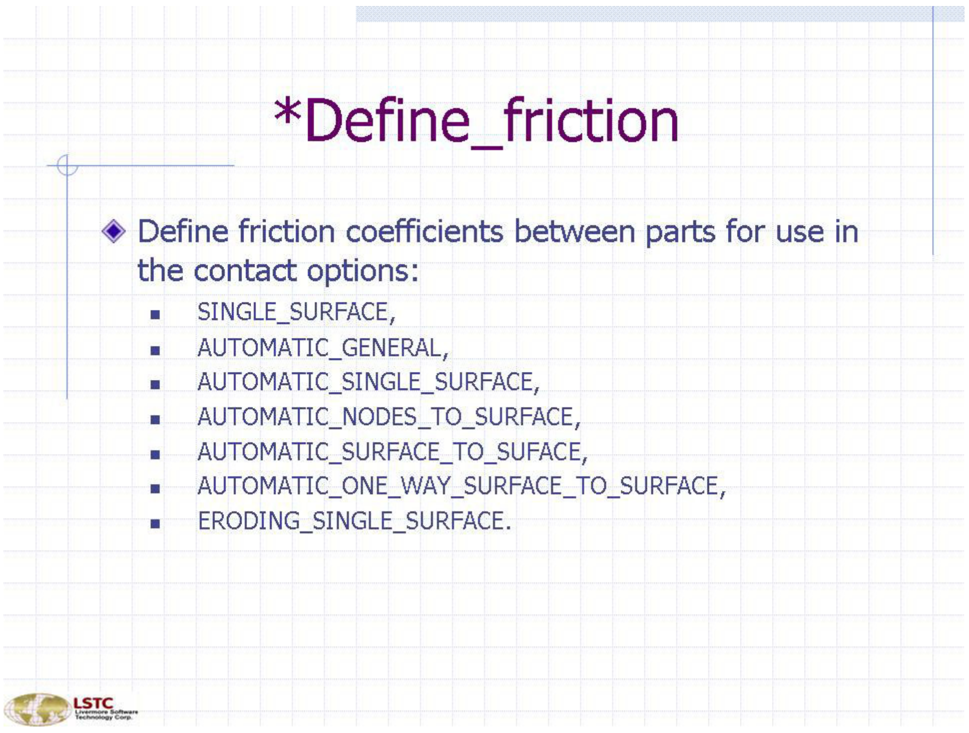
Local coordinate systems

- *CONSTRAINED_LOCAL (new)
  - Like *CONSTRAINED_GLOBAL but in a local system
- *DEFINE_COORDINATE_VECTOR
  - A nodal point can now be referenced. The coordinate system rotates with the node.
- *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



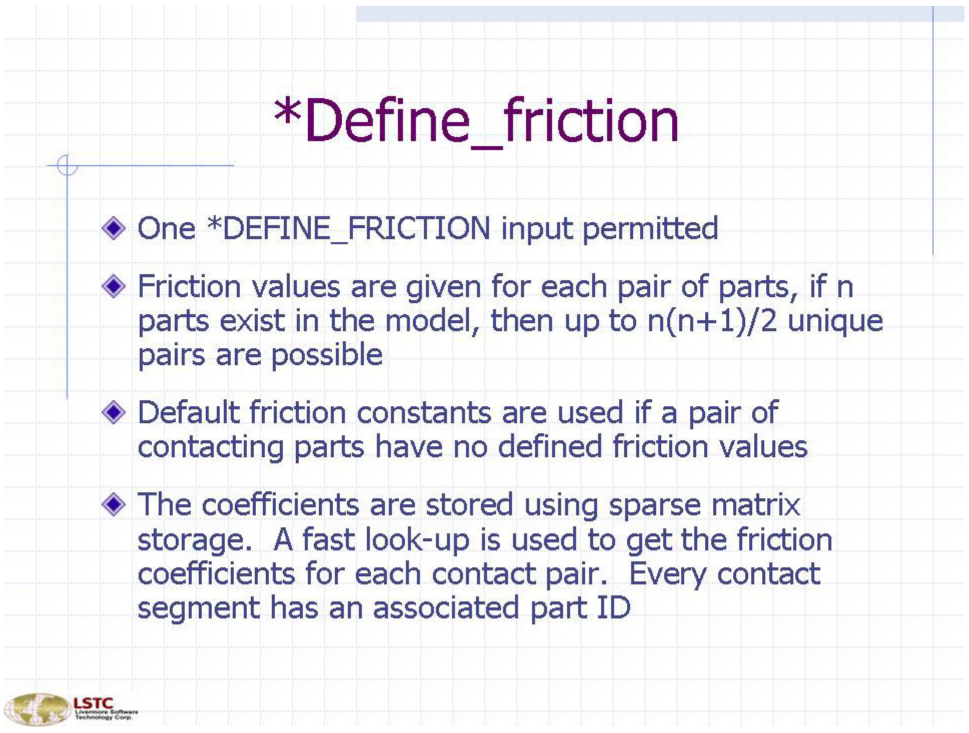
*Constrained_spline

- 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
  - A linear capability



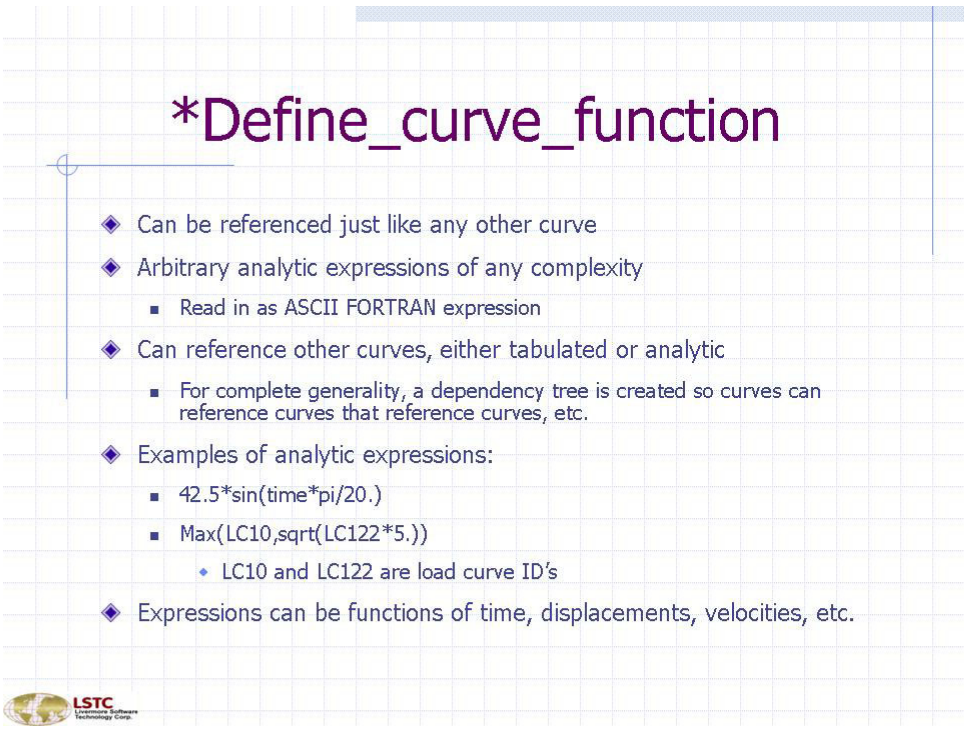
*Define_friction

- Define friction coefficients between parts for use in the contact options:
  - SINGLE_SURFACE,
  - AUTOMATIC_GENERAL,
  - AUTOMATIC_SINGLE_SURFACE,
  - AUTOMATIC_NODES_TO_SURFACE,
  - AUTOMATIC_SURFACE_TO_SUFACE,
  - AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE,
  - ERODING_SINGLE_SURFACE.



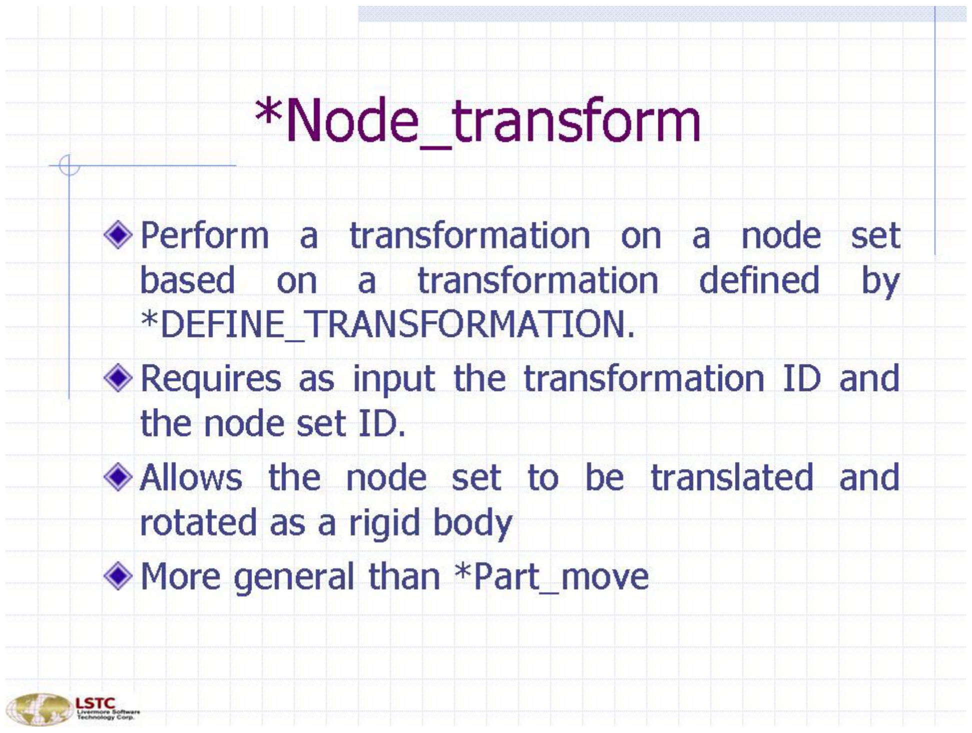
*Define_friction

- One *DEFINE_FRICTION input permitted
- Friction values are given for each pair of parts, if n parts exist in the model, then up to n(n+1)/2 unique pairs are possible
- Default friction constants are used if a pair of contacting parts have no defined friction values
- The coefficients are stored using sparse matrix storage. A fast look-up is used to get the friction coefficients for each contact pair. Every contact segment has an associated part ID



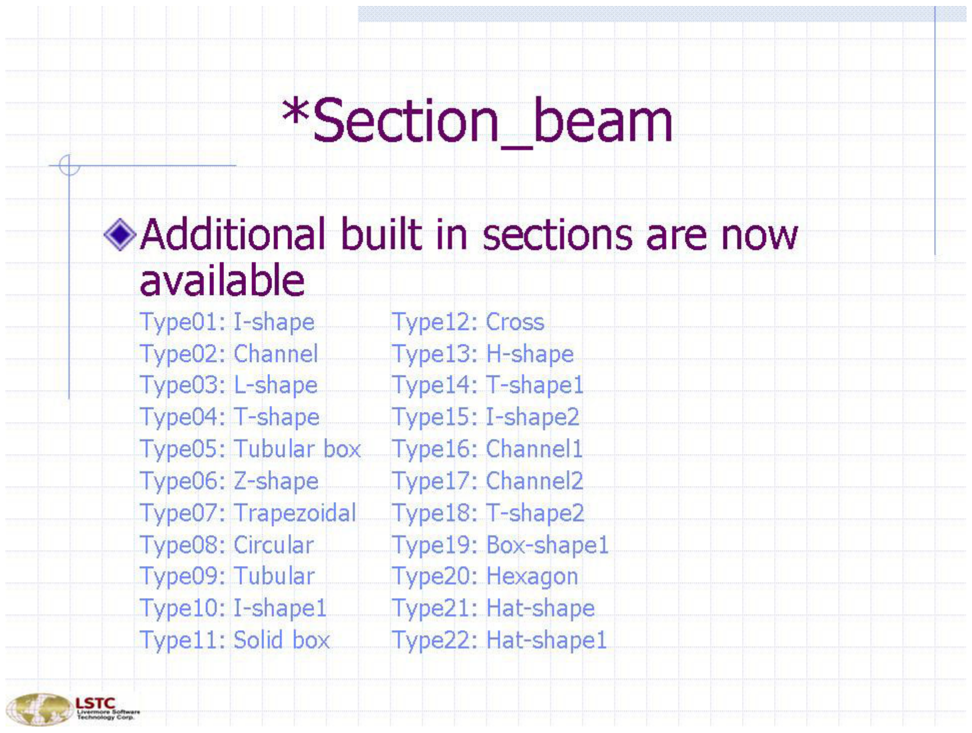
*Define_curve_function

- 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, etc.



*Node_transform

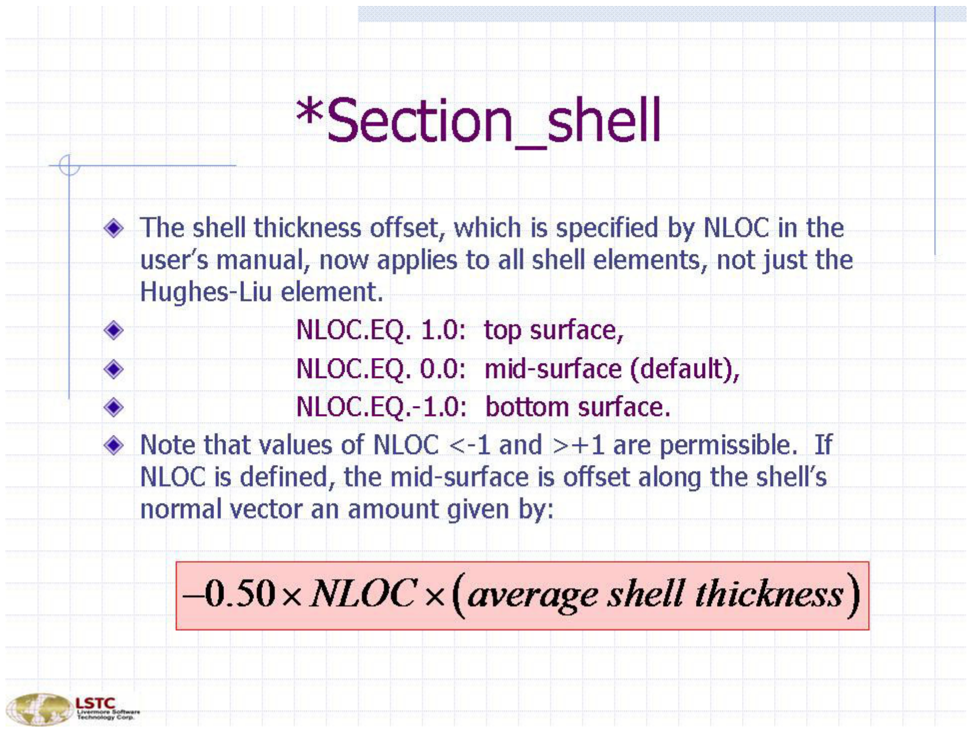
- Perform a transformation on a node set based on a transformation defined by *DEFINE_TRANSFORMATION.
- Requires as input the transformation ID and the node set ID.
- Allows the node set to be translated and rotated as a rigid body
- More general than *Part_move



*Section_beam

- Additional built in sections are now available

Type01: I-shape
Type02: Channel
Type03: L-shape
Type04: T-shape
Type05: Tubular box
Type06: Z-shape
Type07: Trapezoidal
Type08: Circular
Type09: Tubular
Type10: I-shape1
Type11: Solid box
Type12: Cross
Type13: H-shape
Type14: T-shape1
Type15: I-shape2
Type16: Channel1
Type17: Channel2
Type18: T-shape2
Type19: Box-shape1
Type20: Hexagon
Type21: Hat-shape
Type22: Hat-shape1

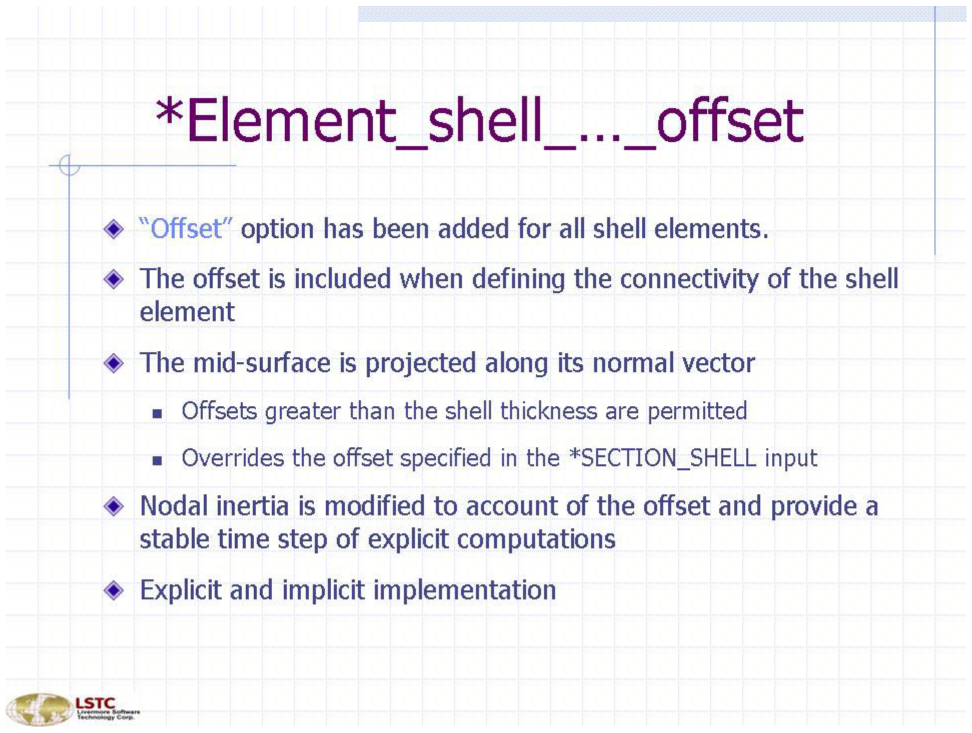


*Section_shell

- The shell thickness offset, which is specified by NLOC in the user's manual, now applies to all shell elements, not just the Hughes-Liu element.
- NLOC.EQ. 1.0: top surface,
- NLOC.EQ. 0.0: mid-surface (default),
- NLOC.EQ.-1.0: bottom surface.
- Note that values of NLOC <-1 and >+1 are permissible. If NLOC is defined, the mid-surface is offset along the shell's normal vector an amount given by:

$$-0.50 \times NLOC \times \left(average\ shell\ thickness\right)$$

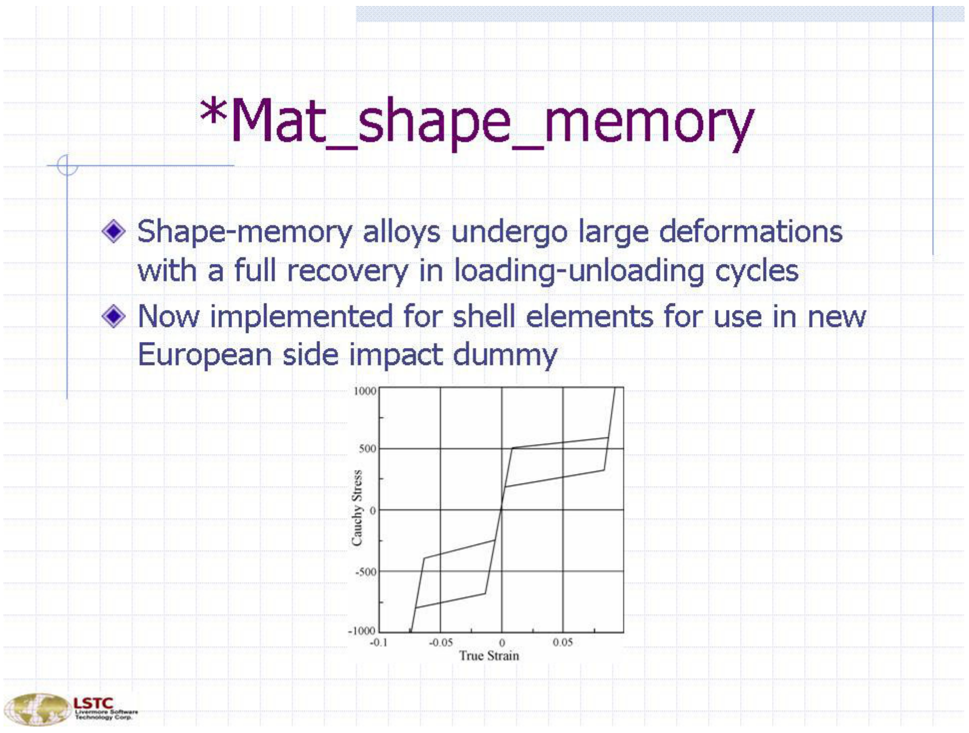
LSTC Livermore Software Technology Corp.



*Element_shell_..._offset

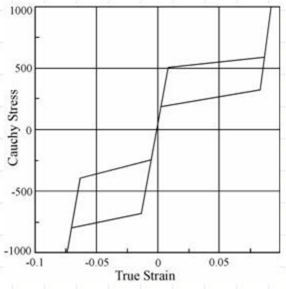
- "Offset" option has been added for all shell elements.
- The offset is included when defining the connectivity of the shell element
- The mid-surface is projected along its normal vector
  - Offsets greater than the shell thickness are permitted
  - Overrides the offset specified in the *SECTION_SHELL input
- Nodal inertia is modified to account of the offset and provide a stable time step of explicit computations
- Explicit and implicit implementation

LSTC Livermore Software Technology Corp.

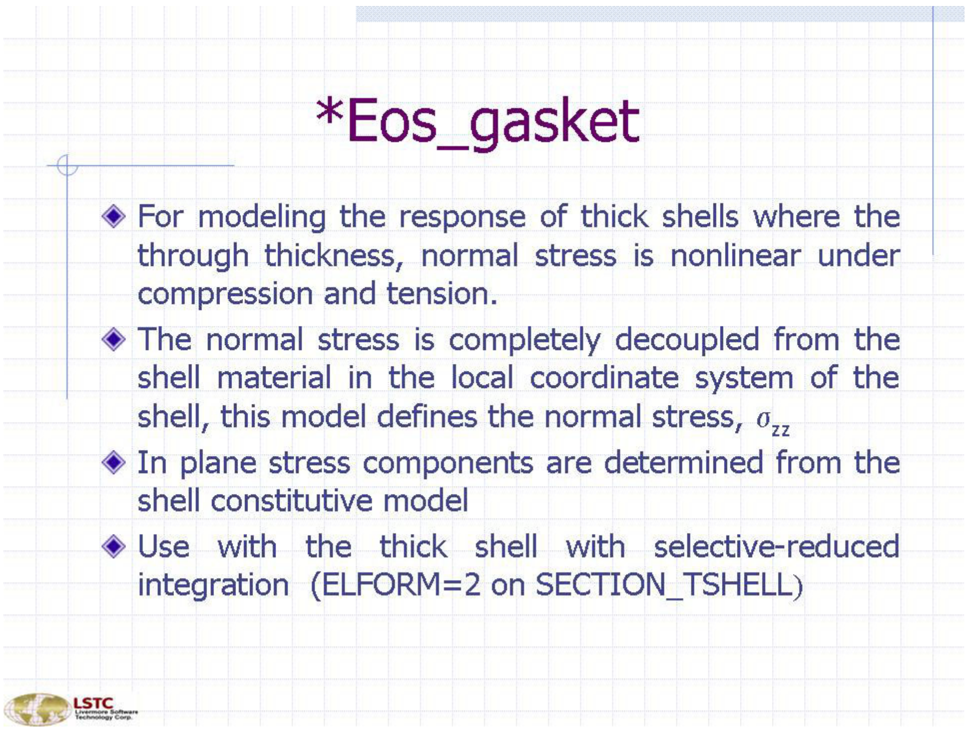


*Mat_shape_memory

- Shape-memory alloys undergo large deformations with a full recovery in loading-unloading cycles
- Now implemented for shell elements for use in new European side impact dummy



LSTC Livermore Software Technology Corp.



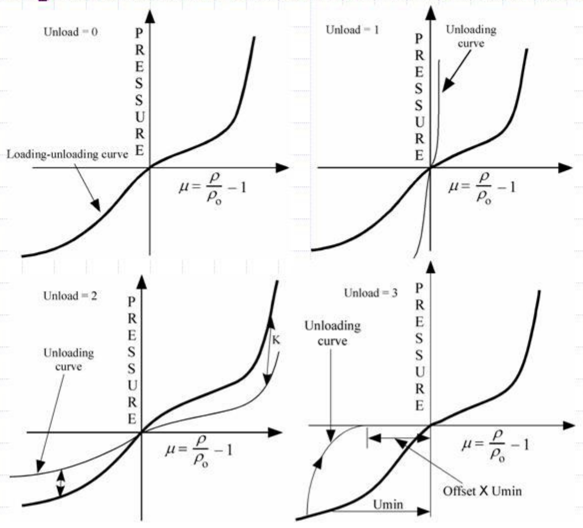
*Eos_gasket

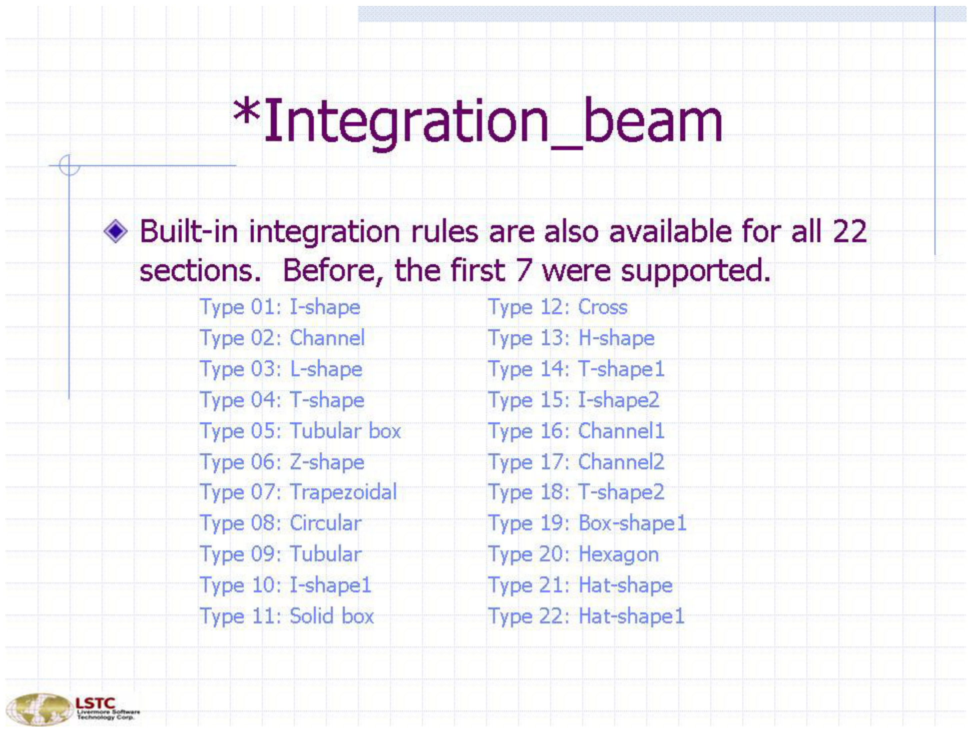
- For modeling the response of thick shells where the through thickness, normal stress is nonlinear under compression and tension.
- The normal stress is completely decoupled from the shell material in the local coordinate system of the shell, this model defines the normal stress, σ_{zz}
- In plane stress components are determined from the shell constitutive model
- Use with the thick shell with selective-reduced integration (ELFORM=2 on SECTION_TSHELL)

LSTC Livermore Software Technology Corp.

*Eos_gasket

Loading and unload behaviors for normal stress

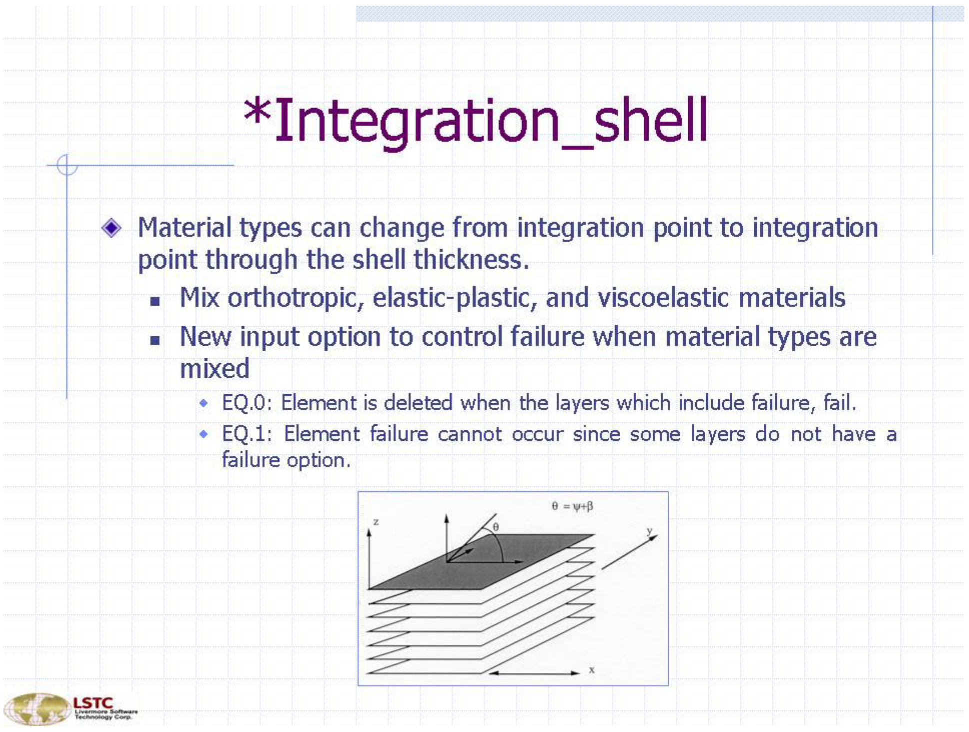




*Integration_beam

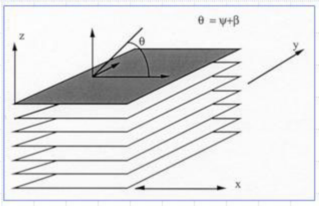
◆ Built-in integration rules are also available for all 22 sections. Before, the first 7 were supported.

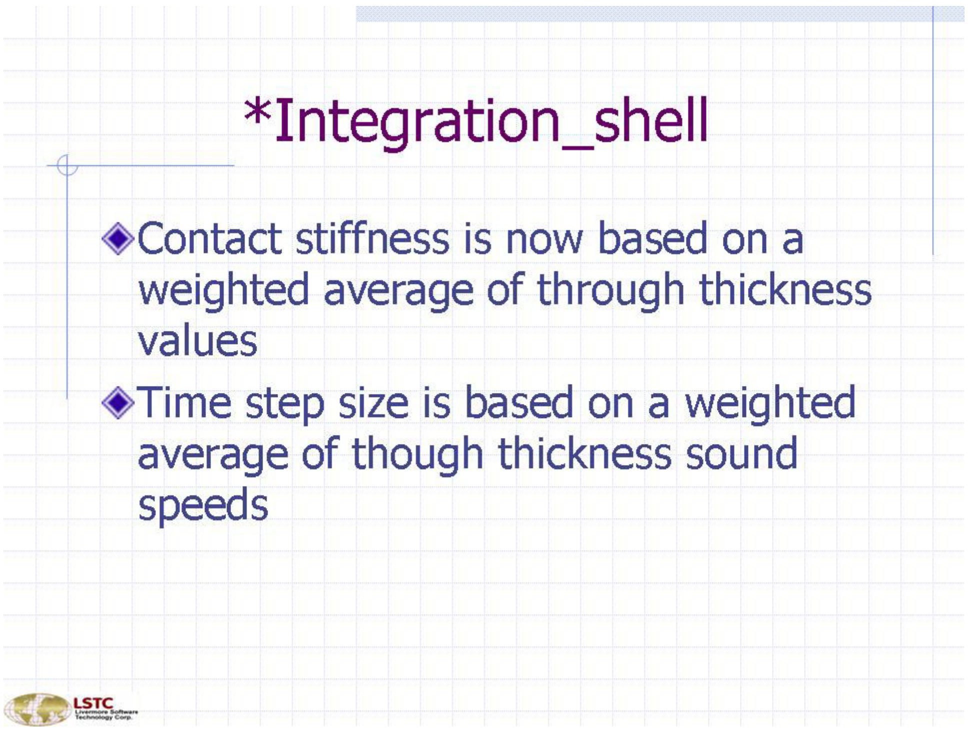
Type 01: I-shape	Type 12: Cross
Type 02: Channel	Type 13: H-shape
Type 03: L-shape	Type 14: T-shape1
Type 04: T-shape	Type 15: I-shape2
Type 05: Tubular box	Type 16: Channel1
Type 06: Z-shape	Type 17: Channel2
Type 07: Trapezoidal	Type 18: T-shape2
Type 08: Circular	Type 19: Box-shape1
Type 09: Tubular	Type 20: Hexagon
Type 10: I-shape1	Type 21: Hat-shape
Type 11: Solid box	Type 22: Hat-shape1



*Integration_shell

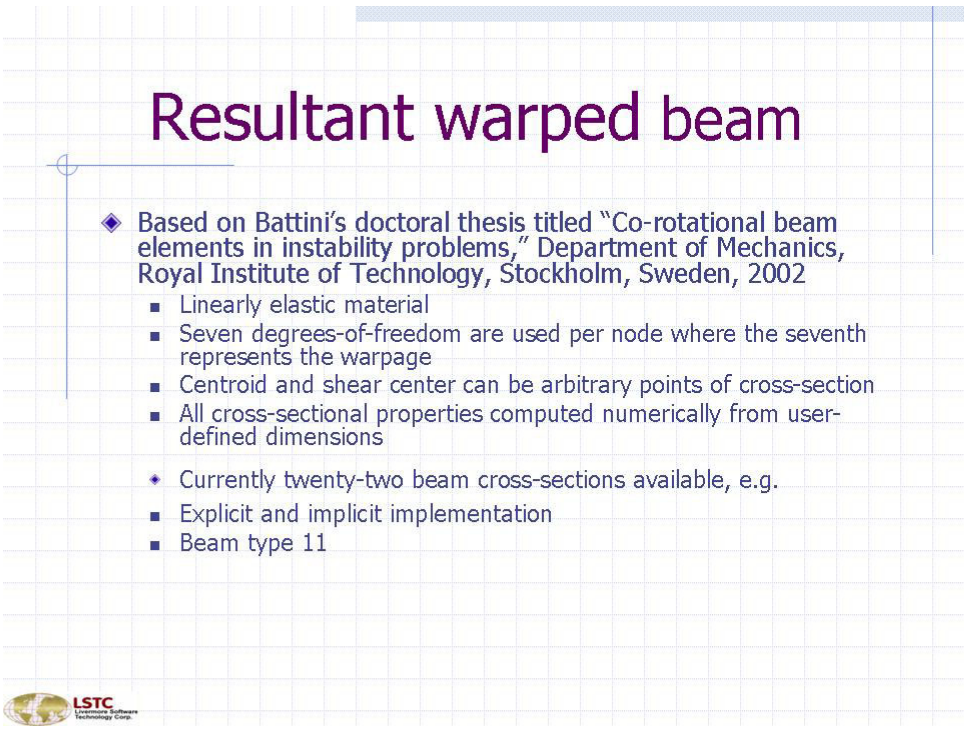
- Material types can change from integration point to integration point through the shell thickness.
  - Mix orthotropic, elastic-plastic, and viscoelastic materials
  - New input option to control failure when material types are mixed
    - EQ.0: Element is deleted when the layers which include failure, fail.
    - EQ.1: Element failure cannot occur since some layers do not have a failure option.





*Integration_shell

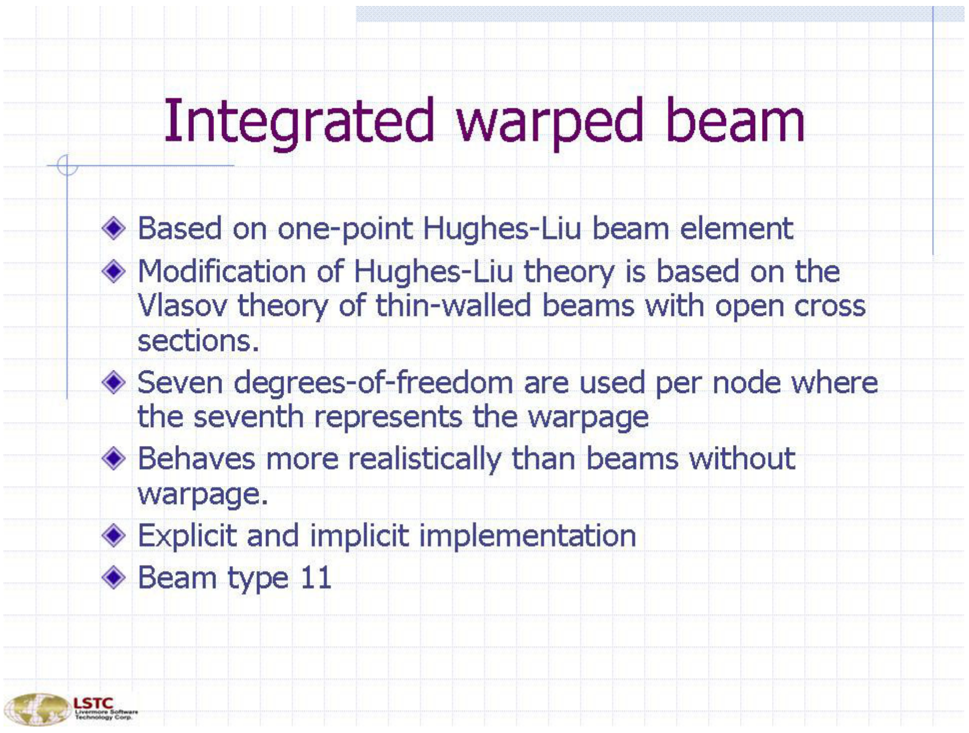
- Contact stiffness is now based on a weighted average of through thickness values
- Time step size is based on a weighted average of though thickness sound speeds



Resultant warped beam

- Based on Battini's doctoral thesis titled "Co-rotational beam elements in instability problems," Department of Mechanics, Royal Institute of Technology, Stockholm, Sweden, 2002
  - Linearly elastic material
  - Seven degrees-of-freedom are used per node where the seventh represents the warpage
  - Centroid and shear center can be arbitrary points of cross-section
  - All cross-sectional properties computed numerically from user-defined dimensions
  - Currently twenty-two beam cross-sections available, e.g.
  - Explicit and implicit implementation
  - Beam type 11

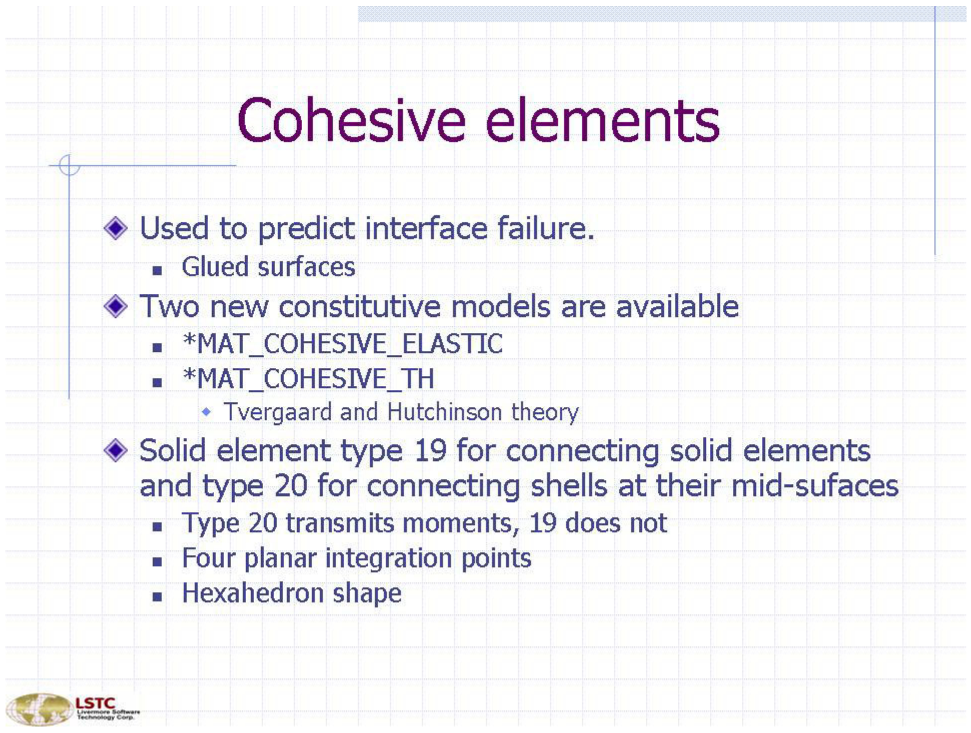
LSTC Livermore Software Technology Corp.



Integrated warped beam

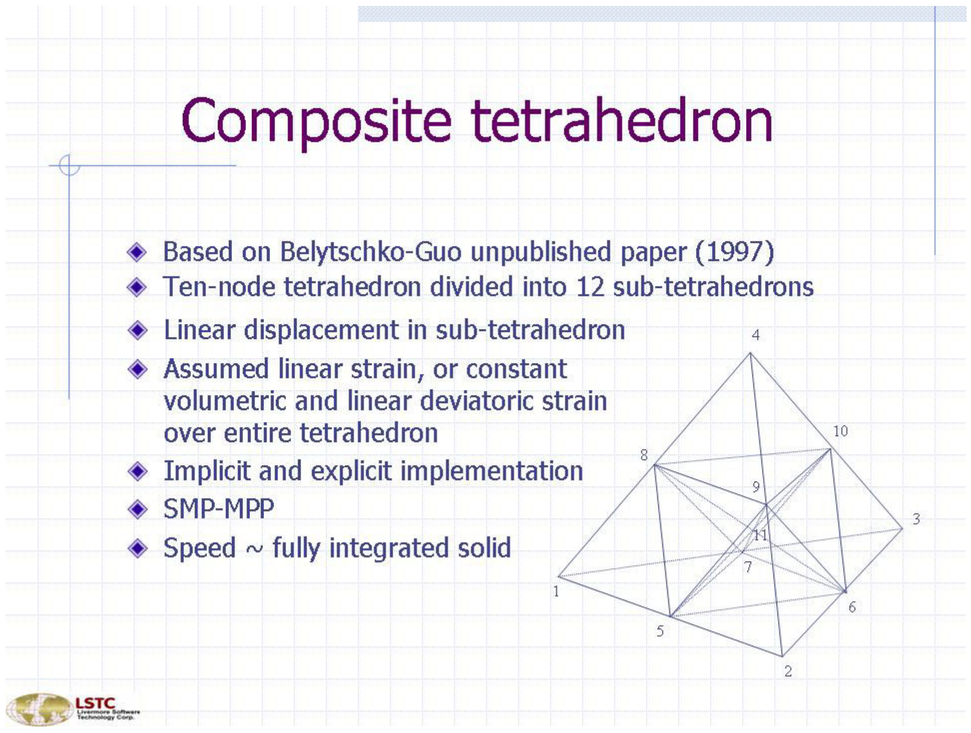
- Based on one-point Hughes-Liu beam element
- Modification of Hughes-Liu theory is based on the Vlasov theory of thin-walled beams with open cross sections.
- Seven degrees-of-freedom are used per node where the seventh represents the warpage
- Behaves more realistically than beams without warpage.
- Explicit and implicit implementation
- Beam type 11

LSTC Livermore Software Technology Corp.



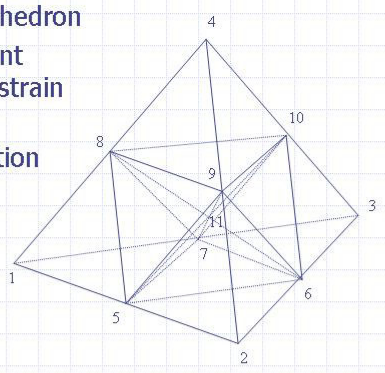
Cohesive elements

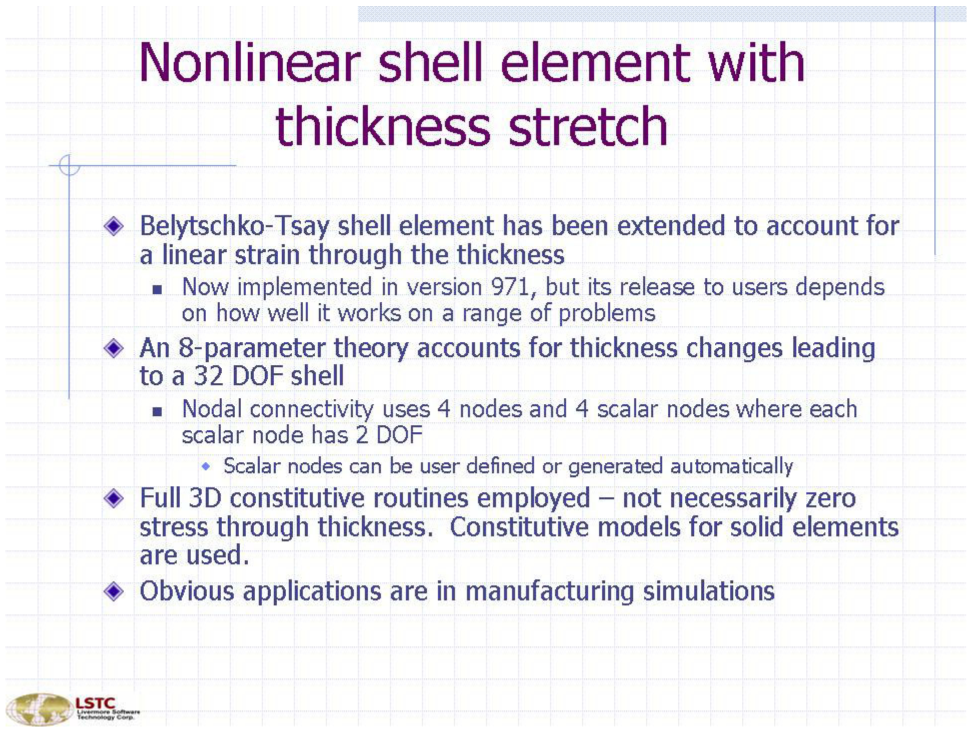
- Used to predict interface failure.
  - Glued surfaces
- Two new constitutive models are available
  - *MAT_COHESIVE_ELASTIC
  - *MAT_COHESIVE_TH
    - Tvergaard and Hutchinson theory
- Solid element type 19 for connecting solid elements and type 20 for connecting shells at their mid-sufaces
  - Type 20 transmits moments, 19 does not
  - Four planar integration points
  - Hexahedron shape



Composite tetrahedron

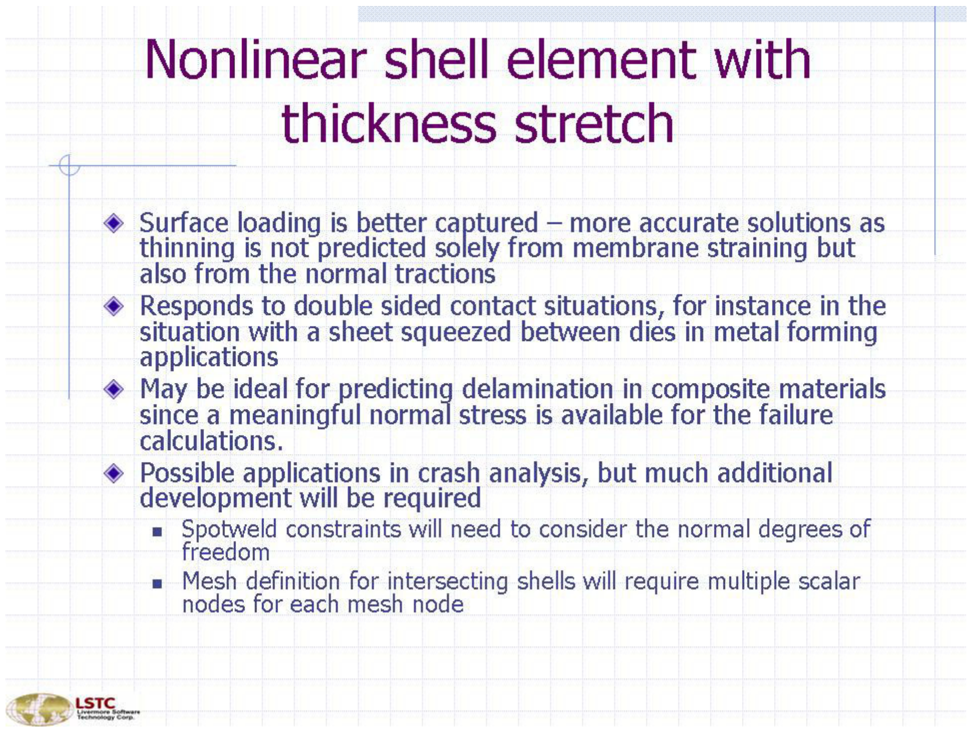
- Based on Belytschko-Guo unpublished paper (1997)
- Ten-node tetrahedron divided into 12 sub-tetrahedrons
- Linear displacement in sub-tetrahedron
- Assumed linear strain, or constant volumetric and linear deviatoric strain over entire tetrahedron
- Implicit and explicit implementation
- SMP-MPP
- Speed ~ fully integrated solid





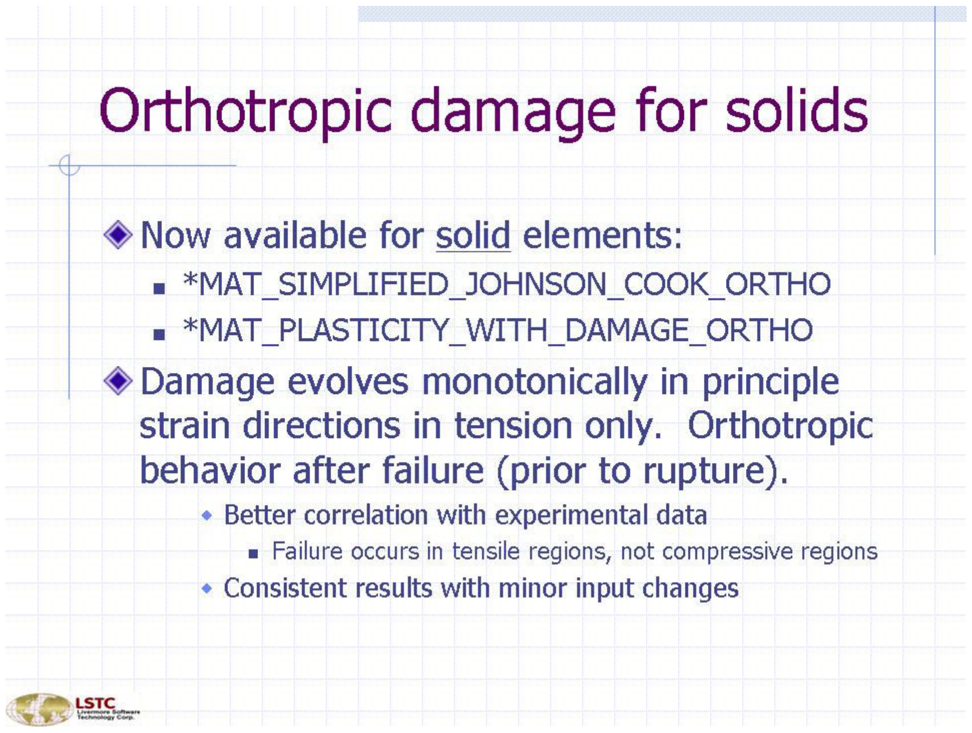
Nonlinear shell element with thickness stretch

- Belytschko-Tsay shell element has been extended to account for a linear strain through the thickness
  - Now implemented in version 971, but its release to users depends on how well it works on a range of problems
- An 8-parameter theory accounts for thickness changes leading to a 32 DOF shell
  - Nodal connectivity uses 4 nodes and 4 scalar nodes where each scalar node has 2 DOF
    - Scalar nodes can be user defined or generated automatically
- Full 3D constitutive routines employed – not necessarily zero stress through thickness. Constitutive models for solid elements are used.
- Obvious applications are in manufacturing simulations



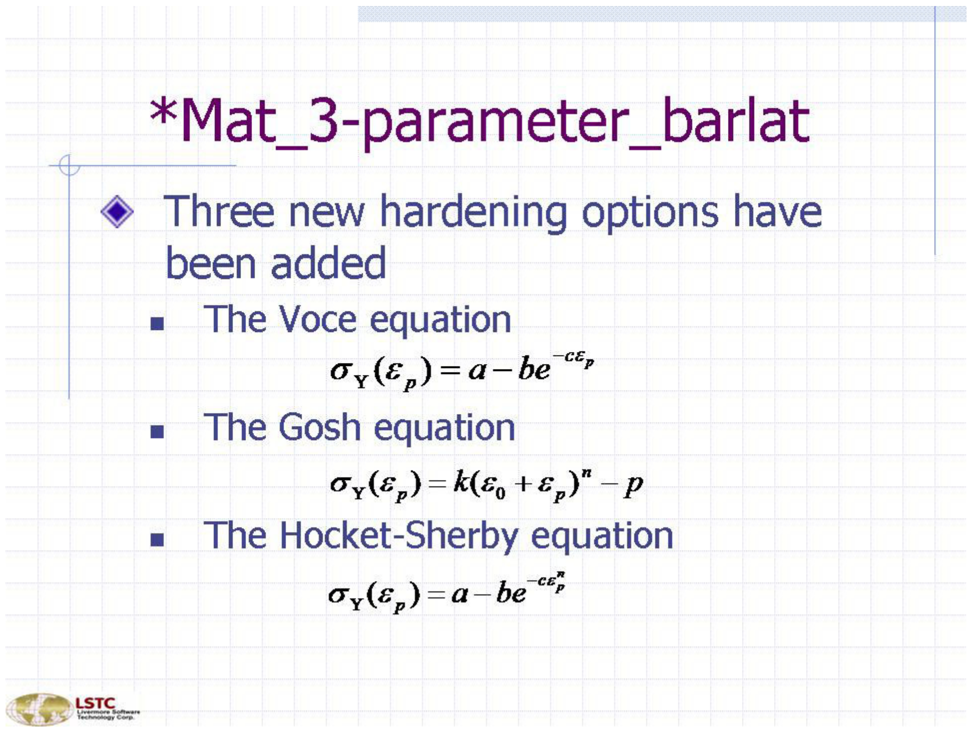
Nonlinear shell element with thickness stretch

- Surface loading is better captured – more accurate solutions as thinning is not predicted solely from membrane straining but also from the normal tractions
- Responds to double sided contact situations, for instance in the situation with a sheet squeezed between dies in metal forming applications
- May be ideal for predicting delamination in composite materials since a meaningful normal stress is available for the failure calculations.
- Possible applications in crash analysis, but much additional development will be required
  - Spotweld constraints will need to consider the normal degrees of freedom
  - Mesh definition for intersecting shells will require multiple scalar nodes for each mesh node



Orthotropic damage for solids

- Now available for solid elements:
  - *MAT_SIMPLIFIED_JOHNSON_COOK_ORTHO
  - *MAT_PLASTICITY_WITH_DAMAGE_ORTHO
- Damage evolves monotonically in principle strain directions in tension only. Orthotropic behavior after failure (prior to rupture).
  - Better correlation with experimental data
    - Failure occurs in tensile regions, not compressive regions
  - Consistent results with minor input changes



*Mat_3-parameter_barlat

- Three new hardening options have been added
  - The Voce equation

$$\sigma_Y(\varepsilon_p) = a - be^{-c\varepsilon_p}$$

  - The Gosh equation

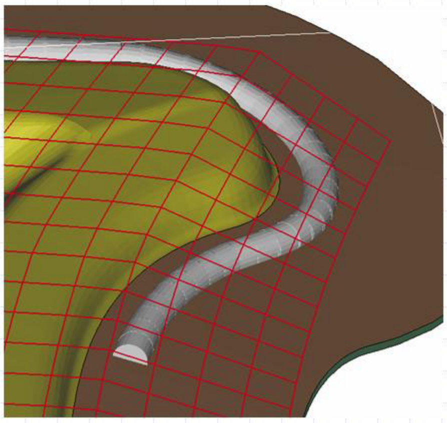
$$\sigma_Y(\varepsilon_p) = k(\varepsilon_0 + \varepsilon_p)^n - p$$

  - The Hocket-Sherby equation

$$\sigma_Y(\varepsilon_p) = a - be^{-c\varepsilon_p^n}$$

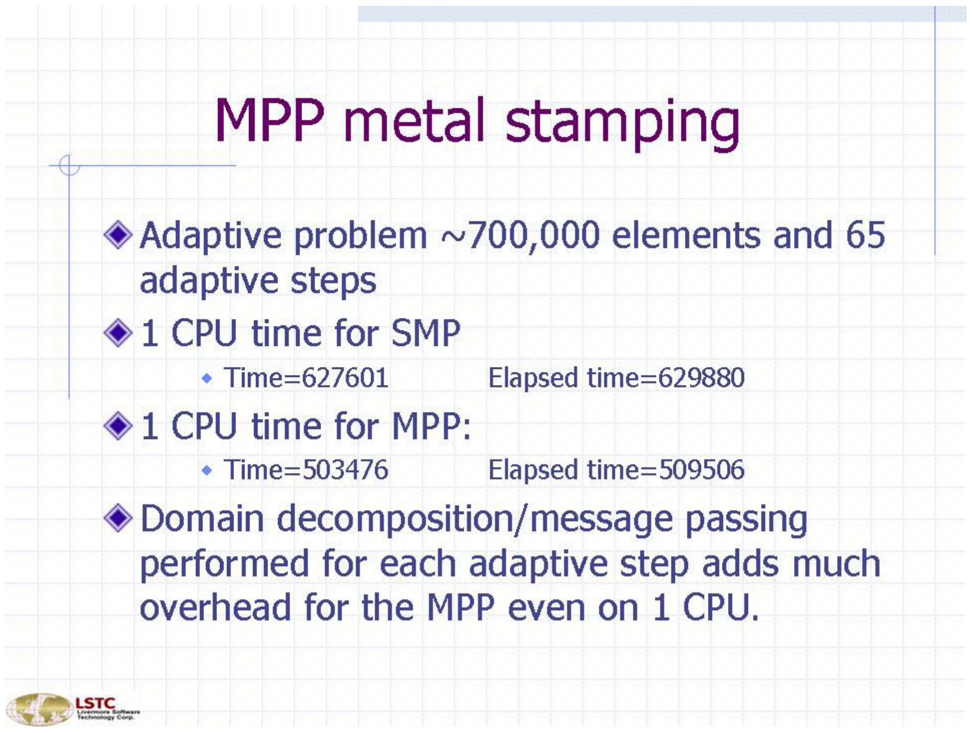
*Define_box_drawbead

Radius of tube centered about the drawbead is now available to limit then number of elements in the drawbead contact



MPP metal stamping

- MPP metal stamping results are now comparable to SMP results on test cases studied by LSTC
- Reasonable speed-up on large problems
  - Overhead due to adaptive steps, which requires redistribution of load based on an updated domain decomposition, hurts scaling
- Scalable MPP implicit seamless springback is now working with answers independent of the number of processors.

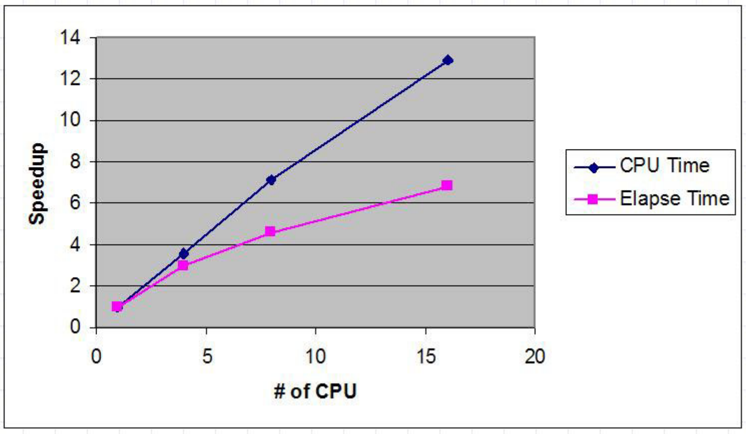


MPP metal stamping

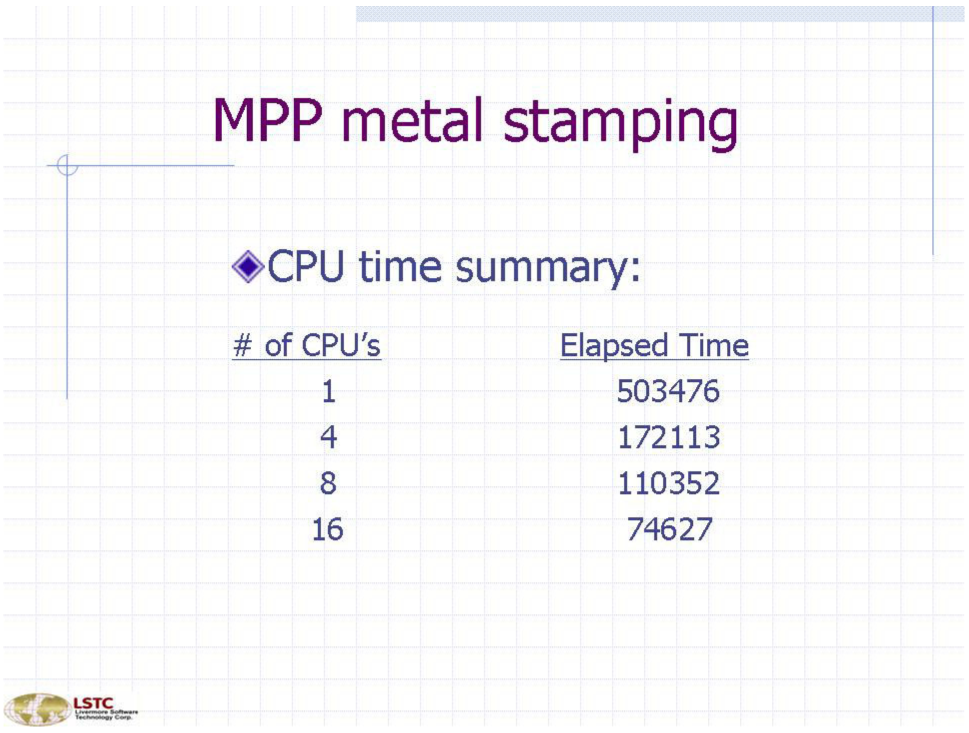
- Adaptive problem ~700,000 elements and 65 adaptive steps
- 1 CPU time for SMP
  - Time=627601 Elapsed time=629880
- 1 CPU time for MPP:
  - Time=503476 Elapsed time=509506
- Domain decomposition/message passing performed for each adaptive step adds much overhead for the MPP even on 1 CPU.

LSTC Livermore Software Technology Corp.

MPP metal stamping



LSTC Livermore Software Technology Corp.



MPP metal stamping

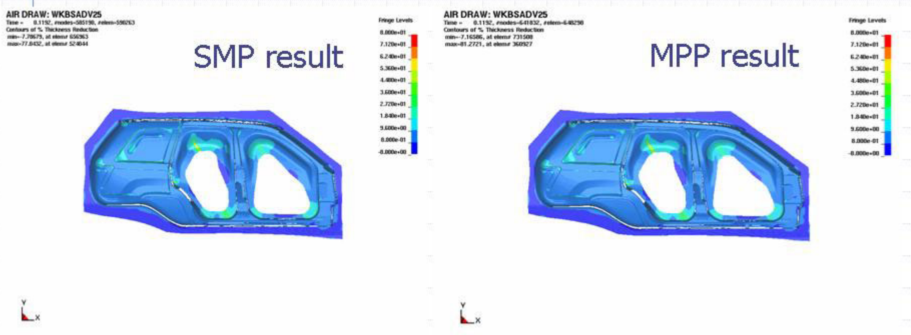
◆CPU time summary:

# of CPU's	Elapsed Time
1	503476
4	172113
8	110352
16	74627

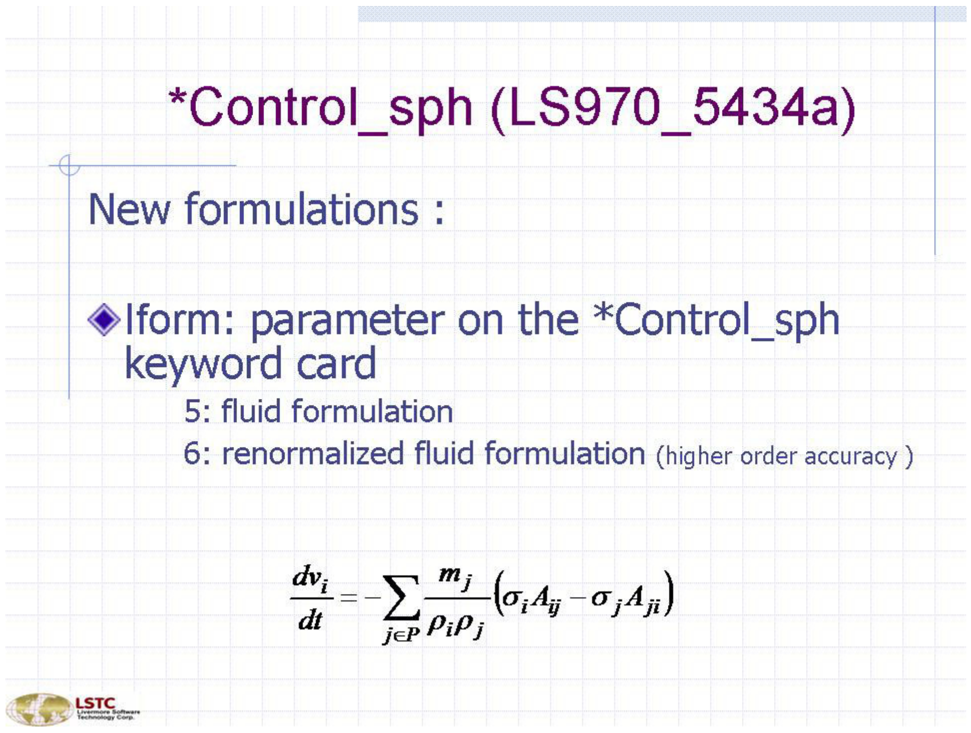
LSTC Livermore Software Technology Corp.

MPP metal stamping

Percent thinning SMP versus MPP



LSTC Livermore Software Technology Corp.



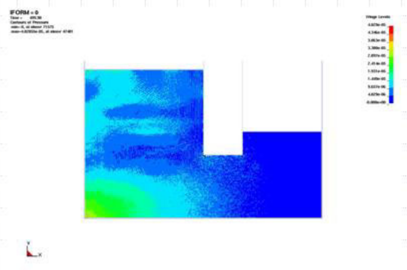
*Control_sph (LS970_5434a)

New formulations :

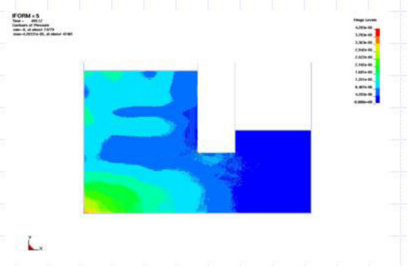
- Iform: parameter on the *Control_sph keyword card
  - 5: fluid formulation
  - 6: renormalized fluid formulation (higher order accuracy)

$$\frac{dv_i}{dt} = -\sum_{j \in P} \frac{m_j}{\rho_i \rho_j} \left(\sigma_i A_{ij} - \sigma_j A_{ji} \right)$$

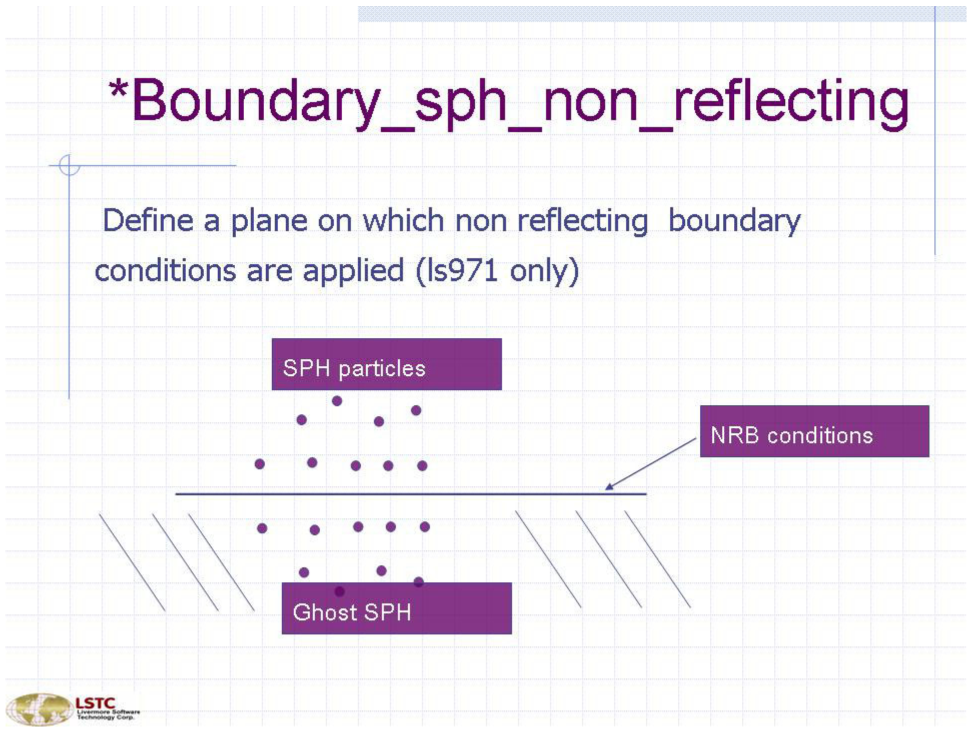
Comparison of 2d plain strain liquid motion



default formulation

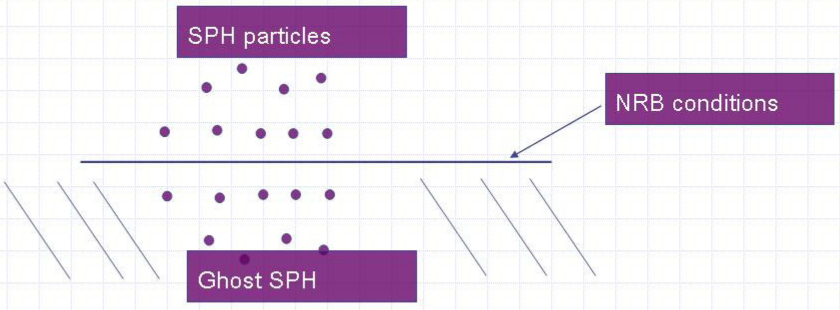


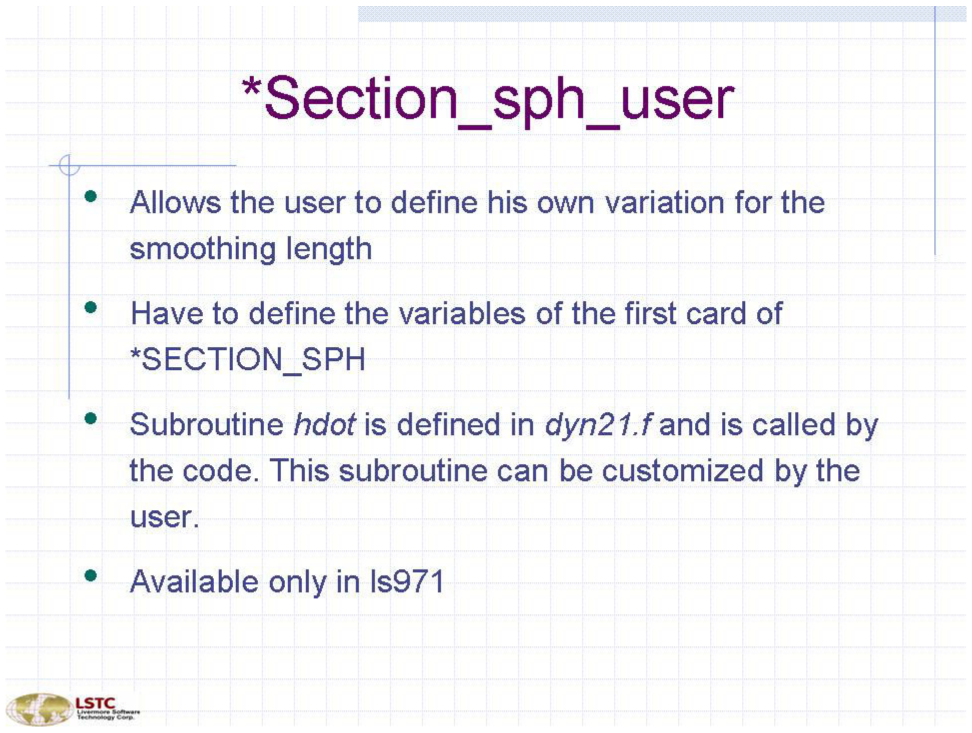
fluid formulation



*Boundary_sph_non_reflecting

Define a plane on which non reflecting boundary conditions are applied (ls971 only)

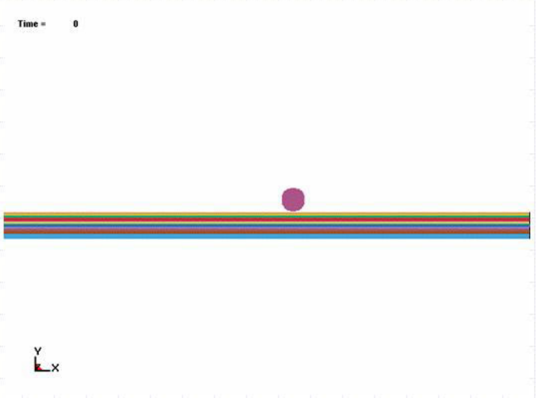




*Section_sph_user

- Allows the user to define his own variation for the smoothing length
- Have to define the variables of the first card of *SECTION_SPH
- Subroutine *hdot* is defined in *dyn21.f* and is called by the code. This subroutine can be customized by the user.
- Available only in ls971

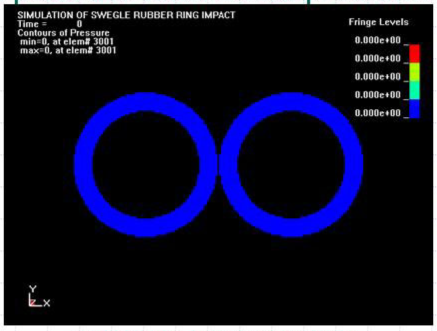
*Section_sph_user



2d plain strain impact of a sphere on a layered target

Tensile instability test

Most SPH formulations suffer from a lack of accuracy and do not pass the tensile test problem:

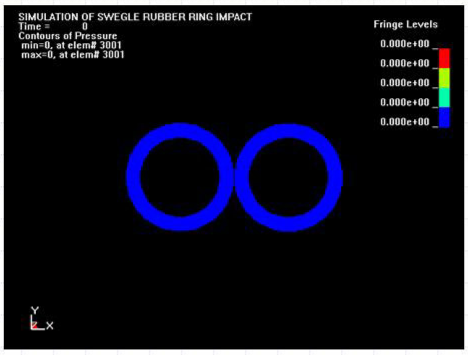


rubber ring impact test from

Swegle and al, S.P.H. Stability Analysis, JCP, 1995

Tensile instability test

Renormalized formulation with iform=6 can pass that test (ls971)



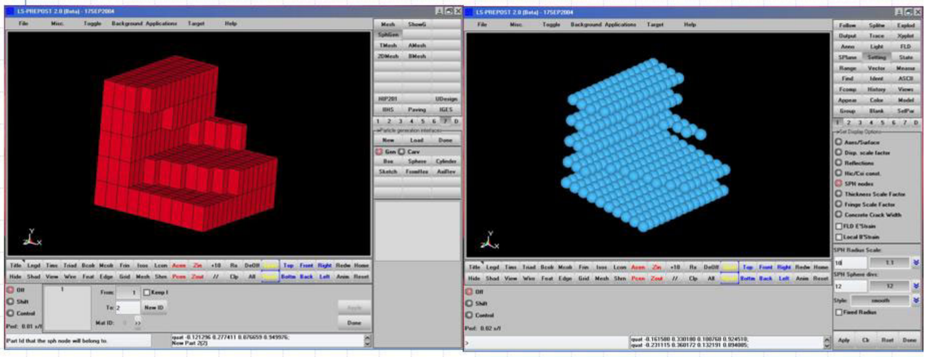
```
SIMULATION OF SWEGLE RUBBER RING IMPACT
Time =        0
Contours of Pressure
 min=0, at elem# 3001
 max=0, at elem# 3001
```

```
Fringe Levels
0.000e+00
0.000e+00
0.000e+00
0.000e+00
0.000e+00
```

LSTC Livermore Software Technology Corp.

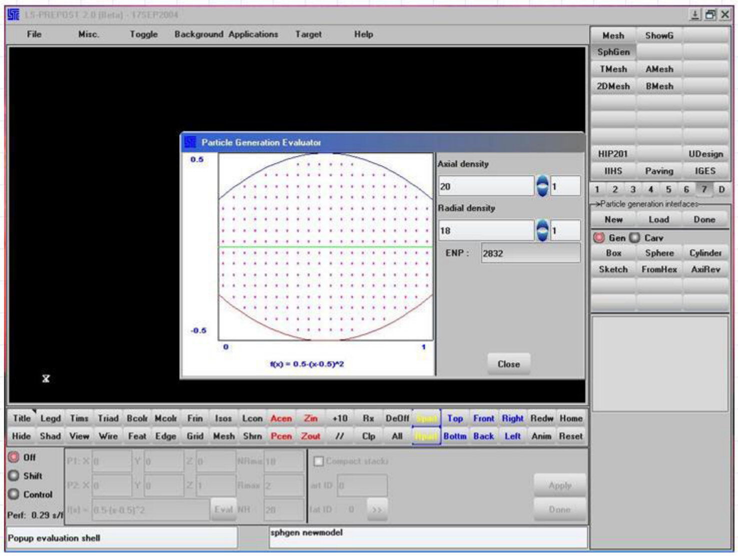
SPH switched from solids

Ls-prepost creates SPH elements



LSTC Livermore Software Technology Corp.

Axis of revolution SPH mesh


Particle Generation Evaluator

Axial density: 20 — 1

Radial density: 18 — 1

ENP : 2832

f(x) = 0.5-(x-0.5)^2

Popup evaluation shell

sphgen newmodel

LSTC Livermore Software Technology Corp.

Mesh-free developments

Mesh-free Basic Features

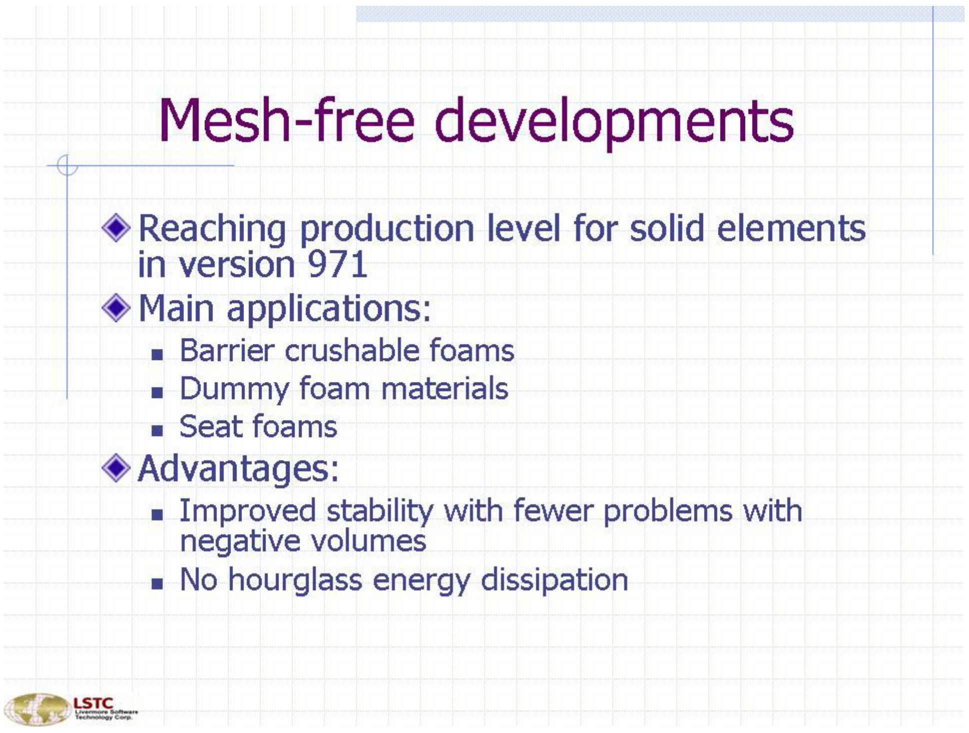
- Higher-order approximation
- Less sensitive to the discretization
- No hourglass control

- Higher CPU cost
- More difficult in theory
- Requires more theoretical developments and refinements

971 developments

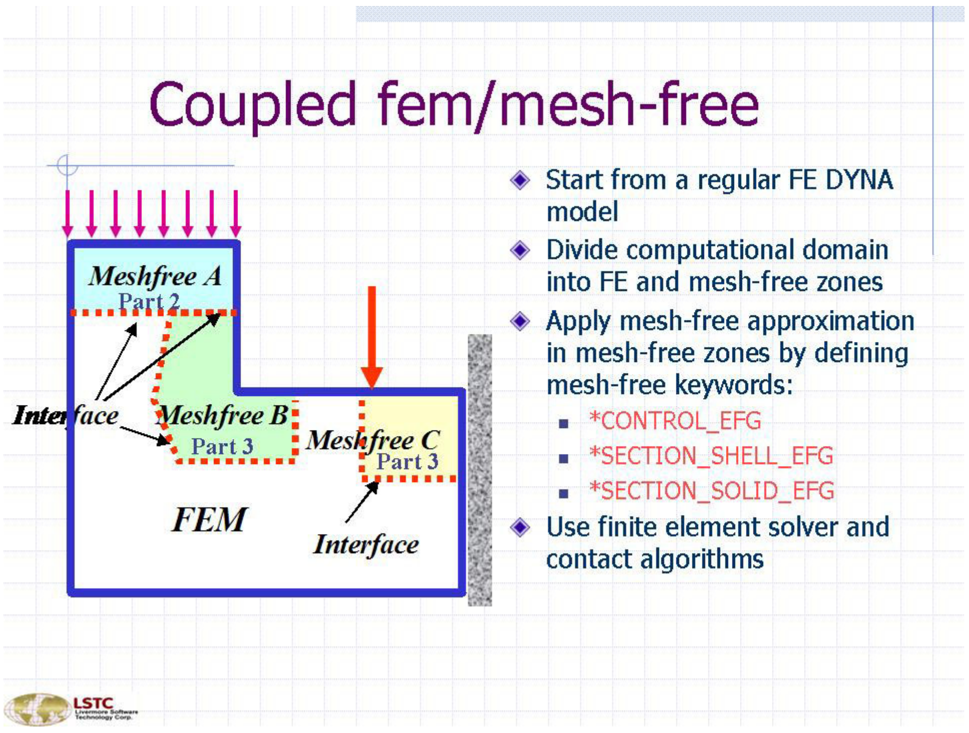
- Allow initial meshes based on prisms, tetrahedrons, degenerate solids
- Scalable MPI implementation

LSTC Livermore Software Technology Corp.

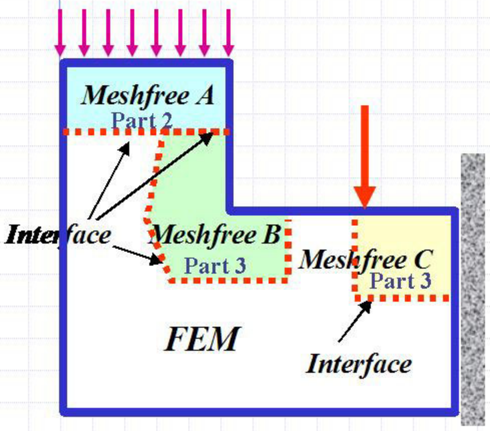


Mesh-free developments

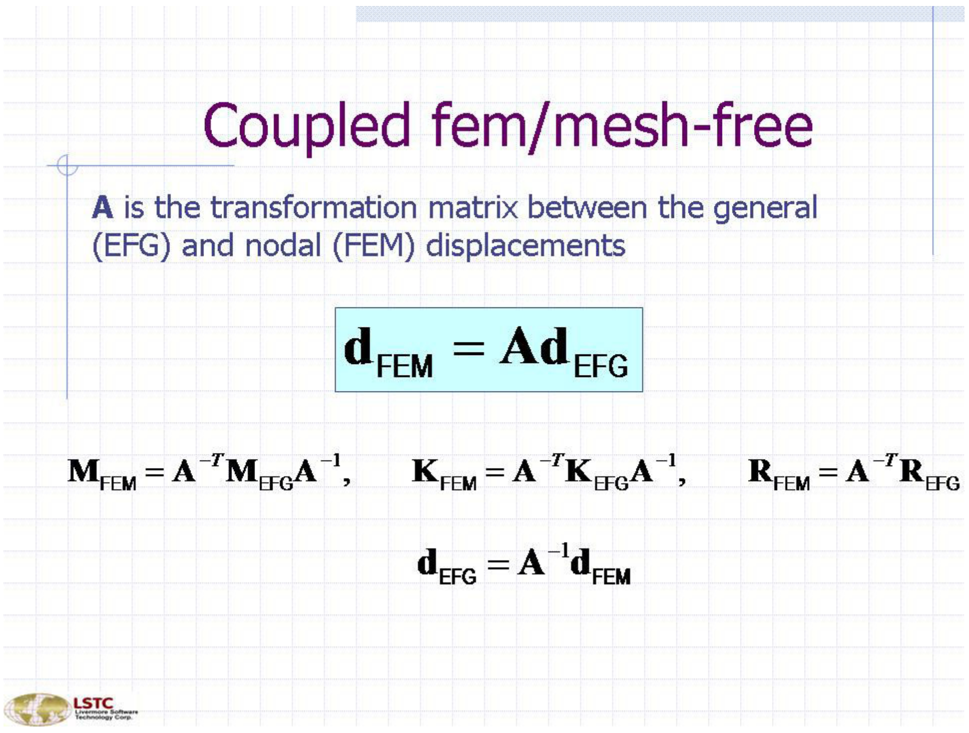
- Reaching production level for solid elements in version 971
- Main applications:
  - Barrier crushable foams
  - Dummy foam materials
  - Seat foams
- Advantages:
  - Improved stability with fewer problems with negative volumes
  - No hourglass energy dissipation



Coupled fem/mesh-free



- Start from a regular FE DYNA model
- Divide computational domain into FE and mesh-free zones
- Apply mesh-free approximation in mesh-free zones by defining mesh-free keywords:
  - *CONTROL_EFG
  - *SECTION_SHELL_EFG
  - *SECTION_SOLID_EFG
- Use finite element solver and contact algorithms



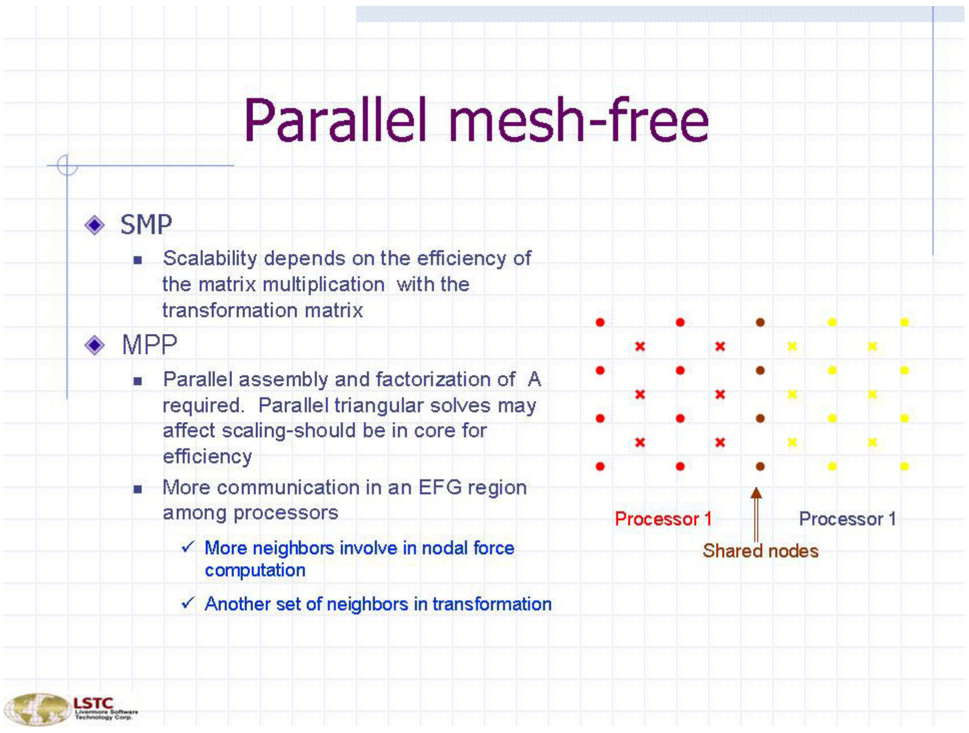
Coupled fem/mesh-free

A is the transformation matrix between the general (EFG) and nodal (FEM) displacements

$$\boxed{\mathbf{d}_{\mathrm{FEM}} = \mathbf{A}\mathbf{d}_{\mathrm{EFG}}}$$

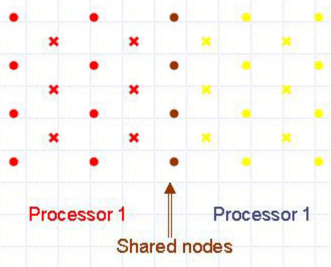
$$\mathbf{M}_{\mathrm{FEM}} = \mathbf{A}^{-T}\mathbf{M}_{\mathrm{EFG}}\mathbf{A}^{-1}, \quad \mathbf{K}_{\mathrm{FEM}} = \mathbf{A}^{-T}\mathbf{K}_{\mathrm{EFG}}\mathbf{A}^{-1}, \quad \mathbf{R}_{\mathrm{FEM}} = \mathbf{A}^{-T}\mathbf{R}_{\mathrm{EFG}}$$

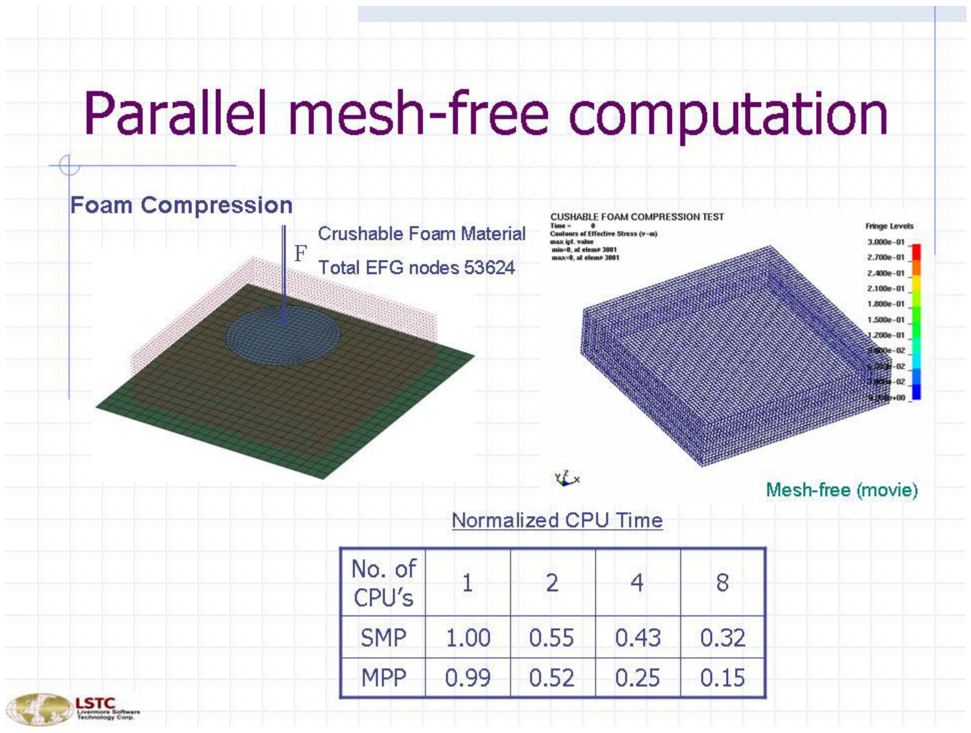
$$\mathbf{d}_{\mathrm{EFG}} = \mathbf{A}^{-1}\mathbf{d}_{\mathrm{FEM}}$$



Parallel mesh-free

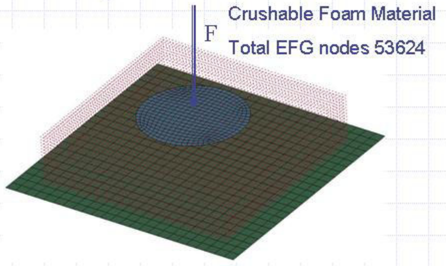
- SMP
  - Scalability depends on the efficiency of the matrix multiplication with the transformation matrix
- MPP
  - Parallel assembly and factorization of A required. Parallel triangular solves may affect scaling-should be in core for efficiency
  - More communication in an EFG region among processors
    - ✓ More neighbors involve in nodal force computation
    - ✓ Another set of neighbors in transformation

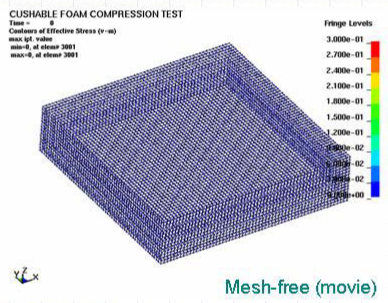




Parallel mesh-free computation

Foam Compression

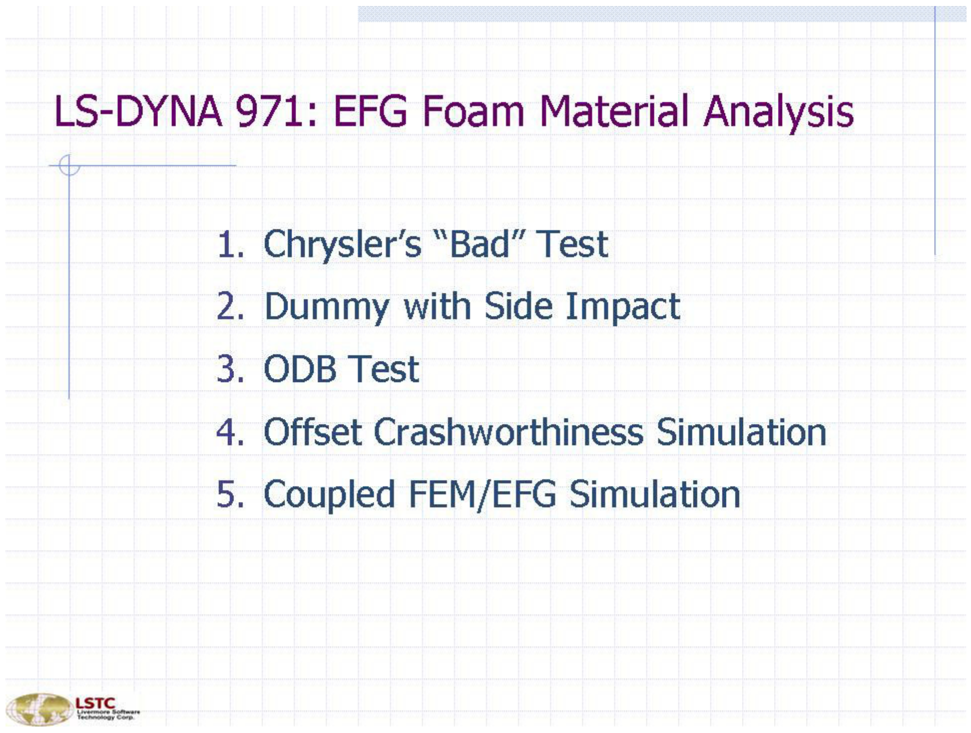




Mesh-free (movie)

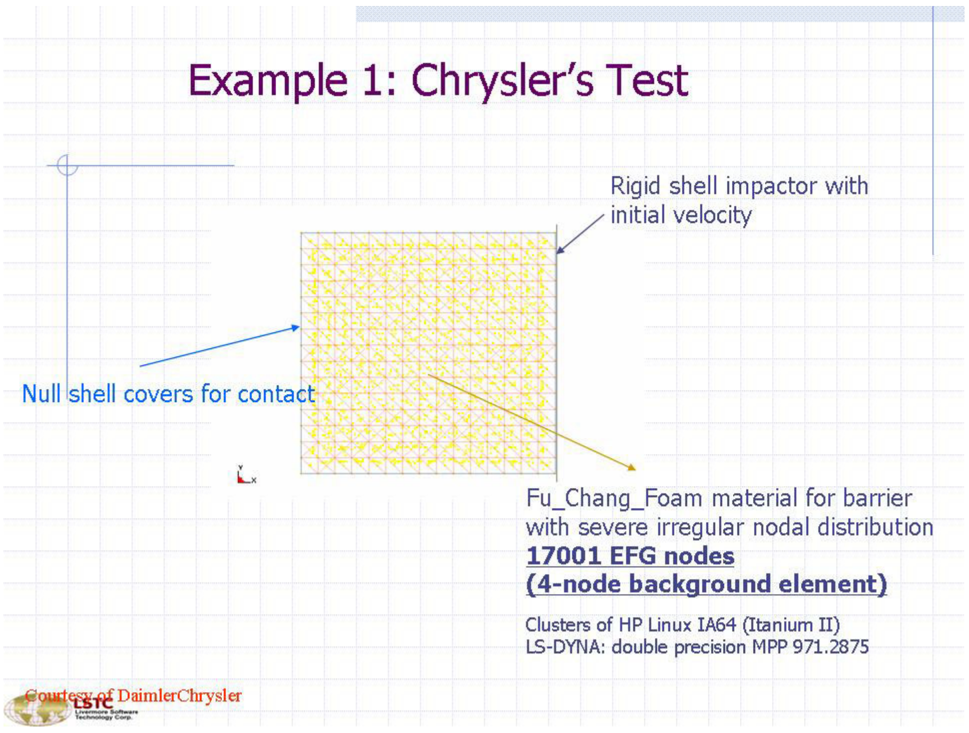
Normalized CPU Time

No. of CPU's	1	2	4	8
SMP	1.00	0.55	0.43	0.32
MPP	0.99	0.52	0.25	0.15



LS-DYNA 971: EFG Foam Material Analysis

1. Chrysler's "Bad" Test
2. Dummy with Side Impact
3. ODB Test
4. Offset Crashworthiness Simulation
5. Coupled FEM/EFG Simulation



Example 1: Chrysler's Test

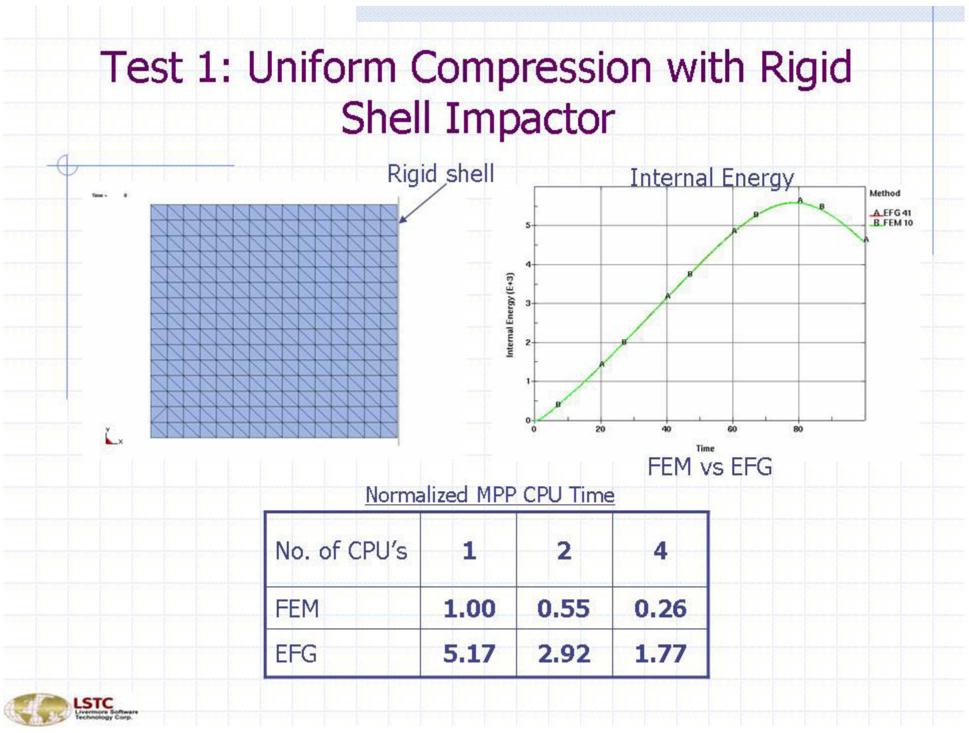
Rigid shell impactor with initial velocity

Null shell covers for contact

Fu_Chang_Foam material for barrier with severe irregular nodal distribution
17001 EFG nodes
(4-node background element)

Clusters of HP Linux IA64 (Itanium II)
LS-DYNA: double precision MPP 971.2875

Courtesy of DaimlerChrysler



Test 1: Uniform Compression with Rigid Shell Impactor

Rigid shell

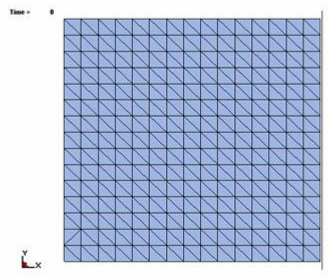
Internal Energy
Internal Energy (E+3)
Time
Method
A EFG 41
B FEM 10

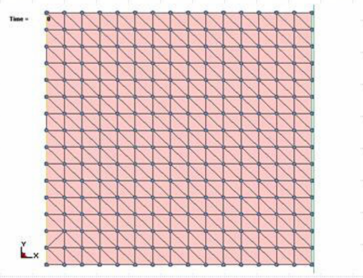
FEM vs EFG

Normalized MPP CPU Time

No. of CPU's	1	2	4
FEM	1.00	0.55	0.26
EFG	5.17	2.92	1.77

Test 1: Uniform Compression with Rigid Shell Impactor

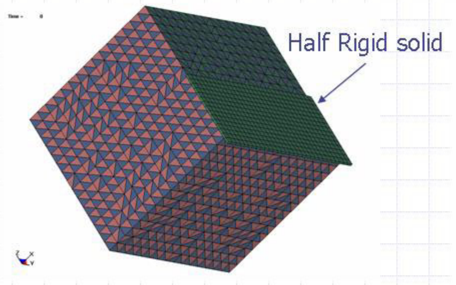


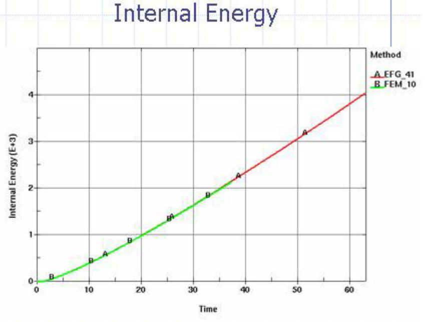


FEM AVI

EFG AVI

Test 2: Non-uniform Compression with Half Rigid Solid Impactor

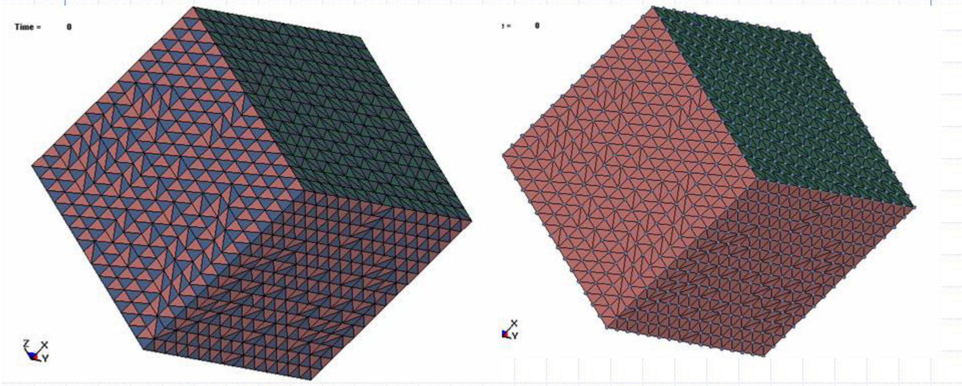




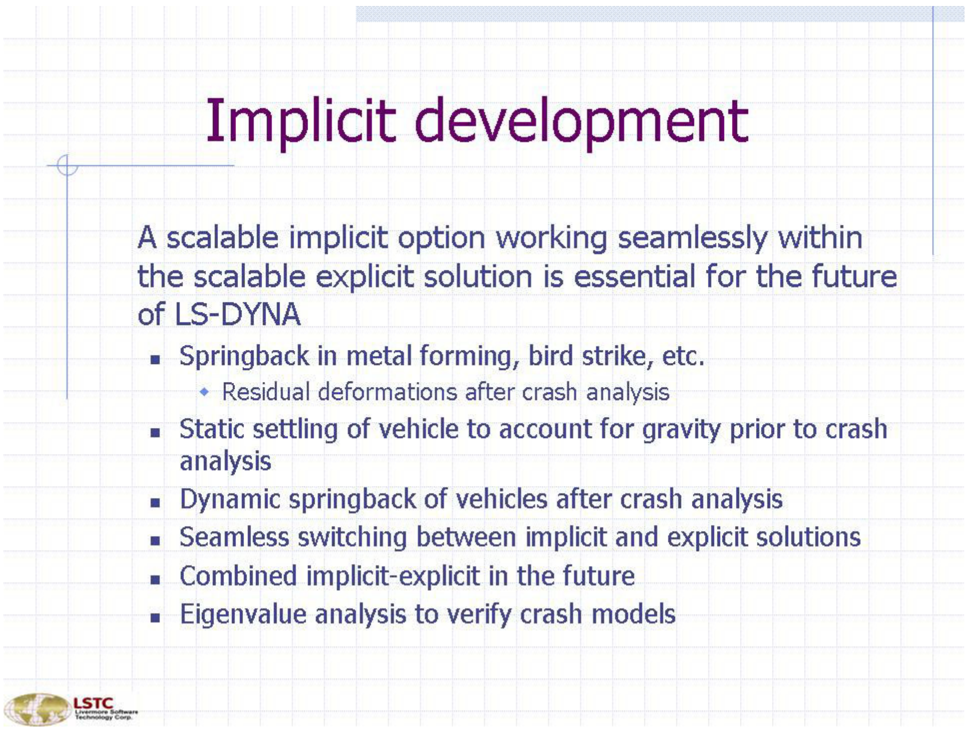
Normalized MPP CPU Time (up to FEM final step)

No. of CPU's	1	2	4
FEM	1.00	0.49	0.22
EFG	5.20	2.80	1.62

Test 2: Non-uniform Compression with Half Rigid Solid Impactor



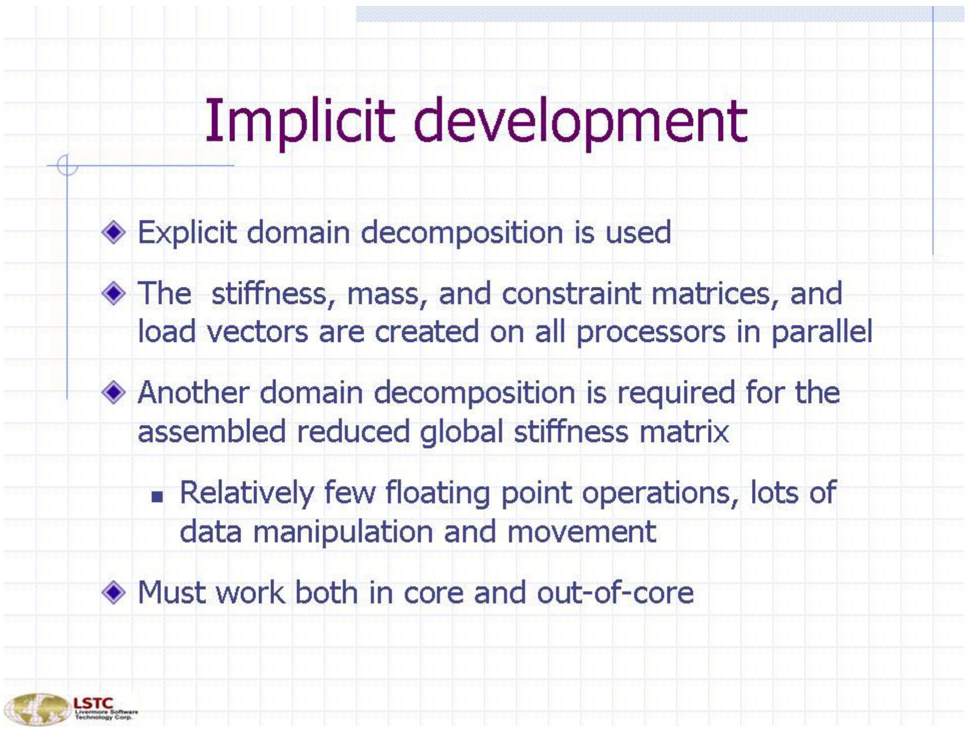
FEM AVI EFG AVI{



Implicit development

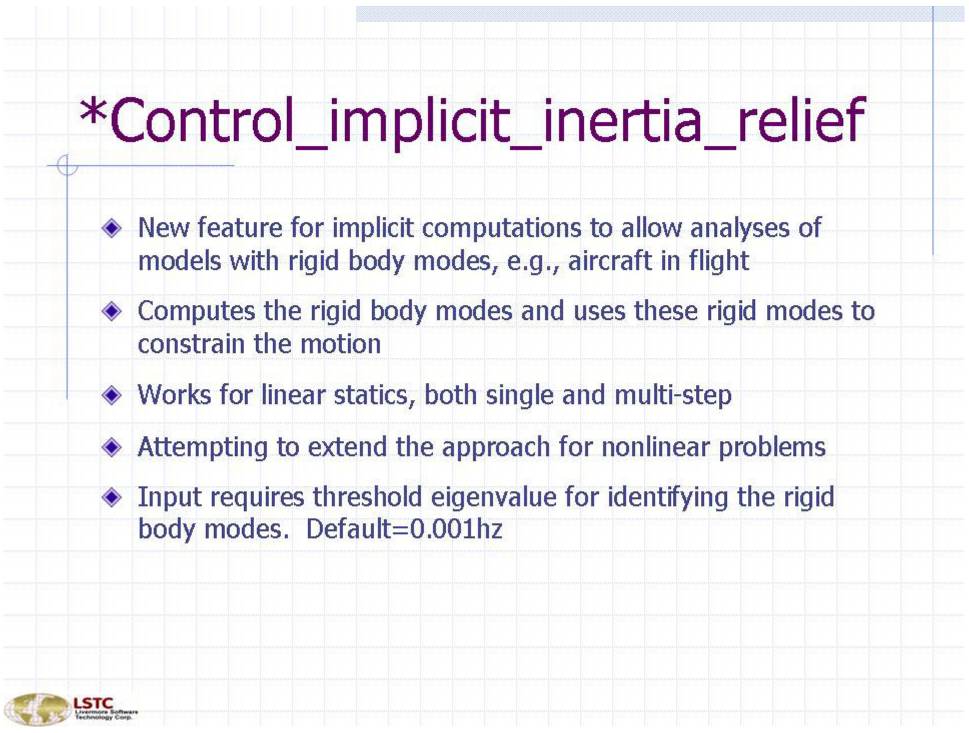
A scalable implicit option working seamlessly within the scalable explicit solution is essential for the future of LS-DYNA

- Springback in metal forming, bird strike, etc.
  - Residual deformations after crash analysis
- Static settling of vehicle to account for gravity prior to crash analysis
- Dynamic springback of vehicles after crash analysis
- Seamless switching between implicit and explicit solutions
- Combined implicit-explicit in the future
- Eigenvalue analysis to verify crash models



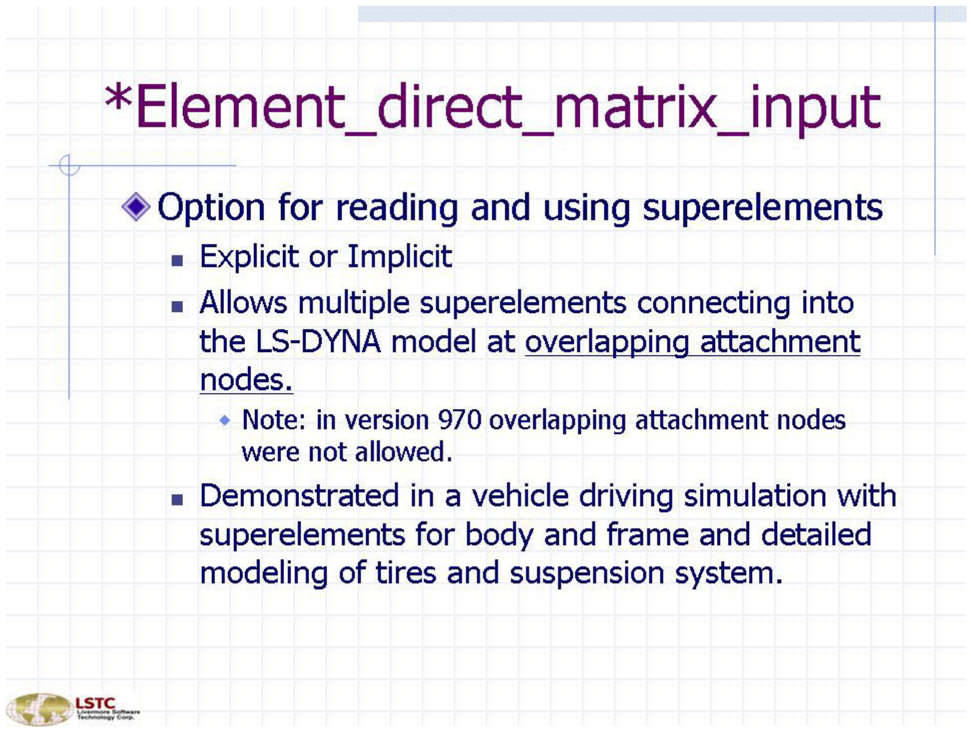
Implicit development

- Explicit domain decomposition is used
- The stiffness, mass, and constraint matrices, and load vectors are created on all processors in parallel
- Another domain decomposition is required for the assembled reduced global stiffness matrix
  - Relatively few floating point operations, lots of data manipulation and movement
- Must work both in core and out-of-core



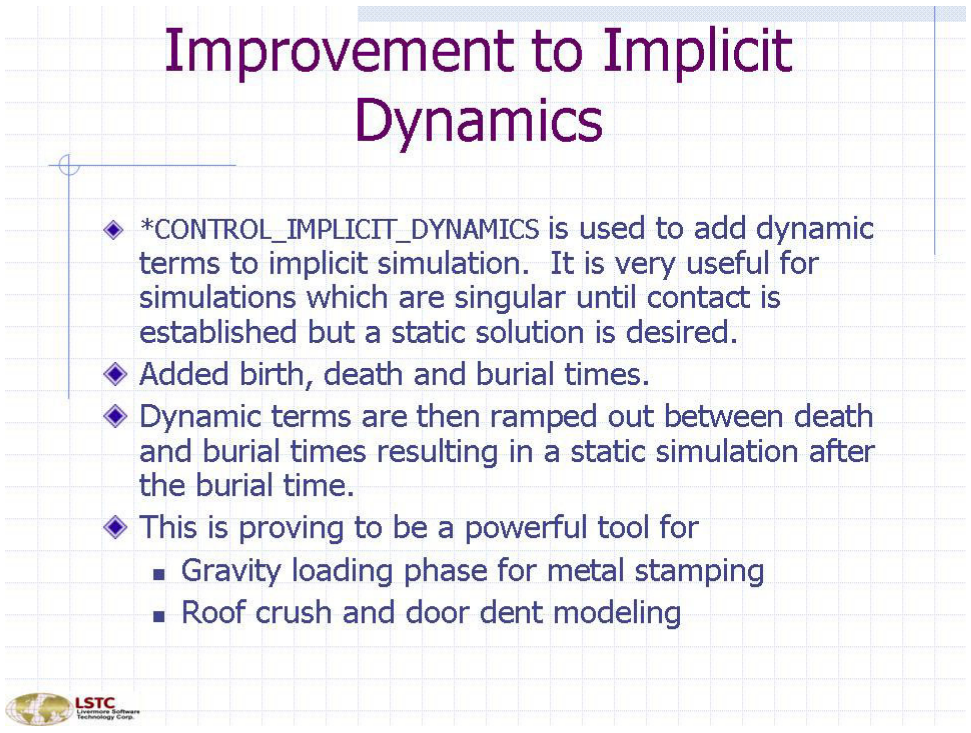
*Control_implicit_inertia_relief

- 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
- Attempting to extend the approach for nonlinear problems
- Input requires threshold eigenvalue for identifying the rigid body modes. Default=0.001hz



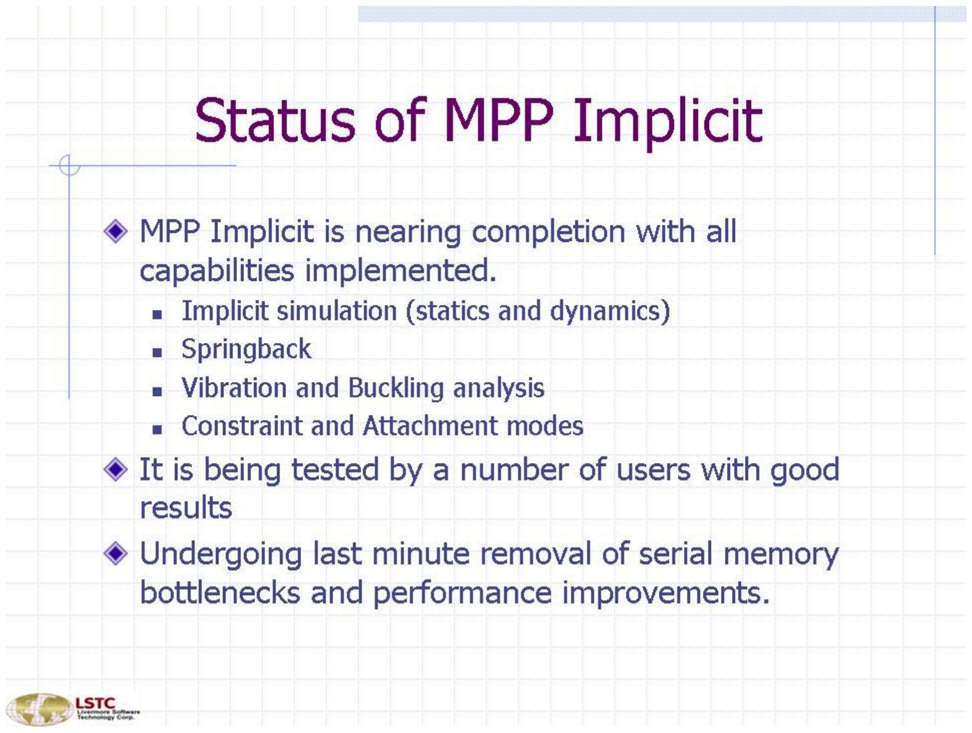
*Element_direct_matrix_input

- Option for reading and using superelements
  - Explicit or Implicit
  - Allows multiple superelements connecting into the LS-DYNA model at overlapping attachment nodes.
    - Note: in version 970 overlapping attachment nodes were not allowed.
  - Demonstrated in a vehicle driving simulation with superelements for body and frame and detailed modeling of tires and suspension system.



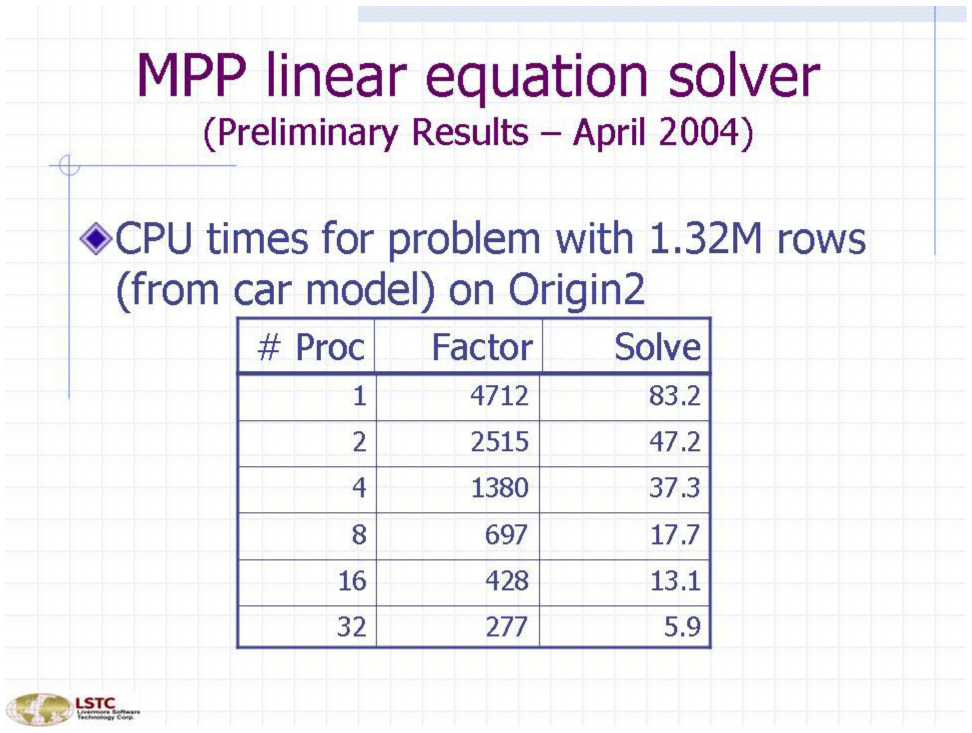
Improvement to Implicit Dynamics

- *CONTROL_IMPLICIT_DYNAMICS is used to add dynamic terms to implicit simulation. It is very useful for simulations which are singular until contact is established but a static solution is desired.
- Added birth, death and burial times.
- Dynamic terms are then ramped out between death and burial times resulting in a static simulation after the burial time.
- This is proving to be a powerful tool for
  - Gravity loading phase for metal stamping
  - Roof crush and door dent modeling



Status of MPP Implicit

- MPP Implicit is nearing completion with all capabilities implemented.
  - Implicit simulation (statics and dynamics)
  - Springback
  - Vibration and Buckling analysis
  - Constraint and Attachment modes
- It is being tested by a number of users with good results
- Undergoing last minute removal of serial memory bottlenecks and performance improvements.

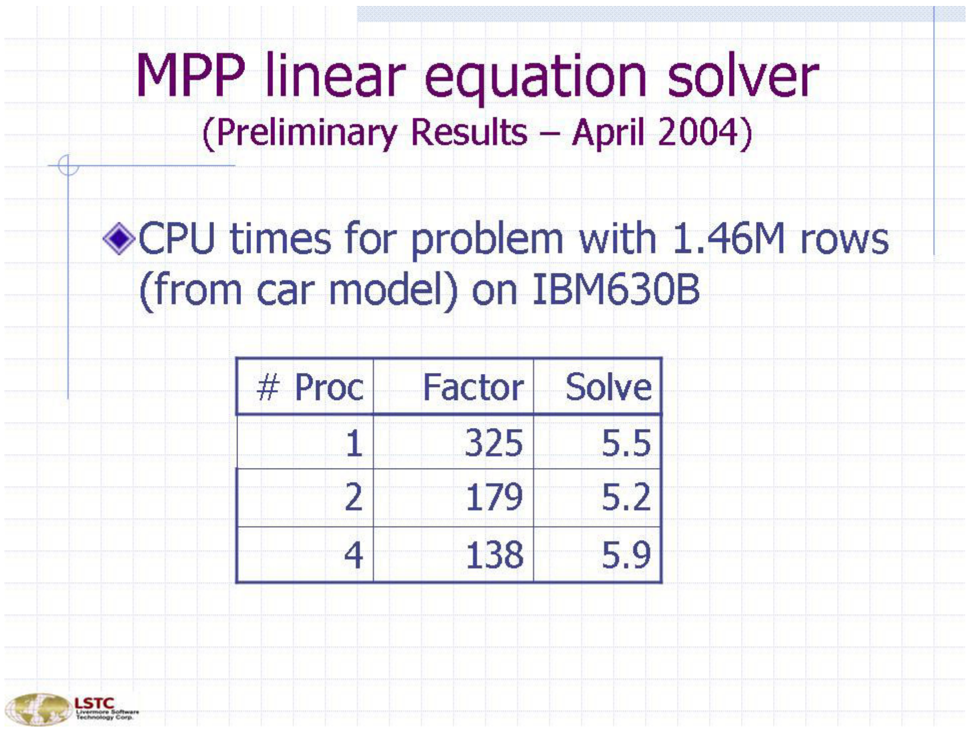


MPP linear equation solver

(Preliminary Results – April 2004)

- CPU times for problem with 1.32M rows (from car model) on Origin2

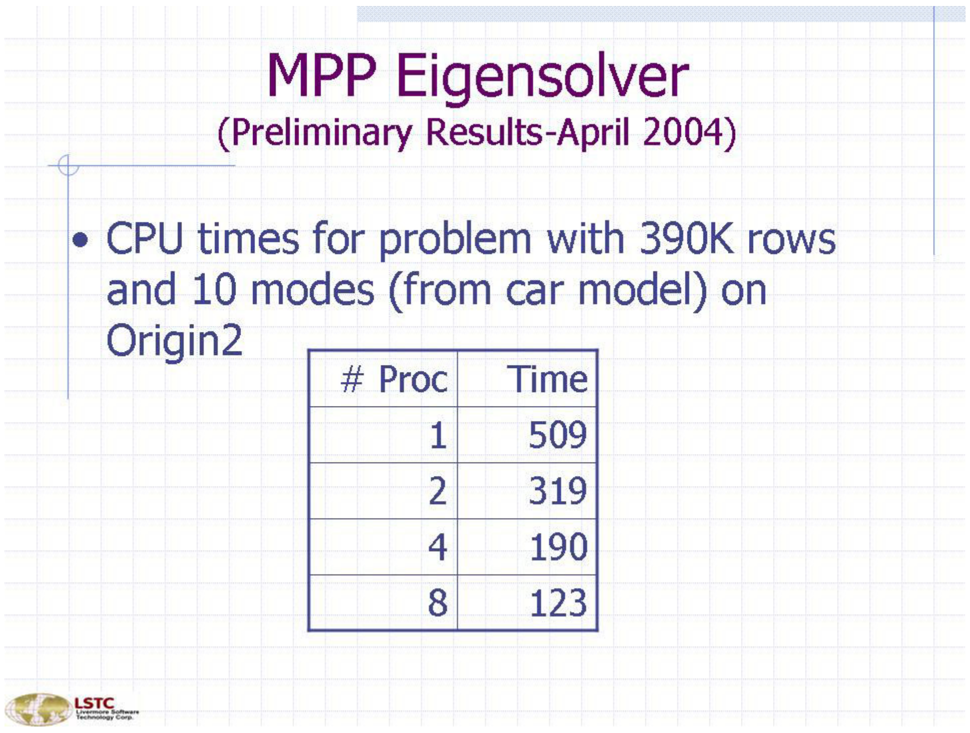
# Proc	Factor	Solve
1	4712	83.2
2	2515	47.2
4	1380	37.3
8	697	17.7
16	428	13.1
32	277	5.9



MPP linear equation solver
(Preliminary Results – April 2004)

◆CPU times for problem with 1.46M rows (from car model) on IBM630B

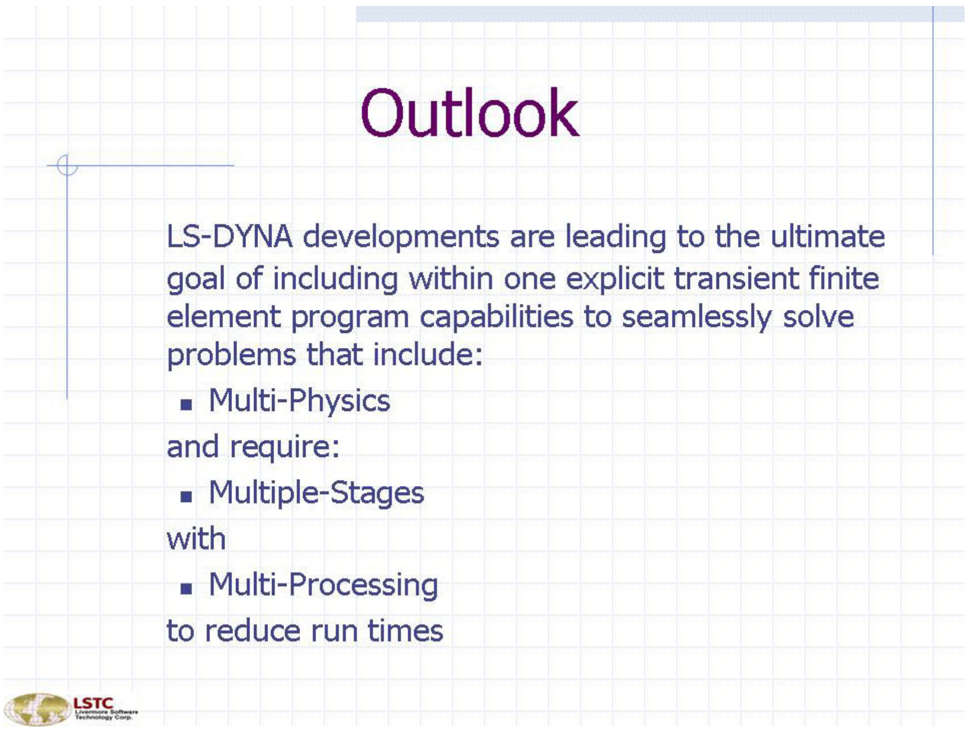
# Proc	Factor	Solve
1	325	5.5
2	179	5.2
4	138	5.9



MPP Eigensolver
(Preliminary Results-April 2004)

- CPU times for problem with 390K rows and 10 modes (from car model) on Origin2

# Proc	Time
1	509
2	319
4	190
8	123



Outlook

LS-DYNA developments are leading to the ultimate goal of including within one explicit transient finite element program capabilities to seamlessly solve problems that include:

- Multi-Physics

and require:

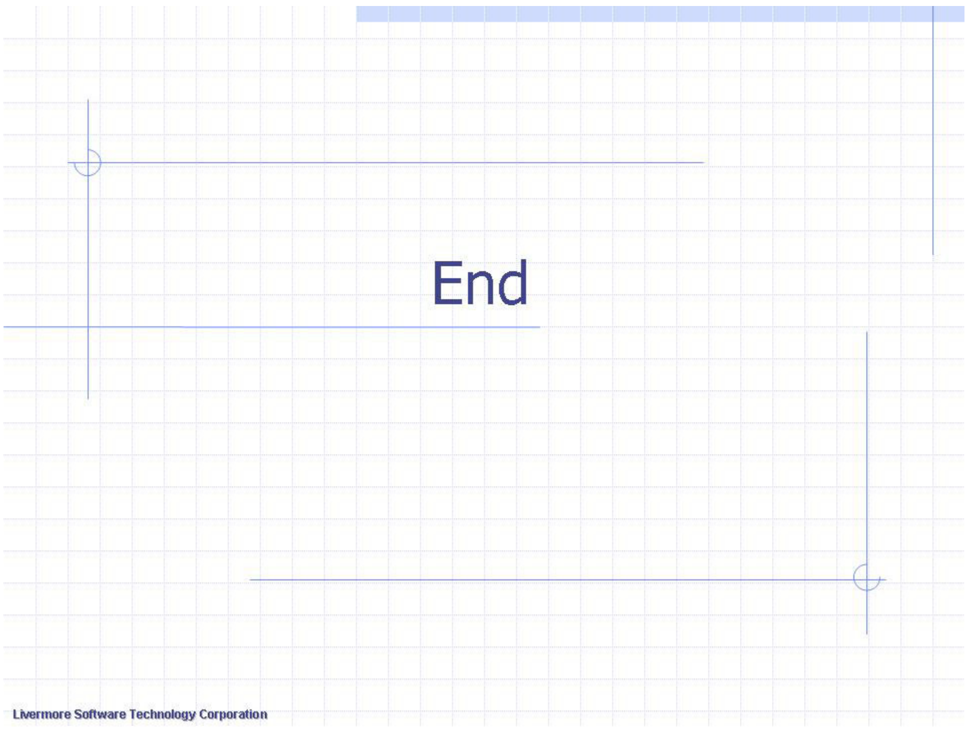
- Multiple-Stages

with

- Multi-Processing

to reduce run times

LSTC Livermore Software Technology Corp.



End

Livermore Software Technology Corporation