LS-DYNA
THEORY MANUAL

November 2005

Compiled By
John O. Hallquist

Livermore Software Technology Corporation
7374 Las Positas Road
Livermore, California 94551

FAX: 925-449-2507
TEL: 925-449-2500
EMAIL: sales@lstc.com
WEBSITE: www.lstc.com

TABLE OF CONTENTS

ABSTRACT

LS-DYNA is a general purpose finite element code for analyzing the large deformation static and dynamic response of structures including structures coupled to fluids. The main solution methodology is based on explicit time integration. An implicit solver is currently available with somewhat limited capabilities including structural analysis and heat transfer. A contact-impact algorithm allows difficult contact problems to be easily treated with heat transfer included across the contact interfaces. By a specialization of this algorithm, such interfaces can be rigidly tied to admit variable zoning without the need of mesh transition regions. Other specializations, allow draw beads in metal stamping applications to be easily modeled simply by defining a line of nodes along the draw bead. Spatial discretization is achieved by the use of four node tetrahedron and eight node solid elements, two node beam elements, three and four node shell elements, eight node solid shell elements, truss elements, membrane elements, discrete elements, and rigid bodies. A variety of element formulations are available for each element type. Specialized capabilities for airbags, sensors, and seatbelts have tailored LS-DYNA for applications in the automotive industry. Adaptive remeshing is available for shell elements and is widely used in sheet metal stamping applications. LS-DYNA currently contains approximately one-hundred constitutive models and ten equations-of-state to cover a wide range of material behavior.

This theoretical manual has been written to provide users and potential users with insight into the mathematical and physical basis of the code.

1. INTRODUCTION

1.1 History of LS-DYNA

The origin of LS-DYNA dates back to the public domain software, DYNA3D, which was developed in the mid-seventies at the Lawrence Livermore National Laboratory.

The first version of DYNA3D [Hallquist 1976a] was released in 1976 with constant stress 4- or 8-node solid elements, 16- and 20-node solid elements with 2 x 2 x 2 Gaussian quadrature, 3, 4, and 8-node membrane elements, and a 2-node cable element. A nodal constraint contact-impact interface algorithm [Hallquist 1977] was available. On the Control Data CDC-7600, a supercomputer in 1976, the speed of the code varied from 36 minutes per 10^6 mesh cycles with 4-8 node solids to 180 minutes per 10^6 mesh cycles with 16 and 20 node solids. Without hourglass control to prevent formation of non-physical zero energy deformation modes, constant stress solids were processed at 12 minutes per 10^6 mesh cycles. A moderate number of very costly solutions were obtained with this version of DYNA3D using 16- and 20-node solids. Hourglass modes combined with the procedure for computing the time step size prevented us from obtaining solutions with constant stress elements.

In this early development, several things became apparent. Hourglass deformation modes of the constant stress elements were invariably excited by the contact-impact algorithm, showing that a new sliding interface algorithm was needed. Higher order elements seemed to be impractical for shock wave propagation because of numerical noise resulting from the *ad hoc* mass lumping necessary to generate a diagonal mass matrix. Although the lower frequency structural response was accurately computed with these elements, their high computer cost made analysis so expensive as to be impractical. It was obvious that realistic three-dimensional

structural calculations were possible, if and only if the under-integrated eight node constant stress solid element could be made to function. This implied a need for a much better sliding interface algorithm, a more cost-effective hourglass control, more optimal programming, and a machine much faster than the CDC-7600. This latter need was fulfilled several years later when LLNL took deliver of its first CRAY-1. At this time, DYNA3D was completely rewritten.

The next version, released in 1979, achieved the aforementioned goals. On the CRAY the vectorized speed was 50 times faster, 0.67 minutes per million mesh cycles. A symmetric, penalty-based, contact-impact algorithm was considerably faster in execution speed and exceedingly reliable. Due to lack of use, the membrane and cable elements were stripped and all higher order elements were eliminated as well. Wilkins' finite difference equations [Wilkins et al. 1974] were implemented in unvectorized form in an overlay to compare their performance with the finite element method. The finite difference algorithm proved to be nearly two times more expensive than the finite element approach (apart from vectorization) with no compensating increase in accuracy, and was removed in the next code update.

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

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

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

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

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

In 1988 the Hallquist began working half-time at LLNL to devote more time to the development and support of LS-DYNA for automotive applications. By the end of 1988 it was obvious that a much more concentrated effort would be required in the development of

LS-DYNA if problems in crashworthiness were to be properly solved; therefore, at the start of 1989 the Hallquist resigned from LLNL to continue code development full time at Livermore Software Technology Corporation. The 1989 version introduced many enhanced capabilities including a one-way treatment of slide surfaces with voids and friction; cross-sectional forces for structural elements; an optional user specified minimum time step size for shell elements using elastic and elastoplastic material models; nodal accelerations in the time history database; a compressible Mooney-Rivlin material model; a closed-form update shell plasticity model; a general rubber material model; unique penalty specifications for each slide surface; external work tracking; optional time step criterion for 4-node shell elements; and internal element sorting to allow full vectorization of right-hand-side force assembly.

Throughout the past decade, considerable progress has been made as may be seen in the chronology of the developments which follows. During 1989 many extensions and developments were completed, and in 1990 the following capabilities were delivered to users:

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

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

Options added in 1991-1992:

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

Options added in 1993-1994:

- Arbitrary Lagrangian Eulerian brick elements,
- Belytschko-Wong-Chiang quadrilateral shell element,
- Warping stiffness in the Belytschko-Tsay shell element,
- Fast Hughes-Liu shell element,
- Fully integrated brick shell element,
- Discrete 3D beam element,
- Generalized dampers,
- Cable modeling,
- Airbag reference geometry,
- Multiple jet model,

- Generalized joint stiffnesses,
- Enhanced rigid body to rigid body contact,
- Orthotropic rigid walls,
- Time zero mass scaling,
- Coupling with USA (Underwater Shock Analysis),
- Layered spot welds with failure based on resultants or plastic strain,
- Fillet welds with failure,
- Butt welds with failure,
- Automatic eroding contact,
- Edge-to-edge contact,
- Automatic mesh generation with contact entities,
- Drawbead modeling,
- Shells constrained inside brick elements,
- NIKE3D coupling for springback,
- Barlat's anisotropic plasticity,
- Superplastic forming option,
- Rigid body stoppers,
- Keyword input,
- Adaptivity,
- First MPP (Massively Parallel) version with limited capabilities.
- Built in least squares fit for rubber model constitutive constants,
- Large hystersis in hyperelastic foam,
- Bilhku/Dubois foam model,
- Generalized rubber model,

New options added to version 936 in 1995 include:

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

- More user control over contact defaults.
- Improved interpenetration warnings printed in automatic contact.
- Flag for using actual shell thickness in single surface contact logic rather than the default.
- Definition by exempted part ID's.
- Airbag to Airbag venting/segmented airbags are now supported.
- Airbag reference geometry speed improvements by using the reference geometry for the time step size calculation.
- Isotropic airbag material may now be directly for cost efficiency.
- Airbag fabric material damping is now specified as the ratio of critical damping.
- Ability to attach jets to the structure so the airbag, jets, and structure to move together.
- PVM 5.1 Madymo coupling is available.
- Meshes are generated within LS-DYNA3D for all standard contact entities.
- Joint damping for translational motion.
- Angular displacements, rates of displacements, damping forces, etc. in JNTFORC file.
- Link between LS-NIKE3D to LS-DYNA3D via *INITIAL_STRESS keywords.
- Trim curves for metal forming springback.
- Sparse equation solver for springback.
- Improved mesh generation for IGES and VDA provides a mesh that can directly be used to model tooling in metal stamping analyses.

New options added to Version 940 in 1996 and 1997:

- Part/Material ID's may be specified with 8 digits.
- Rigid body motion can be prescribed in a local system fixed to the rigid body.
- Nonlinear least squares fit available for the Ogden rubber model.
- Lease squares fit to the relaxation curves for the viscoelasticity in rubber.
- Fu-Chang rate sensitive foam.
- 6 term Prony series expansion for rate effects in model 57-now 73
- Viscoelastic material model 76 implemented for shell elements.
- Mechanical threshold stress (MTS) plasticity model for rate effects.
- Thermoelastic-plastic material model for Hughes-Liu beam element.
- Ramberg-Osgood soil model
- Invariant local coordinate systems for shell elements are optional.
- Second order accurate stress updates.
- Four-noded, linear, tetrahedron element.
- Co-rotational solid element for foam that can invert without stability problems.
- Improved speed in rigid body to rigid body contacts.
- Improved searching for the a_3, a_5 and a10 contact types.
- Invariant results on shared memory parallel machines with the *a_n* contact types.
- Thickness offsets in type 8 and 9 tie break contact algorithms.
- Bucket sort frequency can be controlled by a load curve for airbag applications.
- In automatic contact each part ID in the definition may have unique:
  -Static coefficient of friction
  -Dynamic coefficient of friction
  -Exponential decay coefficient

-Viscous friction coefficient
-Optional contact thickness
-Optional thickness scale factor
-Local penalty scale factor

- Automatic beam-to-beam, shell edge-to-beam, shell edge-to-shell edge and single surface contact algorithm.
- Release criteria may be a multiple of the shell thickness in types a_3, a_5, a10, 13, and 26 contact.
- Force transducers to obtain reaction forces in automatic contact definitions. Defined manually via segments, or automatically via part ID's.
- Searching depth can be defined as a function of time.
- Bucket sort frequency can be defined as a function of time.
- Interior contact for solid (foam) elements to prevent "negative volumes."
- Locking joint
- Temperature dependent heat capacity added to Wang-Nefske inflator models.
- Wang Hybrid inflator model [Wang, 1996] with jetting options and bag-to-bag venting.
- Aspiration included in Wang's hybrid model [Nucholtz, Wang, Wylie, 1996].
- Extended Wang's hybrid inflator with a quadratic temperature variation for heat capacities [Nusholtz, 1996].
- Fabric porosity added as part of the airbag constitutive model.
- Blockage of vent holes and fabric in contact with structure or itself considered in venting with leakage of gas.
- Option to delay airbag liner with using the reference geometry until the reference area is reached.
- Birth time for the reference geometry.
- Multi-material Euler/ALE fluids,
  -2nd order accurate formulations.
  -Automatic coupling to shell, brick, or beam elements
  -Coupling using LS-DYNA contact options.
  -Element with fluid + void and void material
  -Element with multi-materials and pressure equilibrium
- Nodal inertia tensors.
- 2D plane stress, plane strain, rigid, and axisymmetric elements
- 2D plane strain shell element
- 2D axisymmetric shell element.
- Full contact support in 2D, tied, sliding only, penalty and constraint techniques.
- Most material types supported for 2D elements.
- Interactive remeshing and graphics options available for 2D.
- Subsystem definitions for energy and momentum output.

and many more enhancements not mentioned above.

Capabilities added during 1997-1998 in Version 950 include:

- Adaptive refinement can be based on tooling curvature with FORMING contact.
- The display of draw beads is now possible since the draw bead data is output into the D3PLOT database.

- An adaptive box option, *DEFINE_BOX_ADAPTIVE, allows control over the refinement level and location of elements to be adapted.
- A root identification file, ADAPT.RID, gives the parent element ID for adapted elements.
- Draw bead box option, *DEFINE_BOX_DRAWBEAD, simplifies draw bead input.
- The new control option, CONTROL_IMPLICIT, activates an implicit solution scheme.
- 2D Arbitrary-Lagrangian-Eulerian elements.
- 2D automatic contact is defined by listing part ID's.
- 2D r-adaptivity for plane strain and axisymmetric forging simulations is available.
- 2D automatic non-interactive rezoning as in LS-DYNA2D.
- 2D plane strain and axisymmetric element with 2x2 selective-reduced integration are implemented.
- Implicit 2D solid and plane strain elements are available.
- Implicit 2D contact is available.
- The new keyword, *DELETE_CONTACT_2DAUTO, allows the deletion of 2D automatic contact definitions.
- The keyword, *LOAD_BEAM is added for pressure boundary conditions on 2D elements.
- A viscoplastic strain rate option is available for materials:
  - *MAT_PLASTIC_KINEMATIC
  - *MAT_JOHNSON_COOK
  - *MAT_POWER_LAW_PLASTICITY
  - *MAT_STRAIN_RATE_DEPENDENT_PLASTICITY
  - *MAT_PIECEWISE_LINEAR_PLASTICITY
  - *MAT_RATE_SENSITIVE_POWERLAW_PLASTICITY
  - *MAT_ZERILLI-ARMSTRONG
  - *MAT_PLASTICITY_WITH_DAMAGE
  - *MAT_PLASTICITY_COMPRESSION_TENSION
- Material model, *MAT_PLASTICITY_WITH_DAMAGE, has a piecewise linear damage curve given by a load curve ID.
- The Arruda-Boyce hyper-viscoelastic rubber model is available, see *MAT_ ARRUDA_BOYCE.
- Transverse-anisotropic-viscoelastic material for heart tissue, see *MAT_HEART_ TISSUE.
- Lung hyper-viscoelastic material, see *MAT_LUNG_TISSUE.
- Compression/tension plasticity model, see *MAT_PLASTICITY_COMPRESSION_ TENSION.
- The Lund strain rate model, *MAT_STEINBERG_LUND, is added to Steinberg-Guinan plasticity model.
- Rate sensitive foam model, *MAT_FU_CHANG_FOAM, has been extended to include engineering strain rates, etc.
- Model, *MAT_MODIFIED_PIECEWISE_LINEAR_PLASTICITY, is added for modeling the failure of aluminum.
- Material model, *MAT_SPECIAL_ ORTHOTROPIC, added for television shadow mask problems.
- Erosion strain is implemented for material type, *MAT_BAMMAN_DAMAGE.

- The equation of state, *EOS_JWLB, is available for modeling the expansion of explosive gases.
- The reference geometry option is extended for foam and rubber materials and can be used for stress initialization, see *INITIAL_FOAM_REFERENCE_GEOMETRY.
- A vehicle positioning option is available for setting the initial orientation and velocities, see *INITIAL_VEHICLE_KINEMATICS.
- A boundary element method is available for incompressible fluid dynamics problems.
- The thermal materials work with instantaneous coefficients of thermal expansion:
  *MAT_ELASTIC_PLASTIC_THERMAL
  *MAT_ORTHOTROPIC_THERMAL
  *MAT_TEMPERATURE_DEPENDENT_ORTHOTROPIC
  *MAT_ELASTIC_WITH_VISCOSITY.
- Airbag interaction flow rate versus pressure differences.
- Contact segment search option, [bricks first optional]
- A through thickness Gauss integration rule with 1-10 points is available for shell elements. Previously, 5 were available.
- Shell element formulations can be changed in a full deck restart.
- The tied interface which is based on constraint equations, TIED_SURFACE_TO_SURFACE, can now fail if _FAILURE, is appended.
- A general failure criteria for solid elements is independent of the material type, see *MAT_ADD_EROSION
- Load curve control can be based on thinning and a flow limit diagram, see *DEFINE_CURVE_FEEDBACK.
- An option to filter the spotweld resultant forces prior to checking for failure has been added the option, *CONSTRAINED_SPOTWELD, by appending, _FILTERED_FORCE, to the keyword.
- Bulk viscosity is available for shell types 1, 2, 10, and 16.
- When defining the local coordinate system for the rigid body inertia tensor a local coordinate system ID can be used. This simplifies dummy positioning.
- Prescribing displacements, velocities, and accelerations is now possible for rigid body nodes.
- One-way flow is optional for segmented airbag interactions.
- Pressure time history input for airbag type, LINEAR_FLUID, can be used.
- An option is available to independently scale system damping by part ID in each of the global directions.
- An option is available to independently scale global system damping in each of the global directions.
- Added option to constrain global DOF along lines parallel with the global axes. The keyword is *CONSTRAINED_GLOBAL. This option is useful for adaptive remeshing.
- Beam end code releases are available, see *ELEMENT_BEAM.
- An initial force can be directly defined for the cable material, *MAT_CABLE_DISCRETE_BEAM. The specification of slack is not required if this option is used.
- Airbag pop pressure can be activated by accelerometers.
- Termination may now be controlled by contact, via *TERMINATION_CONTACT.
- Modified shell elements types 8, 10 and the warping stiffness option in the Belytschko-Tsay shell to ensure orthogonality with rigid body motions in the event

that the shell is badly warped. This is optional in the Belytschko-Tsay shell and the type 10 shell.

- A one point quadrature brick element with an exact hourglass stiffness matrix has been implemented for implicit and explicit calculations.
- Automatic file length determination for D3PLOT binary database is now implemented. This insures that at least a single state is contained in each D3PLOT file and eliminates the problem with the states being split between files.
- The dump files, which can be very large, can be placed in another directory by specifying *d=/home/user /test/d3dump* on the execution line.
- A print flag controls the output of data into the MATSUM and RBDOUT files by part ID's. The option, PRINT, has been added as an option to the *PART keyword.
- Flag has been added to delete material data from the D3THDT file. See *DATABASE_ EXTENT_BINARY and column 25 of the 19th control card in the structured input.
- After dynamic relaxation completes, a file is written giving the displaced state which can be used for stress initialization in later runs.

Capabilities added during 1998-2000 in Version 960. Most new capabilities work on both the MPP and SMP versions; however, the capabilities that are implemented for the SMP version only, which were not considered critical for this release, are flagged below. These SMP unique capabilities are being extended for MPP calculations and will be available in the near future. The implicit capabilities for MPP require the development of a scalable eigenvalue solver, which is under development for a later release of LS-DYNA.

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

- Reference geometry for material, *MAT_MODIFIED_HONEYCOMB, can be set at arbitrary relative volumes or when the time step size reaches a limiting value. This option is now available for all element types including the fully integrated solid element.
- Non orthogonal material axes are available in the airbag fabric model. See *MAT_FABRIC.
- Other new constitutive models include for the beam elements:
  *MAT_MODIFIED_FORCE_LIMITED
  *MAT_SEISMIC_BEAM
  *MAT_CONCRETE_BEAM
  for shell and solid elements:
  *MAT_ELASTIC_VISCOPLASTIC_THERMAL
  for the shell elements:
  *MAT_GURSON
  *MAT_GEPLASTIC_SRATE2000
  *MAT_ELASTIC_VISCOPLASTIC_THERMAL
  *MAT_COMPOSITE_LAYUP
  *MAT_COMPOSITE_LAYUP
  *MAT_COMPOSITE_DIRECT
  for the solid elements:
  *MAT_JOHNSON_HOLMQUIST_CERAMICS
  *MAT_JOHNSON_HOLMQUIST_CONCRETE
  *MAT_INV_HYPERBOLIC_SIN
  *MAT_UNIFIED_CREEP
  *MAT_SOIL_BRICK
  *MAT_DRUCKER_PRAGER
  *MAT_RC_SHEAR_WALL
  and for all element options a very fast and efficient version of the Johnson-Cook plasticity model is available:
  *MAT_SIMPLIFIED_JOHNSON_COOK
- A fully integrated version of the type 16 shell element is available for the resultant constitutive models.
- A nonlocal failure theory is implemented for predicting failure in metallic materials. The keyword *MAT_NONLOCAL activates this option for a subset of elastoplastic constitutive models.
- A discrete Kirchhoff triangular shell element (DKT) for explicit analysis with three in plane integration points is flagged as a type 17 shell element. This element has much better bending behavior than the C0 triangular element.
- A discrete Kirchhoff linear triangular and quadrilaterial shell element is available as a type 18 shell. This shell is for extracting normal modes and static analysis.
- A C0 linear 4-node quadrilaterial shell element is implemented as element type 20 with drilling stiffness for normal modes and static analysis.
- An assumed strain linear brick element is available for normal modes and statics.
- The fully integrated thick shell element has been extended for use in implicit calculations.

- A fully integrated thick shell element based on an assumed strain formulation is now available. This element uses a full 3D constitutive model which includes the normal stress component and, therefore, does not use the plane stress assumption.
- The 4-node constant strain tetrahedron element has been extended for use in implicit calculations.
- Relative damping between parts is available, see *DAMPING_RELATIVE (SMP only).
- Preload forces are can be input for the discrete beam elements.
- Objective stress updates are implemented for the fully integrated brick shell element.
- Acceleration time histories can be prescribed for rigid bodies.
- Prescribed motion for nodal rigid bodies is now possible.
- Generalized set definitions, i.e., SET_SHELL_GENERAL etc. provide much flexibility in the set definitions.
- The command "sw4." will write a state into the dynamic relaxation file, D3DRLF, during the dynamic relaxation phase if the D3DRLF file is requested in the input.
- Added mass by PART ID is written into the MATSUM file when mass scaling is used to maintain the time step size, (SMP version only).
- Upon termination due to a large mass increase during a mass scaled calculation a print summary of 20 nodes with the maximum added mass is printed.
- Eigenvalue analysis of models containing rigid bodies is now available using BCSLIB-EXT solvers from Boeing. (SMP version only).
- Second order stress updates can be activated by part ID instead of globally on the *CONTROL_ACCURACY input.
- Interface frictional energy is optionally computed for heat generation and is output into the interface force file (SMP version only).
- The interface force binary database now includes the distance from the contact surface for the FORMING contact options. This distance is given after the nodes are detected as possible contact candidates. (SMP version only).
- Type 14 acoustic brick element is implemented. This element is a fully integrated version of type 8, the acoustic element (SMP version only).
- A flooded surface option for acoustic applications is available (SMP version only).
- Attachment nodes can be defined for rigid bodies. This option is useful for NVH applications.
- CONSTRAINED_POINTS tie any two points together. These points must lie on a shell element.
- Soft constraint is available for edge-to-edge contact in type 26 contact.
- CONSTAINED_INTERPOLATION option for beam to solid interfaces and for spreading the mass and loads. (SMP version only).
- A database option has been added that allows the output of added mass for shell elements instead of the time step size.
- A new contact option allows the inclusion of all internal shell edges in contact type *CONTACT_GENERAL, type 26. This option is activated by adding _INTERIOR after the GENERAL keyword.
- A new option allows the use deviatoric strain rates rather than total rates in material model 24 for the Cowper-Symonds rate model.
- The CADFEM option for ASCII databases is now the default. Their option includes more significant figures in the output files.

- When using deformable spot welds, the added mass for spot welds is now printed for the case where global mass scaling is activated. This output is in the log file, D3HSP file, and the MESSAG file.
- Initial penetration warnings for edge-to-edge contact are now written into the MESSAG file and the D3HSP file.
- Each compilation of LS-DYNA is given a unique version number.
- Finite length discrete beams with various local axes options are now available for material types 66, 67, 68, 93, and 95. In this implementation the absolute value of SCOOR must be set to 2 or 3 in the *SECTION_BEAM input.
- New discrete element constitutive models are available:
  *MAT_ELASTIC_SPRING_DISCRETE_BEAM
  *MAT_INELASTIC_SPRING_DISCRETE_BEAM
  *MAT_ELASTIC_6DOF_SPRING_DISCRETE_BEAM
  *MAT_INELASTIC_6DOF_SPRING_DISCRETE_BEAM
  The latter two can be used as finite length beams with local coordinate systems.
- Moving SPC's are optional in that the constraints are applied in a local system that rotates with the 3 defining nodes.
- A moving local coordinate system, CID, can be used to determine orientation of discrete beam elements.
- Modal superposition analysis can be performed after an eigenvalue analysis. Stress recovery is based on type 18 shell and brick (SMP only).
- Rayleigh damping input factor is now input as a fraction of critical damping, i.e. 0.10. The old method required the frequency of interest and could be highly unstable for large input values.
- Airbag option "SIMPLE_PRESSURE_VOLUME" allows for the constant CN to be replaced by a load curve for initialization. Also, another load curve can be defined which allows CN to vary as a function of time during dynamic relaxation. After dynamic relaxation CN can be used as a fixed constant or load curve.
- Hybrid inflator model utilizing CHEMKIN and NIST databases is now available. Up to ten gases can be mixed.
- Option to track initial penetrations has been added in the automatic SMP contact types rather than moving the nodes back to the surface. This option has been available in the MPP contact for some time. This input can be defined on the fourth card of the *CONTROL_CONTACT input and on each contact definition on the third optional card in the *CONTACT definitions.
- If the average acceleration flag is active, the average acceleration for rigid body nodes is now written into the D3THDT and NODOUT files. In previous versions of LS-DYNA, the accelerations on rigid nodes were not averaged.
- A capability to initialize the thickness and plastic strain in the crash model is available through the option *INCLUDE_STAMPED_PART, which takes the results from the LS-DYNA stamping simulation and maps the thickness and strain distribution onto the same part with a different mesh pattern.
- A capability to include finite element data from other models is available through the option, *INCLUDE_TRANSFORM. This option will take the model defined in an INCLUDE file: offset all ID's; translate, rotate, and scale the coordinates; and transform the constitutive constants to another set of units.

Many new capabilities were added during 2001-2002 to create version 970 of LS-DYNA. Some of the new features, which are also listed below, were also added to later releases of version 960. Most new explicit capabilities work for both the MPP and SMP versions; however, the implicit capabilities for MPP require the development of a scalable eigenvalue solver and a parallel implementation of the constraint equations into the global matrices. This work is underway. A later release of version 970 is planned that will be scalable for implicit solutions.

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

*CONSTRAINED_SPOTWELD
*DATABASE_CROSS_SECTION
*ELEMENT_MASS

- The *DATABASE_ADAMS keyword is available to output a modal neutral file d3mnf. This is available upon customer request since it requires linking to an ADAMS library file.
- Penetration warnings for the contact option, "ignore initial penetration," are added as an option. Previously, no penetration warnings were written when this contact option was activated.
- Penetration warnings for nodes in-plane with shell mid-surface are printed for the AUTOMATIC contact options. Previously, these nodes were ignored since it was assumed that they belonged to a tied interface where an offset was not used; consequently, they should not be treated in contact.
- For the arbitrary spot weld option, the spot welded nodes and their contact segments are optionally written into the D3HSP file. See *CONTROL_CONTACT.
- For the arbitrary spot weld option, if a segment cannot be found for the spot welded node, an option now exists to error terminate. See *CONTROL_CONTACT.
- Spot weld resultant forces are written into the SWFORC file for solid elements used as spot welds.
- Solid materials have now been added to the failed element report and additional information is written for the "node is deleted" messages.
- A new option for terminating a calculation is available, *TERMINATION_CURVE.
- A 10-noded tetrahedron solid element is available with either a 4 or 5 point integration rule. This element can also be used for implicit solutions.
- A new 4 node linear shell element is available that is based on Wilson's plate element combined with a Pian-Sumihara membrane element. This is shell type 21.
- A shear panel element has been added for linear applications. This is shell type 22. This element can also be used for implicit solutions.
- A null beam element for visualization is available. The keyword to define this null beam is *ELEMENT_PLOTEL. This element is necessary for compatibility with NASTRAN.
- A scalar node can be defined for spring-mass systems. The keyword to define this node is *NODE_SCALAR. This node can have from 1 to 6 scalar degrees-of-freedom.
- A thermal shell has been added for through-thickness heat conduction. Internally, 8 additional nodes are created, four above and four below the mid-surface of the shell element. A quadratic temperature field is modeled through the shell thickness. Internally, the thermal shell is a 12 node solid element.
- A beam OFFSET option is available for the *ELEMENT_BEAM definition to permit the beam to be offset from its defining nodal points. This has the advantage that all beam formulations can now be used as shell stiffeners.
- A beam ORIENTATION option for orienting the beams by a vector instead of the third node is available in the *ELEMENT_BEAM definition for NASTRAN compatibility.
- Non-structural mass has been added to beam elements for modeling trim mass and for NASTRAN compatibility.

- An optional checking of shell elements to avoid abnormal terminations is available. See *CONTROL_SHELL. If this option is active, every shell is checked each time step to see if the distortion is so large that the element will invert, which will result in an abnormal termination. If a bad shell is detected, either the shell will be deleted or the calculation will terminate. The latter is controlled by the input.
- An offset option is added to the inertia definition. See *ELEMENT_INERTIA_ OFFSET keyword. This allows the inertia tensor to be offset from the nodal point.
- Plastic strain and thickness initialization is added to the draw bead contact option. See *CONTACT_DRAWBEAD_INITIALIZE.
- Tied contact with offsets based on both constraint equations and beam elements for solid elements and shell elements that have 3 and 6 degrees-of-freedom per node, respectively. See BEAM_OFFSET and CONSTRAINED_OFFSET contact options. These options will not cause problems for rigid body motions.
- The segment-based (SOFT=2) contact is implemented for MPP calculations. This enables airbags to be easily deployed on the MPP version.
- Improvements are made to segment-based contact for edge-to-edge and sliding conditions, and for contact conditions involving warped segments.
- An improved interior contact has been implemented to handle large shear deformations in the solid elements. A special interior contact algorithm is available for tetrahedron elements.
- Coupling with MADYMO 6.0 uses an extended coupling that allows users to link most MADYMO geometric entities with LS-DYNA FEM simulations. In this coupling MADYMO contact algorithms are used to calculate interface forces between the two models.
- Release flags for degrees-of-freedom for nodal points within nodal rigid bodies are available. This makes the nodal rigid body option nearly compatible with the RBE2 option in NASTRAN.
- Fast updates of rigid bodies for metalforming applications can now be accomplished by ignoring the rotational degrees-of-freedom in the rigid bodies that are typically inactive during sheet metal stamping simulations. See the keyword: *CONTROL_RIGID.
- Center of mass constraints can be imposed on nodal rigid bodies with the SPC option in either a local or a global coordinate system.
- Joint failure based on resultant forces and moments can now be used to simulate the failure of joints.
- CONSTRAINED_JOINT_STIFFNESS now has a TRANSLATIONAL option for the translational and cylindrical joints.
- Joint friction has been added using table look-up so that the frictional moment can now be a function of the resultant translational force.
- The nodal constraint options *CONSTRAINED_INTERPOLATION and *CONSTRAINED_LINEAR now have a local option to allow these constraints to be applied in a local coordinate system.
- Mesh coarsening can now be applied to automotive crash models at the beginning of an analysis to reduce computation times. See the new keyword: *CONTROL_COARSEN.
- Force versus time seatbelt pretensioner option has been added.

- Both static and dynamic coefficients of friction are available for seat belt slip rings. Previously, only one friction constant could be defined.
- *MAT_SPOTWELD now includes a new failure model with rate effects as well as additional failure options.
- Constitutive models added for the discrete beam elements:
  *MAT_1DOF_GENERALIZED_SPRING
  *MAT_GENERAL_NONLINEAR_6DOF_DISCRETE_BEAM
  *MAT_GENERAL_NONLINEAR_1DOF_DISCRETE_BEAM
  *MAT_GENERAL_SPRING_DISCRETE_BEAM
  *MAT_GENERAL_JOINT_DISCRETE_BEAM
  *MAT_SEISMIC_ISOLATOR
  for shell and solid elements:
  *MAT_PLASTICITY_WITH_DAMAGE_ORTHO
  *MAT_SIMPLIFIED_JOHNSON_COOK_ORTHOTROPIC_DAMAGE
  *MAT_HILL_3R
  *MAT_GURSON_RCDC
  for the solid elements:
  *MAT_SPOTWELD
  *MAT_HILL_FOAM
  *MAT_WOOD
  *MAT_VISCOELASTIC_HILL_FOAM
  *MAT_LOW_DENSITY_SYNTHETIC_FOAM
  *MAT_RATE_SENSITIVE_POLYMER
  *MAT_QUASILINEAR VISCOELASTIC
  *MAT_TRANSVERSELY_ANISOTROPIC_CRUSHABLE_FOAM
  *MAT_VACUUM
  *MAT_MODIFIED_CRUSHABLE_FOAM
  *MAT_PITZER_CRUSHABLE FOAM
  *MAT_JOINTED_ROCK
  *MAT_SIMPLIFIED_RUBBER
  *MAT_FHWA_SOIL
  *MAT_SCHWER_MURRAY_CAP_MODEL
- Failure time added to MAT_EROSION for solid elements.
- Damping in the material models *MAT_LOW_DENSITY_FOAM and *MAT_LOW_DENSITY_VISCOUS_FOAM can now be a tabulated function of the smallest stretch ratio.
- The material model *MAT_PLASTICITY_WITH_DAMAGE allows the table definitions for strain rate.
- Improvements in the option *INCLUDE_STAMPED_PART now allow all history data to be mapped to the crash part from the stamped part. Also, symmetry planes can be used to allow the use of a single stamping to initialize symmetric parts.
- Extensive improvements in trimming result in much better elements after the trimming is completed. Also, trimming can be defined in either a local or global coordinate system. This is a new option in *DEFINE_CURVE_TRIM.
- An option to move parts close before solving the contact problem is available, see *CONTACT_AUTO_MOVE.

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

2. PRELIMINARIES

Consider the body shown in Figure 2.1. We are interested in time-dependent deformation in which a point in b initially at X_α (α = 1, 2, 3) in a fixed rectangular Cartesian coordinate system moves to a point x_i (i = 1, 2, 3) in the same coordinate system. Since a Lagrangian formulation is considered, the deformation can be expressed in terms of the convected coordinates X_α, and time t

$$x_i = x_i\left(X_\alpha, t\right) \tag{2.1}$$

At time $t=0$, we have the initial conditions

$$x_i(X_\alpha, 0) = X_\alpha \tag{2.2a}$$

$$\dot{x}_i(X_\alpha, 0) = V_i\left(X_\alpha\right) \tag{2.2b}$$

where V_i defines the initial velocities.

2.1 Governing Equations

We seek a solution to the momentum equation

$$\sigma_{ij,j} + \rho\, f_i = \rho \ddot{x}_i \tag{2.3}$$

satisfying the traction boundary conditions

$$\sigma_{ij} n_i = t_i(t) \tag{2.4}$$

on boundary $\partial \mathrm{b}_1$, the displacement boundary conditions

$$x_i(X_\alpha, t) = D_i(t) \tag{2.5}$$

on boundary $\partial \mathrm{b}_2$, the contact discontinuity

$$\left(\sigma_{ij}^{+} - \sigma_{ij}^{-}\right)\ n_i = 0 \tag{2.6}$$

along an interior boundary $\partial \mathrm{b}_3$ when $x_i^{+} = x_i^{-}$. Here σ_{ij} is the Cauchy stress, ρ is the current density, f is the body force density, $\ddot{x}$ is acceleration, the comma denotes covariant differentiation, and n_j is a unit outward normal to a boundary element of ∂b.

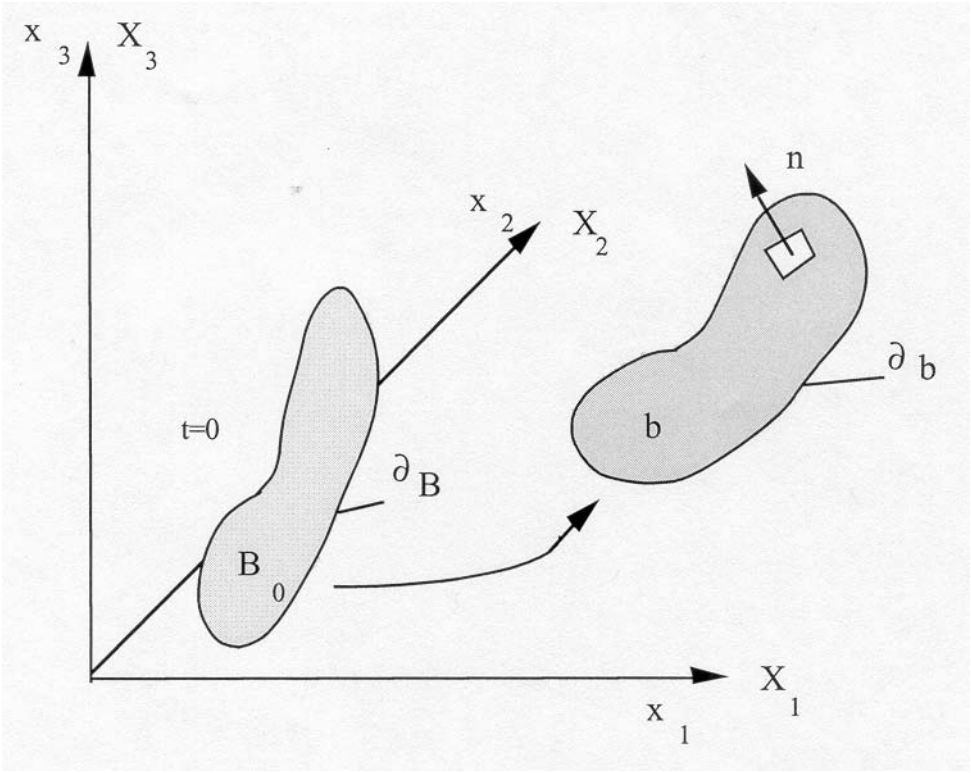


Figure 2.1. Notation.

Mass conservation is trivially stated

$$\rho V = \rho_0 \tag{2.7}$$

where V is the relative volume, i.e., the determinant of the deformation gradient matrix, F_{ij},

$$F_{ij} = \frac{\partial x_i}{\partial X_j} \tag{2.8}$$

and ρ_0 is the reference density. The energy equation

$$\dot{E} = V s_{ij} \dot{\varepsilon}_{ij} - (p+q)\dot{V} \tag{2.9}$$

is integrated in time and is used for equation of state evaluations and a global energy balance. In Equation (2.9), s_{ij} and p represent the deviatoric stresses and pressure,

$$s_{ij} = \sigma_{ij} + (p+q)\delta_{ij} \tag{2.10}$$

$$p = -\frac{1}{3}\sigma_{ij}\delta_{ij} - q = -\frac{1}{3}\sigma_{kk} - q \tag{2.11}$$

respectively, q is the bulk viscosity, δ_{ij} is the Kronecker delta ($\delta_{ij}=1$ if $i=j$; otherwise $\delta_{ij}=0$) and $\dot{\varepsilon}_{ij}$ is the strain rate tensor. The strain rates and bulk viscosity are discussed later.

We can write:

$$\begin{aligned}\int_v \left(\rho\ddot{x}_i - \sigma_{ij,j} - \rho f\right)\ \delta x_i dv + \int_{\partial b_1}\left(\sigma_{ij}n_j - t_i\right)\ \delta x_i ds \\ + \int_{\partial b_3}\left(\sigma_{ij}^{+} - \sigma_{ij}^{-}\right)\ n_j\delta x_i ds = 0\end{aligned} \tag{2.12}$$

where δx_i satisfies all boundary conditions on ∂b_2, and the integrations are over the current geometry. Application of the divergence theorem gives

$$\int_v \left(\sigma_{ij}\delta x_i\right),_j dv = \int_{\partial b_1}\sigma_{ij}n_j\delta x_i ds + \int_{\partial b_3}\left(\sigma_{ij}^{+} - \sigma_{ij}^{-}\right)n_j\delta x_i ds \tag{2.13}$$

and noting that

$$(\sigma_{ij}\delta x_i),_j\ \sigma_{ij,j}\delta x_i = \sigma_{ij}\delta x_{i,j} \tag{2.14}$$

leads to the weak form of the equilibrium equations:

$$\delta\pi = \int_v\ \rho\ddot{x}_i\,\delta x_i dv + \int_v \sigma_{ij}\delta x_{i,j}dv - \int_v \rho f_i\delta x_i dv - \int_{\partial b_1} t_i\delta x_i ds = 0 \tag{2.15}$$

a statement of the principle of virtual work.

We superimpose a mesh of finite elements interconnected at nodal points on a reference configuration and track particles through time, i.e.,

$$x_i\left(X_\alpha, t\right) = x_i\left(X_\alpha\left(\xi,\eta,\zeta\right), t\right) = \sum_{j=1}^{k}\phi_j\left(\xi,\eta,\zeta\right)x_i^j\left(t\right) \tag{2.16}$$

where ϕ_j are shape (interpolation) functions of the parametric coordinates (ξ,η,ζ), k is the number of nodal points defining the element, and x_i^j is the nodal coordinate of the jth node in the ith direction.

Summing over the n elements we may approximate $\delta\pi$ with

$$\delta\pi = \sum_{m=1}^{n}\delta\pi_m = 0 \tag{2.17}$$

and write

$$\sum_{m=1}^{n}\left\{\int_{v_m}\rho\ddot{x}_i\Phi_i^m dv+\int_{v_m}\sigma_{ij}^m\Phi_{i,j}^m dv-\int_{v_m}\rho f_i\Phi_i^m dv-\int_{\partial b_1}t_i\Phi_i^m ds\right\}=0 \tag{2.18}$$

where

$$\Phi_i^m=\left(\phi_1,\phi_2,...,\phi_k\right)_i^m \tag{2.19}$$

In matrix notation Equation (2.18) becomes

$$\sum_{m=1}^{n}\left\{\int_{v_m}\rho\,\mathrm{N}^t\mathrm{N}\,a\,dv+\int_{v_m}\mathrm{B}^t\sigma\,dv-\int_{v_m}\rho\,\mathrm{N}^t\mathrm{b}\,dv-\int_{\partial b_1}\mathrm{N}^t\mathrm{t}\,ds\right\}^m=0 \tag{2.20}$$

where N is an interpolation matrix, σ is the stress vector

$$\sigma^t=(\sigma_{xx},\sigma_{yy},\sigma_{zz},\sigma_{xy},\sigma_{yz},\sigma_{zx}) \tag{2.21}$$

B is the strain-displacement matrix, a is the nodal acceleration vector

$$\begin{bmatrix}\ddot{x}_1\\ \ddot{x}_2\\ \ddot{x}_3\end{bmatrix}=\mathrm{N}\begin{bmatrix}a_{x_1}\\ a_{y_1}\\ \vdots\\ a_{y_k}\\ a_{z_k}\end{bmatrix}=\mathrm{N}\,a \tag{2.22}$$

b is the body force load vector, and t are applied traction loads.

$$b=\begin{bmatrix}f_x\\ f_y\\ f_z\end{bmatrix},\quad t=\begin{bmatrix}t_x\\ t_y\\ t_z\end{bmatrix} \tag{2.23}$$

3. SOLID ELEMENTS

For a mesh of 8-node hexahedron solid elements, Equation (2.16) becomes:

$$x_i\left(X_\alpha,t\right)=x_i\left(X_\alpha\left(\xi,\eta,\zeta\right),t\right)=\sum_{j=1}^{8}\phi_j\left(\xi,\eta,\zeta\right)x_i^j\left(t\right) \tag{3.1}$$

The shape function ϕ_j is defined for the 8-node hexahedron as

$$\phi_j=\frac{1}{8}\left(1+\xi\xi_j\right)\ \left(1+\eta\eta_j\right)\ \left(1+\zeta\zeta_j\right) \tag{3.2}$$

where ξ_j, η_j, ζ_j take on their nodal values of (±1, ±1, ±1) and x_i^j is the nodal coordinate of the *j*th node in the *i*th direction (see Figure 3.1).

For a solid element, N is the 3 x 24 rectangular interpolation matrix given by

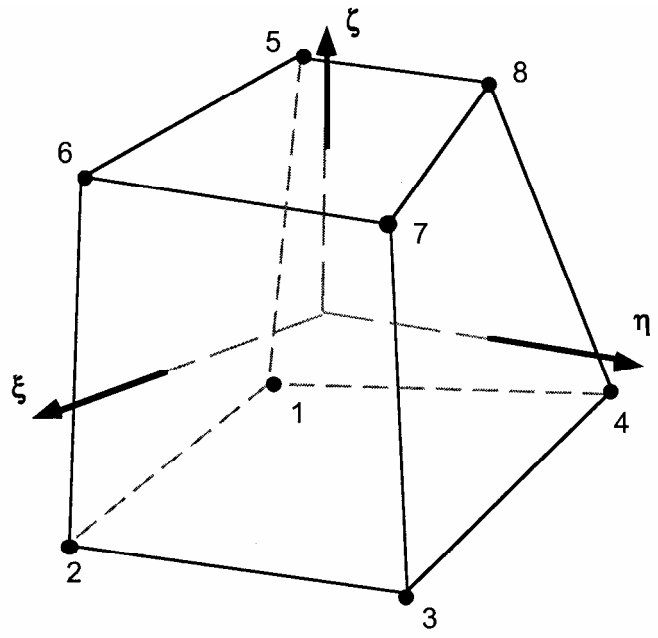
$$\mathrm{N}\left(\xi,\eta,\zeta\right)=\begin{bmatrix}\phi_1 & 0 & 0 & \phi_2 & 0 & \cdots & 0 & 0\\ 0 & \phi_1 & 0 & 0 & \phi_2 & \cdots & \phi_8 & 0\\ 0 & 0 & \phi_1 & 0 & 0 & \cdots & 0 & \phi_8\end{bmatrix} \tag{3.3}$$

σ is the stress vector

$$\sigma^t=\left(\sigma_{xx},\sigma_{yy},\sigma_{zz},\sigma_{xy},\sigma_{yz},\sigma_{zx}\right) \tag{3.4}$$

B is the 6 x 24 strain-displacement matrix

$$\mathrm{B}=\begin{bmatrix}\frac{\partial}{\partial x} & 0 & 0\\ 0 & \frac{\partial}{\partial y} & 0\\ 0 & 0 & \frac{\partial}{\partial z}\\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0\\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y}\\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x}\end{bmatrix}\mathrm{N} \tag{3.5}$$



Node	ξ	η	ζ
1	-1	-1	-1
2	1	-1	-1
3	1	1	-1
4	-1	1	-1
5	-1	-1	1
6	1	-1	1
7	1	1	1
8	-1	1	1

Figure 3.1. Eight-node solid hexahedron element.

In order to achieve a diagonal mass matrix the rows are summed giving the *k*th diagonal term as

$$m_{kk} = \int_v \rho \phi_k \sum_{i=1}^{8} \phi_i dv = \int_v \rho \phi_k dv \tag{3.6}$$

since the basis functions sum to unity.

Terms in the strain-displacement matrix are readily calculated. Note that

$$\begin{aligned}
\frac{\partial \phi_i}{\partial \xi} &= \frac{\partial \phi_i}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial \phi_i}{\partial y}\frac{\partial y}{\partial \xi} + \frac{\partial \phi_i}{\partial z}\frac{\partial z}{\partial \xi} \\
\frac{\partial \phi_i}{\partial \eta} &= \frac{\partial \phi_i}{\partial x}\frac{\partial x}{\partial \eta} + \frac{\partial \phi_i}{\partial y}\frac{\partial y}{\partial \eta} + \frac{\partial \phi_i}{\partial z}\frac{\partial z}{\partial \eta} \\
\frac{\partial \phi_i}{\partial \zeta} &= \frac{\partial \phi_i}{\partial x}\frac{\partial x}{\partial \zeta} + \frac{\partial \phi_i}{\partial y}\frac{\partial y}{\partial \zeta} + \frac{\partial \phi_i}{\partial z}\frac{\partial z}{\partial \zeta}
\end{aligned} \tag{3.7}$$

which can be rewritten as

$$\begin{bmatrix} \frac{\partial \phi_i}{\partial \xi} \\ \frac{\partial \phi_i}{\partial \eta} \\ \frac{\partial \phi_i}{\partial \zeta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \begin{bmatrix} \frac{\partial \phi_i}{\partial x} \\ \frac{\partial \phi_i}{\partial y} \\ \frac{\partial \phi_i}{\partial z} \end{bmatrix} = J \begin{bmatrix} \frac{\partial \phi_i}{\partial x} \\ \frac{\partial \phi_i}{\partial y} \\ \frac{\partial \phi_i}{\partial z} \end{bmatrix} \tag{3.8}$$

Inverting the Jacobian matrix, J, we can solve for the desired terms

$$\begin{bmatrix} \frac{\partial \phi_i}{\partial x} \\ \frac{\partial \phi_i}{\partial y} \\ \frac{\partial \phi_i}{\partial z} \end{bmatrix} = J^{-1} \begin{bmatrix} \frac{\partial \phi_i}{\partial \xi} \\ \frac{\partial \phi_i}{\partial \eta} \\ \frac{\partial \phi_i}{\partial \zeta} \end{bmatrix} \tag{3.9}$$

3.1 Volume Integration

Volume integration is carried out with Gaussian quadrature. If g is some function defined over the volume, and n is the number of integration points, then

$$\int_v g \, dv = \int_{-1}^{1} \int_{-1}^{1} \int_{-1}^{1} g \left| J \right| d\xi \, d\eta \, d\zeta \tag{3.10}$$

is approximated by

$$\sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} g_{jkl} \left| J_{jkl} \right| w_j w_k w_l \tag{3.11}$$

where w_j, w_k, w_l are the weighting factors,

$$g_{jkl} = g\left(\xi_j, \eta_k, \zeta_l\right) \tag{3.12}$$

and J is the determinant of the Jacobian matrix. For one-point quadrature

$$\begin{gathered} n = 1 \\ w_i = w_j = w_k = 2 \\ \xi_1 = \eta_1 = \zeta_1 = 0 \end{gathered} \tag{3.13}$$

and we can write

$$\int g dv = 8 g\left(0,0,0\right) \left| J\left(0,0,0\right) \right| \tag{3.14}$$

Note that $8\left|J\left(0,0,0\right)\right|$ approximates the element volume.

Perhaps the biggest advantage to one-point integration is a substantial savings in computer time. An anti-symmetry property of the strain matrix

$$\frac{\partial \phi_1}{\partial x_i} = -\frac{\partial \phi_7}{\partial x_i} \quad \frac{\partial \phi_3}{\partial x_i} = -\frac{\partial \phi_5}{\partial x_i}$$
$$\frac{\partial \phi_2}{\partial x_i} = -\frac{\partial \phi_8}{\partial x_i} \quad \frac{\partial \phi_4}{\partial x_i} = -\frac{\partial \phi_6}{\partial x_i} \tag{3.15}$$

at $\xi = \eta = \zeta = 0$ reduces the amount of effort required to compute this matrix by more than 25 times over an 8-point integration. This cost savings extends to strain and element nodal force calculations where the number of multiplies is reduced by a factor of 16. Because only one constitutive evaluation is needed, the time spent determining stresses is reduced by a factor of 8. Operation counts for the constant stress hexahedron are given in Table 3.2. Included are counts for the Flanagan and Belytschko [1981] hexahedron and the hexahedron used by Wilkins [1974] in his integral finite difference method, which was also implemented [Hallquist 1979].

It may be noted that 8-point integration has another disadvantage in addition to cost. Fully integrated elements used in the solution of plasticity problems and other problems where Poisson's ratio approaches .5 lock up in the constant volume bending modes. To preclude locking, an average pressure must be used over the elements; consequently, the zero energy modes are resisted by the deviatoric stresses. If the deviatoric stresses are insignificant relative to the pressure or, even worse, if material failure causes loss of this stress state component, hourglassing will still occur, but with no means of resisting it. Sometimes, however, the cost of the fully integrated element may be justified by increased reliability and if used sparingly may actually increase the overall speed.

3.2 Hourglass Control

The biggest disadvantage to one-point integration is the need to control the zero energy modes, which arise, called hourglassing modes. Undesirable hourglass modes tend to have periods that are typically much shorter than the periods of the structural response, and they are often observed to be oscillatory. However, hourglass modes that have periods that are comparable to the structural response periods may be a stable kinematic component of the global deformation modes and must be admissible. One way of resisting undesirable hourglassing is with a viscous damping or small elastic stiffness capable of stopping the formation of the anomalous modes but having a negligible affect on the stable global modes. Two of the early three-dimensional algorithms for controlling the hourglass modes were developed by Kosloff and Frazier [1974] and Wilkins et al. [1974].

Since the hourglass deformation modes are orthogonal to the strain calculations, work done by the hourglass resistance is neglected in the energy equation. This may lead to a slight loss of energy; however, hourglass control is always recommended for the under integrated solid elements. The energy dissipated by the hourglass forces reacting against the formations of the hourglass modes is tracked and reported in the output files MATSUM and GLSTAT.

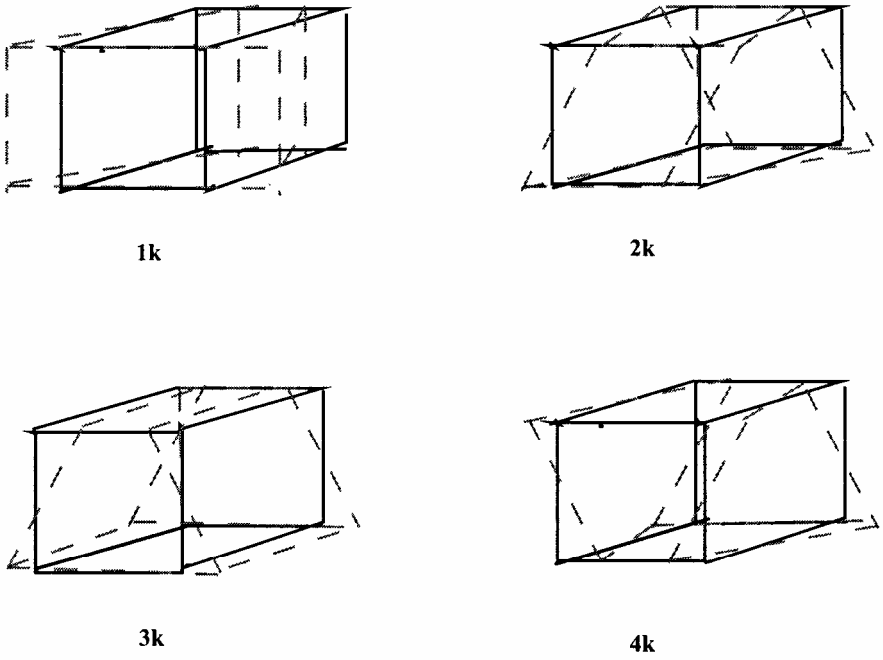


Figure 3.2. The hourglass modes of an eight-node element with one integration point are shown [Flanagan and Belytschko 1981]. A total of twelve modes exist.

It is easy to understand the reasons for the formation of the hourglass modes. Consider the following strain rate calculations for the 8-node solid element

$$\dot{\varepsilon}_{ij} = \frac{1}{2}\left(\sum_{k=1}^{8} \frac{\partial \phi_k}{\partial x_i} \dot{x}_j^k + \frac{\partial \phi_k}{\partial x_j} \dot{x}_i^k \right) \tag{3.16}$$

Whenever diagonally opposite nodes have identical velocities, i.e.,

$$\dot{x}_i^1 = \dot{x}_i^7 \ , \dot{x}_i^2 = \dot{x}_i^8 \ , \dot{x}_i^3 = \dot{x}_i^5 \ , \dot{x}_i^4 = \dot{x}_i^6 \tag{3.17}$$

the strain rates are identically zero:

$$\dot{\varepsilon}_{ij} = 0 \tag{3.18}$$

due to the asymmetries in Equations (3.15). It is easy to prove the orthogonality of the hourglass shape vectors, which are listed in Table 3.1 and shown in Figure 3.2 with the derivatives of the shape functions:

$$\sum_{k=1}^{8} \frac{\partial \phi}{\partial x_i} \Gamma_{\alpha k} = 0 \qquad i = 1,2,3 \qquad \alpha = 1,2,3,4 \tag{3.19}$$

The product of the base vectors with the nodal velocities

$$h_{i\alpha} = \sum_{k=1}^{8} \dot{x}_i^k \Gamma_{\alpha k} = 0 \tag{3.20}$$

are nonzero if hourglass modes are present. The 12 hourglass-resisting force vectors, $f_{i\alpha}^k$ are

$$f_{i\alpha}^k = a_h h_{i\alpha} \Gamma_{\alpha k} \tag{3.21}$$

where

$$a_h = Q_{hg} \rho v_e^{2/3} \frac{c}{4} \tag{3.22}$$

in which v_e is the element volume, c is the material sound speed, and Q_{hg} is a user-defined constant usually set to a value between .05 and .15. The hourglass resisting forces of Equation (3.21) are not orthogonal to rigid body rotations; however, the approach of Flanagan and Belytschko [1981] is orthogonal.

	$\alpha = 1$	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$
Γ_{j1}	1	1	1	1
Γ_{j2}	1	-1	-1	-1
Γ_{j3}	-1	-1	1	1
Γ_{j4}	-1	1	-1	-1
Γ_{j5}	-1	-1	1	-1
Γ_{j6}	-1	1	-1	1
Γ_{j7}	1	1	1	-1
Γ_{j8}	1	-1	-1	1

Table 3.1. Hourglass base vectors.

	DYNA3D	Flanagan-Belytschko [1981]	Wilkins FDM
Strain displacement matrix	94	357	843
Strain rates	87	156	
Force	117	195	270
Subtotal	298	708	1,113
Hourglass control	130	620	680
Total	428	1,328	1,793

Table 3.2. Operation counts for a constant stress hexahedron (includes adds, subtracts, multiplies, and divides in major subroutines, and is independent of vectorization). Material subroutines will add as little as 60 operations for the bilinear elastic-plastic routine to ten times as much for multi-surface plasticity and reactive flow models. Unvectorized material models will increase that share of the cost a factor of four or more.

Instead of resisting components of the bilinear velocity field that are orthogonal to the strain calculation, Flanagan and Belytschko resist components of the velocity field that are not part of a fully linear field. They call this field, defined below, the hourglass velocity field

$$\dot{x}_i^{k^{HG}} = \dot{x}_i - \dot{x}_i^{k^{LIN}} \tag{3.23}$$

where

$$\dot{x}_i^{k^{LIN}} = \dot{\bar{x}} + \dot{\bar{x}}_{i,j}\left(x_j^k - \bar{x}_j\right) \tag{3.24}$$

$$\bar{x}_i = \frac{1}{8}\sum_{k=1}^{8} x_i^k \qquad \dot{\bar{x}}_i = \frac{1}{8}\sum_{k=1}^{8} \dot{x}_i^k \tag{3.25}$$

Flanagan and Belytschko construct geometry-dependent hourglass shape vectors that are orthogonal to the fully linear velocity field and the rigid body field. With these vectors they resist the hourglass velocity deformations. Defining hourglass shape vectors in terms of the base vectors as

$$\gamma_{\alpha k} = \Gamma_{\alpha k} - \phi_{k,i}\sum_{n=1}^{8} x_i^n \Gamma_{\alpha n} \tag{3.26}$$

and setting

$$g_{i\alpha} = \sum_{k=1}^{8} \dot{x}_i^k \gamma_{\alpha k} = 0\,,$$

the 12 resisting force vectors become

$$f_{i\alpha}^{k} = a_h g_{i\alpha} \gamma_{\alpha k} \tag{3.27}$$

where a_h is a constant given in Equation (3.21).

The hourglass forces given by Equations (3.21) and (3.27) are identical if the, hexahedron element is a parallelepiped. The default hourglass control method for solid element is given by Equation (3.21); however, we recommend the Flanagan-Belytschko approach for problems that have large rigid body rotations since the default approach is not orthogonal to rigid body rotations.

A cost comparison in Table 3.2 shows that the default hourglass viscosity requires approximately 130 adds or multiplies per hexahedron, compared to 620 and 680 for the algorithms of Flanagan-Belytschko and Wilkins.

The type 6 hourglass stabilization for the 3D hexahedral element is available for both implicit and explicit solutions. Based on material properties and element geometry, this stiffness type stabilization is developed by an assumed strain method [Belytschko and Bindeman 1993] such that the element does not lock with nearly incompressible material. When the user defined hourglass constant Q_{hg} is set to 1.0, accurate coarse mesh bending stiffness is obtained for elastic material. For nonlinear material, a smaller value of Q_{hg} is suggested and the default value is set to 0.1. In the implicit form, the assumed strain stabilization matrix is:

$$K^{stab} = 2\mu Q_{hg} \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \tag{3.28}$$

where the 8 x 8 submatricies are calculated by:

$$k_{ii} \equiv H_{ii} \left[\left(\frac{1}{1-\upsilon} \right) \left(\gamma_j \gamma_j^t + \gamma_k \gamma_k^t \right) + \left(\frac{1+\upsilon}{3} \right) \gamma_4 \gamma_4^t \right] + \frac{1}{2} \left(H_{jj} + H_{kk} \right) \gamma_i \gamma_i^t \tag{3.29}$$

$$k_{ij} \equiv H_{ij} \left[\left(\frac{\upsilon}{1-\upsilon} \right) \gamma_j \gamma_i^t + \frac{1}{2} \gamma_i \gamma_j^t \right] \tag{3.30}$$

$$H_{ii} \equiv \int_v \left(h_{j,i} \right)^2 dv = \int_v \left(h_{k,i} \right)^2 dv = 3 \int_v \left(h_{4,i} \right)^2 dv \tag{3.68}$$

$$H_{ij} \equiv \int_v h_{i,j} h_{j,i} dv \tag{3.31}$$

$$h_1 = \xi\eta \quad h_2 = \eta\zeta \quad h_3 = \zeta\xi \quad h_4 = \xi\eta\zeta \tag{3.32}$$

Subscripts $i, j, and\ k$ are permuted as in Table 3.3. A comma indicates a derivative with respect to the spatial variable that follows. The hourglass vectors, γ_α are defined by equation (3.26). The stiffness matrix is evaluated in a corotational coordinate system that is aligned with the referential axis of the element. The use of a corotational system allows direct evaluation of integrals in equations (3.30) and (3.31) by simplified equations that produce a more accurate element than full integration.

$$H_{ii} = \frac{1}{3}\frac{\left(\Lambda_j^t x_j\right)\left(\Lambda_k^t x_k\right)}{\left(\Lambda_i^t x_i\right)} \tag{3.33a}$$

$$H_{ij} = \frac{1}{3}\left(\Lambda_k^t x_k\right) \tag{3.33b}$$

Λ_i are 8 x 1 matrices of the referential coordinates of the nodes as given in Figure 3.1, and x_i are 8x1 matrices of the nodal coordinates in the corotational system. For each material type, a Poisson's ratio, ν, and an effective shear modulus, μ, is needed.

i	j	k
1	2	3
1	3	2
2	3	1
2	1	3
3	1	2
3	2	1

Table 3.3. Permutations of *i*, *j*, and *k*.

In the explicit form, the 12 hourglass force stabilization vectors are

$$f_i^{stab} = \sum_{\alpha=1}^{4} Q_{hg} Q_{i\alpha} \gamma_\alpha^t \tag{3.34}$$

where the 12 generalized stresses are calculated incrementally by

$$Q_{i\alpha}^n = Q_{i\alpha}^{n-1} + \Delta t \dot{Q}_{i\alpha}^{n-\frac{1}{2}} \tag{3.35}$$

and

$$\dot{Q}_{ii} = \mu[(H_{jj} + H_{kk})\dot{q}_{ii} + H_{ij}\dot{q}_{jj} + H_{ik}\dot{q}_{kk}] \tag{3.36}$$

$$\dot{Q}_{ij} = 2\mu[\frac{1}{1-\upsilon} H_{ii}\dot{q}_{ij} + \upsilon H_{kk}\dot{q}_{ii}] \tag{3.37}$$

$$\dot{Q}_{i4} = 2\mu\left(\frac{1+\nu}{3}\right)H_{ii}\dot{q}_{i4} \tag{3.38}$$

$$\dot{q}_{i\alpha} = \gamma_\alpha^t \dot{x}_i \tag{3.39}$$

Subscripts $i, j, and\ k$ are permuted as per Table 3.3. As with the implicit form, calculations are done in a corotational coordinate system in order to use the simplified equations (3.33a) and (3.33b).

3.3 Puso Hourglass Control

Regarding the solid elements in LS-DYNA, the fully integrated brick uses selective-reduced integration, which is known to alleviate volumetric locking but not shear locking for elements with poor aspect ratio. The enhanced assumed strain methods have been the most successful at providing coarse mesh accuracy for general non-linear material models. In short, these elements tend to sacrifice computational efficiency for accuracy and are hence of little interest in explicit analysis. Puso [2000] developed an enhanced assumed strain element that combines coarse mesh accuracy with computational efficiency. It is formulated as a single point integrated brick with an enhanced assumed strain physical stabilization. In this project, we have implemented this element in LS-DYNA and made comparisons with the assumed strain element developed by Belytschko and Bindeman [1993] to see whether it brings anything new to the existing LS-DYNA element library.

The element formulation is that of Puso [2000], and is essentially the mean strain hexahedral element by Flanagan and Belytschko [1981] in which the perturbation hourglass control is substituted for an enhanced assumed strain stabilization force.

Given the matrices

$$\mathbf{s} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \qquad \mathbf{\Xi} = \begin{bmatrix} -1 & -1 & -1 \\ 1 & -1 & -1 \\ 1 & 1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & 1 \\ -1 & 1 & 1 \end{bmatrix} \qquad \mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 \\ -1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & -1 \end{bmatrix} \tag{3.40}$$

we can define the vector of shape functions as

$$\mathbf{N}(\xi) = \frac{1}{8}(\mathbf{s} + \mathbf{\Xi}\xi + \mathbf{H}\mathbf{h}(\xi)) \tag{3.41}$$

where

$$\boldsymbol{\xi} = \begin{bmatrix} \xi \\ \eta \\ \varsigma \end{bmatrix} \qquad \mathbf{h}(\boldsymbol{\xi}) = \begin{bmatrix} \eta\varsigma \\ \xi\varsigma \\ \xi\eta \\ \xi\eta\varsigma \end{bmatrix}. \tag{3.42}$$

The position vector

$$\mathbf{x}(\boldsymbol{\xi}) = \begin{bmatrix} x(\boldsymbol{\xi}) \\ y(\boldsymbol{\xi}) \\ z(\boldsymbol{\xi}) \end{bmatrix} \tag{3.43}$$

is for isoparametric finite elements given as

$$\mathbf{x}(\boldsymbol{\xi}) = \mathbf{X}^T \mathbf{N}(\boldsymbol{\xi}), \tag{3.44}$$

where

$$\mathbf{X} = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ x_4 & y_4 & z_4 \\ x_5 & y_5 & z_5 \\ x_6 & y_6 & z_6 \\ x_7 & y_7 & z_7 \\ x_8 & y_8 & z_8 \end{bmatrix} \tag{3.45}$$

is the matrix of nodal coordinates. The Jacobian matrix maps the isoparametric domain to the physical domain as

$$\mathbf{J}(\boldsymbol{\xi}) = \frac{\partial \mathbf{x}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \tag{3.46}$$

and we find the Jabobian matrix at the element centroid to be

$$\mathbf{J}_0 = \mathbf{J}(\mathbf{0}) = \frac{1}{8}\mathbf{X}^T \boldsymbol{\Xi}. \tag{3.47}$$

We may use this to rewrite the vector of shape functions partially in terms of the position vector as

$$\mathbf{N}(\xi) = \mathbf{b}_0 + \mathbf{B}_0\mathbf{x} + \mathbf{\Gamma h}(\xi) \tag{3.48}$$

where

$$\begin{aligned}
\mathbf{b}_0 &= \frac{1}{8}\left\{\mathbf{I} - \mathbf{B}_0\mathbf{X}^T\right\}\mathbf{s} \\
\mathbf{\Gamma} &= \frac{1}{8}\left\{\mathbf{I} - \mathbf{B}_0\mathbf{X}^T\right\}\mathbf{H} \\
\mathbf{B}_0 &= \frac{1}{8}\mathbf{\Xi J}_0^{-1}
\end{aligned} \tag{3.49}$$

The gradient-displacement matrix from this expression is given as

$$\mathbf{B}(\xi) = \mathbf{B}_0 + \mathbf{B}_s(\xi) \tag{3.50}$$

where

$$\mathbf{B}_s(\xi) = \mathbf{\Gamma}\frac{\partial \mathbf{h}(\xi)}{\partial \xi}\mathbf{J}(\xi)^{-1}. \tag{3.51}$$

We have

$$\frac{\partial \mathbf{h}(\xi)}{\partial \xi} = \begin{bmatrix} 0 & \varsigma & \eta \\ \varsigma & 0 & \xi \\ \eta & \xi & 0 \\ \eta\varsigma & \xi\varsigma & \xi\eta \end{bmatrix}. \tag{3.52}$$

At this point we substitute the gradient-displacement matrix at the centroid of the element $\mathbf{B}_0$ with the mean gradient-displacement matrix $\overline{\mathbf{B}}$ defined as

$$\overline{\mathbf{B}} = \frac{1}{V_e}\int_e \mathbf{B}(\xi)dV_e\ , \tag{3.53}$$

where e refers to the element domain and V_e is the volume of the element, in all of the expressions above. That is

$$\overline{\mathbf{\Gamma}} = \frac{1}{8}\left\{\mathbf{I} - \overline{\mathbf{B}}\mathbf{X}^T\right\}\mathbf{H} \tag{3.54}$$

and

$$\mathbf{B}(\xi) = \overline{\mathbf{B}} + \overline{\mathbf{B}}_s(\xi) \tag{3.55}$$

where

$$\overline{\mathbf{B}}_s(\xi) = \overline{\mathbf{\Gamma}} \frac{\partial \mathbf{h}(\xi)}{\partial \xi} \mathbf{J}(\xi)^{-1}. \tag{3.56}$$

Proceeding, we write the expression for the rate-of-deformation as

$$\begin{aligned}
\dot{\boldsymbol{\varepsilon}} &= \frac{1}{2}(\dot{\mathbf{X}}^T \mathbf{B}(\xi) + \mathbf{B}(\xi)^T \dot{\mathbf{X}}) = \frac{1}{2}(\dot{\mathbf{X}}^T \overline{\mathbf{B}} + \overline{\mathbf{B}}^T \dot{\mathbf{X}}) + \frac{1}{2}(\dot{\mathbf{X}}^T \overline{\mathbf{B}}_s(\xi) + \overline{\mathbf{B}}_s(\xi)^T \dot{\mathbf{X}}) \\
&= \frac{1}{2}(\dot{\mathbf{X}}^T \overline{\mathbf{B}} + \overline{\mathbf{B}}^T \dot{\mathbf{X}}) + \frac{1}{2}(\dot{\mathbf{X}}^T \overline{\mathbf{\Gamma}} \frac{\partial \mathbf{h}(\xi)}{\partial \xi} \mathbf{J}(\xi)^{-1} + \mathbf{J}(\xi)^{-T} \left(\frac{\partial \mathbf{h}(\xi)}{\partial \xi} \right)^T \overline{\mathbf{\Gamma}}^T \dot{\mathbf{X}}) \\
&= \frac{1}{2}(\dot{\mathbf{X}}^T \overline{\mathbf{B}} + \overline{\mathbf{B}}^T \dot{\mathbf{X}}) + \frac{1}{2} \mathbf{J}(\xi)^{-T} (\mathbf{J}(\xi)^T \dot{\mathbf{X}}^T \overline{\mathbf{\Gamma}} \frac{\partial \mathbf{h}(\xi)}{\partial \xi} + \left(\frac{\partial \mathbf{h}(\xi)}{\partial \xi} \right)^T \overline{\mathbf{\Gamma}}^T \dot{\mathbf{X}} \mathbf{J}(\xi)) \mathbf{J}(\xi)^{-1}
\end{aligned} \tag{3.57}$$

where we substitute the occurrences of the jacobian matrix $\mathbf{J}(\xi)$ with the following expressions

$$\dot{\boldsymbol{\varepsilon}} \approx \frac{1}{2}(\dot{\mathbf{X}}^T \overline{\mathbf{B}} + \overline{\mathbf{B}}^T \dot{\mathbf{X}}) + \frac{1}{2} \hat{\mathbf{J}}_0^{-T} (\mathbf{J}_0^T \dot{\mathbf{X}}^T \overline{\mathbf{\Gamma}} \frac{\partial \mathbf{h}(\xi)}{\partial \xi} + \left(\frac{\partial \mathbf{h}(\xi)}{\partial \xi} \right)^T \overline{\mathbf{\Gamma}}^T \dot{\mathbf{X}} \mathbf{J}_0) \hat{\mathbf{J}}_0^{-1} \tag{3.58}$$

where

$$\hat{\mathbf{J}}_0 = \begin{bmatrix} \|\mathbf{j}_1\| & & \\ & \|\mathbf{j}_2\| & \\ & & \|\mathbf{j}_3\| \end{bmatrix} \tag{3.59}$$

and $\mathbf{j}_i$ is the i:th column in the matrix $\mathbf{J}_0$. This last approximation is the key to the mesh distortion insensitivity that characterizes the element.

Changing to Voigt notation, we define the stabilization portion of the strain rate as

$$\dot{\boldsymbol{\varepsilon}}_s = \hat{\mathbf{J}}_0^{-1} \tilde{\mathbf{B}}_s(\xi) \tilde{\dot{\mathbf{u}}} \tag{3.60}$$

where now

$$\hat{\mathbf{J}}_0^{-1} = \begin{bmatrix} \|\mathbf{j}_1\|^{-2} & & & & & & & \\ & \|\mathbf{j}_2\|^{-2} & & & & & & \\ & & \|\mathbf{j}_3\|^{-2} & & & & & \\ & & & \|\mathbf{j}_1\|^{-1}\|\mathbf{j}_2\|^{-1} & & & & \\ & & & & \|\mathbf{j}_3\|^{-1}\|\mathbf{j}_2\|^{-1} & & & \\ & & & & & \|\mathbf{j}_1\|^{-1}\|\mathbf{j}_3\|^{-1} & & \end{bmatrix} \tag{3.61}$$

$$\tilde{\mathbf{B}}_s(\xi) = \begin{bmatrix} \bar{\gamma}_2\varsigma + \bar{\gamma}_3\eta + \bar{\gamma}_4\varsigma\eta & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \bar{\gamma}_1\varsigma + \bar{\gamma}_3\xi + \bar{\gamma}_4\varsigma\xi & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \bar{\gamma}_1\eta + \bar{\gamma}_2\xi + \bar{\gamma}_4\xi\eta \\ \bar{\gamma}_1\varsigma & \bar{\gamma}_2\varsigma & \mathbf{0} \\ \mathbf{0} & \bar{\gamma}_2\xi & \bar{\gamma}_3\xi \\ \bar{\gamma}_1\eta & \mathbf{0} & \bar{\gamma}_3\eta \end{bmatrix} \tag{3.62}$$

and $\tilde{\mathbf{u}}$ is the vector of nodal velocities transformed to the isoparametric system according to

$$\tilde{\dot{\mathbf{u}}} = \Im\dot{\mathbf{u}} \tag{3.63}$$

where $\Im$ is the 24 by 24 matrix that transforms the 8 nodal velocity vectors to the isoparametric domain given by

$$\Im = \text{perm}\begin{bmatrix} \mathbf{J}_0^T & & \\ & \ddots & \\ & & \mathbf{J}_0^T \end{bmatrix}. \tag{3.64}$$

Moreover, $\bar{\gamma}_i$ is the i:th row of $\bar{\mathbf{\Gamma}}^T$. We have deliberately neglected terms that cause parallelepiped finite elements to lock in shear.

To eliminate Poisson type locking in bending and volumetric locking, an enhanced isoparametric rate-of-strain field is introduced as

$$\dot{\boldsymbol{\varepsilon}}_e = \hat{\mathbf{J}}_0^{-1}\tilde{\mathbf{G}}(\boldsymbol{\xi})\dot{\boldsymbol{\alpha}} \tag{3.65}$$

with

$$\tilde{\mathbf{G}}(\xi) = \begin{bmatrix} \xi & 0 & 0 & \xi\eta & \eta\varsigma & \varsigma\xi \\ 0 & \eta & 0 & \xi\eta & \eta\varsigma & \varsigma\xi \\ 0 & 0 & \varsigma & \xi\eta & \eta\varsigma & \varsigma\xi \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.66}$$

Hence, the stabilized strain field becomes

$$\dot{\boldsymbol{\varepsilon}}_s = \hat{\mathbf{J}}_0^{-1}(\tilde{\mathbf{B}}_s(\xi)\dot{\tilde{\mathbf{u}}} + \tilde{\mathbf{G}}(\xi)\dot{\boldsymbol{\alpha}}) = \hat{\mathbf{J}}_0^{-1}\tilde{\dot{\boldsymbol{\varepsilon}}}_s \tag{3.67}$$

where $\dot{\boldsymbol{\alpha}}$ is the enhanced strain vector that must be determined from an equilibrium condition.

The virtual work equation can be written

$$\begin{aligned} \delta W_{\text{int}} &= \int_e \delta\boldsymbol{\varepsilon}^T \boldsymbol{\sigma} dV_e \\ &= \int_e \delta\boldsymbol{\varepsilon}^T J^{-1}\mathbf{J}^T \mathbf{S} dV_e \\ &= \int_e \delta\boldsymbol{\varepsilon}^T J^{-1}\mathbf{J}^T \int_0^t \mathbf{C}^{SE}\dot{\mathbf{E}} d\tau dV_e \\ &= \int_e \delta\boldsymbol{\varepsilon}^T J^{-1}\mathbf{J}^T \int_0^t J\mathbf{J}^{-T}\mathbf{C}^{\sigma}\dot{\boldsymbol{\varepsilon}} d\tau dV_e \\ &= \int_p \delta\boldsymbol{\varepsilon}^T \bar{j}_0 \mathbf{J}_0^T \int_0^t J\mathbf{J}_0^{-T}\mathbf{C}^{\sigma}\dot{\boldsymbol{\varepsilon}} d\tau dV_p \\ &= \int_p \delta\boldsymbol{\varepsilon}^T \mathbf{J}_0^T \int_0^t j_0\mathbf{J}_0^{-T}\mathbf{C}^{\sigma}\mathbf{J}_0^{-1}\mathbf{J}_0\dot{\boldsymbol{\varepsilon}} d\tau dV_p \\ &\approx \int_p \delta\tilde{\boldsymbol{\varepsilon}}^T \int_0^t \frac{V_e}{8}\hat{\mathbf{J}}_0^{-T}\mathbf{C}^{\sigma}\hat{\mathbf{J}}_0^{-1}\tilde{\dot{\boldsymbol{\varepsilon}}} d\tau dV_p \end{aligned} \tag{3.68}$$

where J is the determinant of the deformation gradient, $\mathbf{J}$ is the push-forward operator of a symmetric 2nd order tensor, j_0 is the determinant of the jacobian matrix, $\bar{j}_0$ is the determinant of the jacobian matrix at time 0, $\boldsymbol{\sigma}$ is the true stress tensor, $\mathbf{S}$ is the 2nd Piola-Kirchhoff stress tensor, $\mathbf{C}^{SE}$ is the material tangent modulus, $\mathbf{C}^{\sigma}$ is the spatial tangent modulus and V_e is the volume of the element. In the above, we have used various transformation formulae between different stress and constitutive tensors. At this point we are only interested in how to handle the stabilization portion of the strain rate field, the constant part is only used to update the midpoint

stress as usual. Because of orthogonality properties of the involved matrices, it turns out that we may just insert the expression for the stabilization strain rate field to get

$$\begin{aligned}\delta W_{\text{int}}^{s} &\approx \int_p \delta\tilde{\boldsymbol{\varepsilon}}_s^T \int_0^t \frac{V_e}{8} \hat{\mathbf{J}}_0^{-T} \mathbf{C}^{\sigma} \hat{\mathbf{J}}_0^{-1} \tilde{\dot{\boldsymbol{\varepsilon}}}_s d\tau dV_p \\ &= \begin{bmatrix} \delta\tilde{\mathbf{u}}^T & \delta\boldsymbol{\alpha}^T \end{bmatrix} \int_p \begin{bmatrix} \tilde{\mathbf{B}}_s(\xi)^T \\ \tilde{\mathbf{G}}(\xi)^T \end{bmatrix} \int_0^t \frac{V_e}{8} \hat{\mathbf{J}}_0^{-T} \mathbf{C}^{\sigma} \hat{\mathbf{J}}_0^{-1} \begin{bmatrix} \tilde{\mathbf{B}}_s(\xi) & \tilde{\mathbf{G}}(\xi) \end{bmatrix} \begin{bmatrix} \dot{\tilde{\mathbf{u}}} \\ \dot{\boldsymbol{\alpha}} \end{bmatrix} d\tau dV_p \end{aligned} \tag{3.69}$$

The stabilization contribution to the internal force vector is given by

$$\begin{bmatrix} \mathbf{f}_u \\ \mathbf{f}_\alpha \end{bmatrix} = \begin{bmatrix} \mathfrak{I}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \int_p \begin{bmatrix} \tilde{\mathbf{B}}_s(\xi)^T \\ \tilde{\mathbf{G}}(\xi)^T \end{bmatrix} \int_0^t \frac{V_e}{8} \hat{\mathbf{J}}_0^{-T} \mathbf{C}^{\sigma} \hat{\mathbf{J}}_0^{-1} \begin{bmatrix} \tilde{\mathbf{B}}_s(\xi) & \tilde{\mathbf{G}}(\xi) \end{bmatrix} \begin{bmatrix} \mathfrak{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{u}} \\ \dot{\boldsymbol{\alpha}} \end{bmatrix} d\tau dV_p \,. \tag{3.70}$$

In a discretization, the condition $\mathbf{f}_\alpha = \mathbf{0}$ is used to determine $\Delta\boldsymbol{\alpha}$, the increment of the enhanced strain variables, from $\Delta\mathbf{u}$, the increment in displacements. This is inserted back into the expression for the internal force vectors to determine $\mathbf{f}_u$, the stabilization contribution to the internal force vector.

The implementation of the element is very similar to the implementation of the one point integrated mean strain hexahedral by Flanagan and Belytschko [1981]. The hourglass forces are calculated in a different manner.

From the midpoint stress update we get a bulk and shear modulus characterizing the material at this specific point in time. From this we form the isotropic spatial tangent modulus $\mathbf{C}^{\sigma}$ to be used for computing the stabilization force from Equation (3.70).

3.4 Fully Integrated Brick Elements and Mid-Step Strain Evaluation

To avoid locking in the fully integrated brick elements strain increments at a point in a constant pressure, solid element are defined by [see Nagtegaal, Parks, and Rice 1974]

$$\begin{aligned} \Delta\varepsilon_{xx} &= \frac{\partial \Delta u}{\partial x^{n+\frac{1}{2}}} + \phi & \Delta\varepsilon_{xy} &= \frac{\dfrac{\partial \Delta v}{\partial x^{n+\frac{1}{2}}} + \dfrac{\partial \Delta u}{\partial y^{n+\frac{1}{2}}}}{2} \\ \Delta\varepsilon_{yy} &= \frac{\partial \Delta v}{\partial y^{n+\frac{1}{2}}} + \phi & \Delta\varepsilon_{yz} &= \frac{\dfrac{\partial \Delta w}{\partial y^{n+\frac{1}{2}}} + \dfrac{\partial \Delta v}{\partial z^{n+\frac{1}{2}}}}{2} \\ \Delta\varepsilon_{zz} &= \frac{\partial \Delta w}{\partial z^{n+\frac{1}{2}}} + \phi & \Delta\varepsilon_{zx} &= \frac{\dfrac{\partial \Delta u}{\partial z^{n+\frac{1}{2}}} + \dfrac{\partial \Delta w}{\partial x^{n+\frac{1}{2}}}}{2} \end{aligned} \tag{3.71}$$

where ϕ modifies the normal strains to ensure that the total volumetric strain increment at each integration point is identical

$$\phi = \Delta\varepsilon_v^{n+\frac{1}{2}} - \frac{\dfrac{\partial \Delta u}{\partial x^{n+\frac{1}{2}}} + \dfrac{\partial \Delta v}{\partial y^{n+\frac{1}{2}}} + \dfrac{\partial \Delta w}{\partial z^{n+\frac{1}{2}}}}{3}$$

and $\Delta\varepsilon_v^{n+\frac{1}{2}}$ is the average volumetric strain increment in the midpoint geometry

$$\frac{\dfrac{1}{3}\displaystyle\int_{v^{n+\frac{1}{2}}} \left(\frac{\partial \Delta u}{\partial x^{n+\frac{1}{2}}} + \frac{\partial \Delta v}{\partial y^{n+\frac{1}{2}}} + \frac{\partial \Delta w}{\partial z^{n+\frac{1}{2}}} \right) dv^{n+\frac{1}{2}}}{\displaystyle\int_{v^{n+\frac{1}{2}}} dv^{n+\frac{1}{2}}}, \tag{3.72}$$

Δu, Δv, and Δw are displacement increments in the x, y, and z directions, respectively, and

$$x^{n+\frac{1}{2}} = \frac{\left(x^n + x^{n+1}\right)}{2}, \tag{3.73a}$$

$$y^{n+\frac{1}{2}} = \frac{\left(y^n + y^{n+1}\right)}{2}, \tag{3.73b}$$

$$z^{n+\frac{1}{2}} = \frac{\left(z^n + z^{n+1}\right)}{2}, \tag{3.73c}$$

To satisfy the condition that rigid body rotations cause zero straining, it is necessary to use the geometry at the mid-step in the evaluation of the strain increments. As the default, LS-DYNA currently uses the geometry at step $n+1$ to save operations; however, for implicit calculations the mid-step strain calculation is always recommended, and, for explicit calculations, which involve rotating parts, the mid-step geometry should be used especially if the number of revolutions is large. The mid-step geometry can be activated either globally or for a subset of parts in the model by using the options on the control card, *CONTROL_ACCURACY.

Since the bulk modulus is constant in the plastic and viscoelastic material models, constant pressure solid elements result. In the thermoelastoplastic material, a constant temperature is assumed over the element. In the soil and crushable foam material, an average relative volume is computed for the element at time step $n+1$, and the pressure and bulk modulus associated with this relative volume is used at each integration point. For equations of state, one pressure evaluation is done per element and is added to the deviatoric stress tensor at each integration point.

The foregoing procedure requires that the strain-displacement matrix corresponding to Equations (3.40) and consistent with a constant volumetric strain, $\bar{B}$, be used in the nodal force calculations [Hughes 1980]. It is easy to show that:

$$F = \int_{v^{n+1}} \bar{B}^{n+1^t} \sigma^{n+1} dv^{n+1} = \int_{v^{n+1}} B^{n+1^t} \sigma^{n+1} dv^{n+1} \tag{3.74}$$

and avoid the needless complexities of computing $\bar{B}$.

3.5 Four Node Tetrahedron Element

The four node tetrahedron element with one point integration, shown in Figure 3.3, is a simple, fast, solid element that has proven to be very useful in modeling low density foams that have high compressibility. For most applications, however, this element is too stiff to give reliable results and is primarily used for transitions in meshes. The formulation follows the formulation for the one point solid element with the difference that there are no kinematic modes, so hourglass control is not needed. The basis functions are given by:

$$N_1(r,s,t) = r$$
$$N_2(r,s,t) = s$$
$$N_3(r,s,t) = 1 - r - s - t$$
$$N_4(r,s,t) = t$$

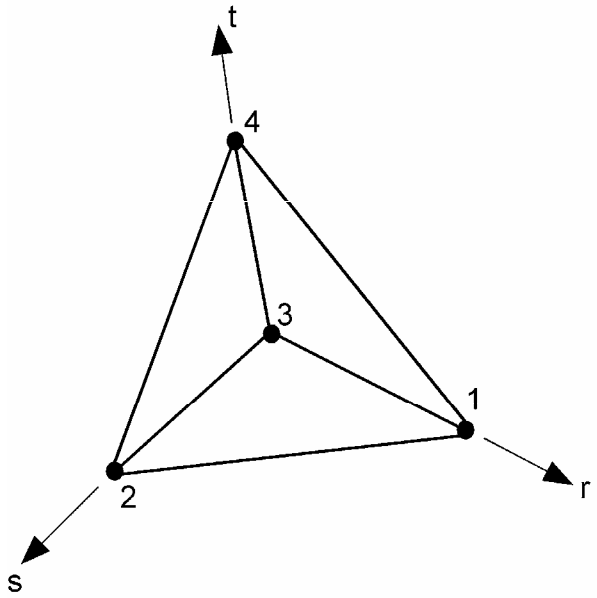


Figure 3.3. Four node tetrahedron.

If a tetrahedron element is needed, this element should be used instead of the collapsed solid element since it is, in general, considerably more stable in addition to being much faster.

Automatic sorting can be used, see *CONTROL_SOLID keyword, to segregate these elements in a mesh of 8 node solids for treatment as tetrahedrons.

3.6 Six Node Pentahedron Element

The pentahedron element with two point Gauss integration along its length, shown in Figure 3.4, is a solid element that has proven to be very useful in modeling axisymmetric structures where wedge shaped elements are used along the axis-of-revolution. The formulation follows the formulation for the one point solid element with the difference that, like the tetrahedron element, there are no kinematic modes, so hourglass control is not needed. The basis functions are given by:

$$N_1(r,s,t) = \frac{1}{2}(1-t)r$$

$$N_2(r,s,t) = \frac{1}{2}(1-t)(1-r-s)$$

$$N_3(r,s,t) = \frac{1}{2}(1+t)(1-r-s)$$

$$N_4(r,s,t) = \frac{1}{2}(1+t)r$$

$$N_5(r,s,t) = \frac{1}{2}(1-t)s$$

$$N_6(r,s,t) = \frac{1}{2}(1+t)s$$

If a pentahedron element is needed, this element should be used instead of the collapsed solid element since it is, in general, more stable and significantly faster. Automatic sorting can be used, see *CONTROL_SOLID keyword, to segregate these elements in a mesh of 8 node solids for treatment as pentahedrons. Selective-reduced integration is used to prevent volumetric locking, i.e., a constant pressure over the domain of the element is assumed.

3.7 Fully Integrated Brick Element With 48 Degrees-of-Freedom

The forty-eight degree of freedom brick element is derived from the twenty node solid element; see Figure 3.5, through a transformation of the nodal displacements and rotations of the mid-side nodes [Yunus, Pawlak, and Cook, 1989]. This element has the advantage that shell nodes can be shared with brick nodes and that the faces have just four nodes — a real advantage for the contact-impact logic. The accuracy of this element is relatively good for problems in linear elasticity but degrades as Poisson's ratio approaches the incompressible limit. This can be remedied by using incompatible modes in the element formulation, but such an approach seems impractical for explicit computations.

The instantaneous velocity for a midside node k is given as a function of the corner node velocities as (See Figure 3.6),

$$\dot{u}_k = \frac{1}{2}\left(\dot{u}_i + \dot{u}_j\right) + \frac{y_j - y_i}{8}\left(\dot{\theta}_{zj} - \dot{\theta}_{zi}\right) + \frac{z_j - z_i}{8}\left(\dot{\theta}_{yi} - \dot{\theta}_{yj}\right)$$

$$\dot{v}_k = \frac{1}{2}\left(\dot{v}_i + \dot{v}_j\right) + \frac{z_j - z_i}{8}\left(\dot{\theta}_{xj} - \dot{\theta}_{xi}\right) + \frac{x_j - x_i}{8}\left(\dot{\theta}_{zi} - \dot{\theta}_{zj}\right) \tag{3.75}$$

$$\dot{w}_k = \frac{1}{2}\left(\dot{w}_i + \dot{w}_j\right) + \frac{x_j - x_i}{8}\left(\dot{\theta}_{yj} - \dot{\theta}_{yi}\right) + \frac{y_j - y_i}{8}\left(\dot{\theta}_{xi} - \dot{\theta}_{xj}\right)$$

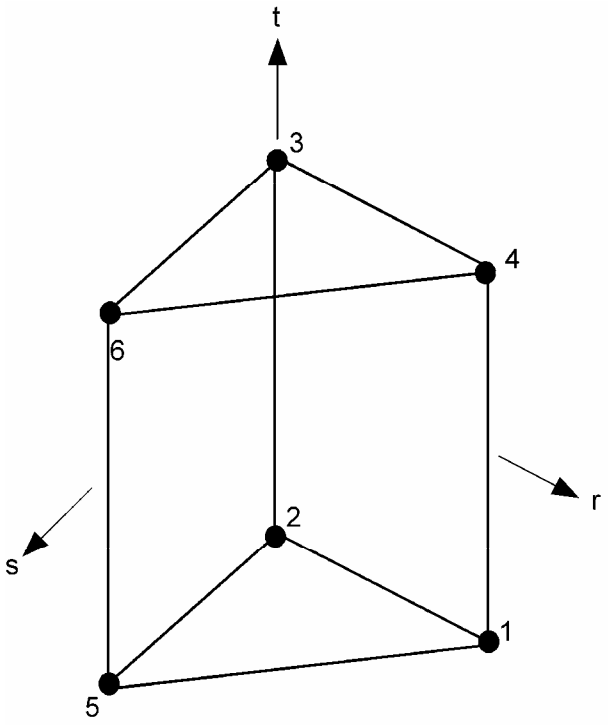


Figure 3.4. Six node Pentahedron.

where $u, v, w, \theta_x, \theta_y,$ and θ_z are the translational and rotational displacements in the global $x, y,$ and z directions. The velocity field for the twenty-node hexahedron element in terms of the nodal velocities is:

$$\begin{Bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{Bmatrix} = \begin{bmatrix} \phi_1 & \phi_2 \ldots \phi_{20} & 0 & 0 \ldots 0 & 0 & 0 \ldots 0 \\ 0 & 0 \ldots 0 & \phi_1 & \phi_2 \ldots \phi_{20} & 0 & 0 \ldots 0 \\ 0 & 0 \ldots 0 & 0 & 0 \ldots 0 & \phi_1 & \phi_2 \ldots \phi_{20} \end{bmatrix} \begin{Bmatrix} \dot{u}_1 \\ \vdots \\ \dot{u}_{20} \\ \dot{v}_1 \\ \vdots \\ \dot{v}_{20} \\ \dot{w}_1 \\ \vdots \\ \dot{w}_{20} \end{Bmatrix} \tag{3.76}$$

where ϕ_i are given by [Bathe and Wilson 1976] as,

$$\begin{aligned}
\phi_1 &= g_1 - \frac{(g_9 + g_{12} + g_{17})}{2} \qquad \phi_6 = g_6 - \frac{(g_{13} + g_{14} + g_{18})}{2} \\
\phi_2 &= g_2 - \frac{(g_9 + g_{10} + g_{18})}{2} \qquad \phi_7 = g_7 - \frac{(g_{14} + g_{15} + g_{19})}{2} \\
\phi_3 &= g_3 - \frac{(g_{10} + g_{11} + g_{19})}{2} \qquad \phi_8 = g_8 - \frac{(g_{15} + g_{16} + g_{20})}{2} \\
\phi_4 &= g_4 - \frac{(g_{11} + g_{12} + g_{20})}{2} \qquad \phi_i = g_i \ \ for\ j = 9,...20 \\
\phi_5 &= g_5 - \frac{(g_{13} + g_{16} + g_{17})}{2}
\end{aligned} \tag{3.77}$$

$$g_i = G(\xi, \xi_i)G(\eta, \eta_i)G(\zeta, \zeta_i)$$

$$G(\beta, \beta_i) = \frac{1}{2}(1 + \beta\beta i) \quad for \quad \beta_i = \pm 1; \quad \beta = \xi, \eta, \zeta$$

$$G(\beta, \beta_i) = 1 - \beta^2 \quad for \quad \beta_i = 0$$

The standard formulation for the twenty node solid element is used with the above transformations. The element is integrated with a fourteen point integration rule [Cook 1974]:

$$\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} f(\xi, \eta, \zeta)\, d\xi\, d\eta\, d\zeta =$$

$$B_6 \left[f(-b, 0, 0) + f(b, 0, 0) + f(0, -b, 0) + ...(6\, terms) \right] + \tag{3.78}$$

$$C_8 \left[f(-c, -c, -c) + f(c, -c, -c) + f(c, c, -c) + ...(8\, terms) \right]$$

where

$$B_6 = 0.8864265927977938 \quad b = 0.7958224257542215$$

$$C_8 = 0.3351800554016621 \quad c = 0.7587869106393281$$

Cook reports that this rule has nearly the same accuracy as the twenty-seven point Gauss rule, which is very costly. The difference in cost between eight point and fourteen point integration, though significant, is necessary to eliminate the zero energy modes.

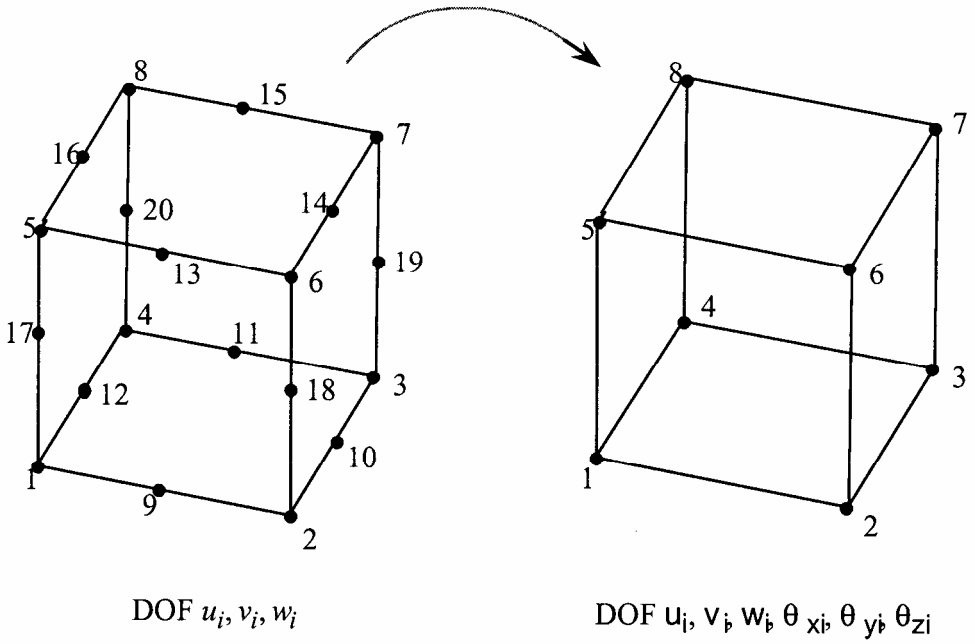


Figure 3.5. The 20-node solid element is transformed to an 8-node solid with 6 degrees-of-freedom per node.

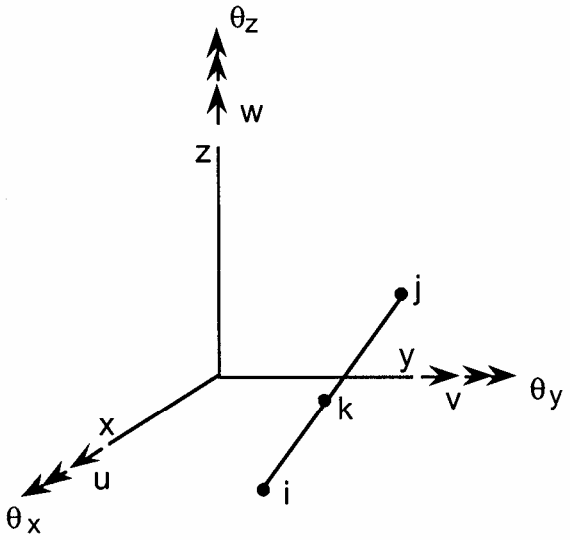


Figure 3.6. A typical element edge is shown from [Yunus, Pawlak, and Cook, 1989].

3.8 Fully Integrated Tetrahedron Element With 24 Degrees-of-Freedom

The twenty-four degree of freedom tetrahedron element is derived from the ten-node tetrahedron element; see Figure 3.7, following the same procedure used above for the forty-eight degree of freedom brick element [Yunus, Pawlak, and Cook, 1989]. This element has the advantage that shell nodes can be shared with its nodes and it is compatible with the brick element discussed above. The accuracy of this element is relatively good-at least when compared

to the constant strain tetrahedron element. This is illustrated by the bar impact example in Figure 3.8 which compares the 12 and 24 degree of freedom tetrahedron elements. The 12 degree-of-freedom tetrahedron displays severe volumetric locking.

In our implementation we have not strictly followed the reference. In order to prevent locking in applications that involve incompressible behavior, selective reduced integration is used with a total of 5 integration points. Although this is rather expensive, no zero energy modes exist. We use the same approach in determining the rotary mass that is used in the implementation of the shell elements.

Figures 3.9 and 3.10 show the construction of a hexahedron element from five and six tetrahedron elements, respectively. When two sides of the adjacent bricks made from five tetrahedrons are together, it is likely that four unique triangular segments exist. This creates a problem in LS-PREPOST, which uses the numbering as a basis for eliminating interior polygons prior to display. Consequently, the graphics in the post-processing phase can be considerably slower with the degeneration in Figure 3.9. However, marginally better results may be obtained with five tetrahedrons per hexahedron due to a better constraint count.

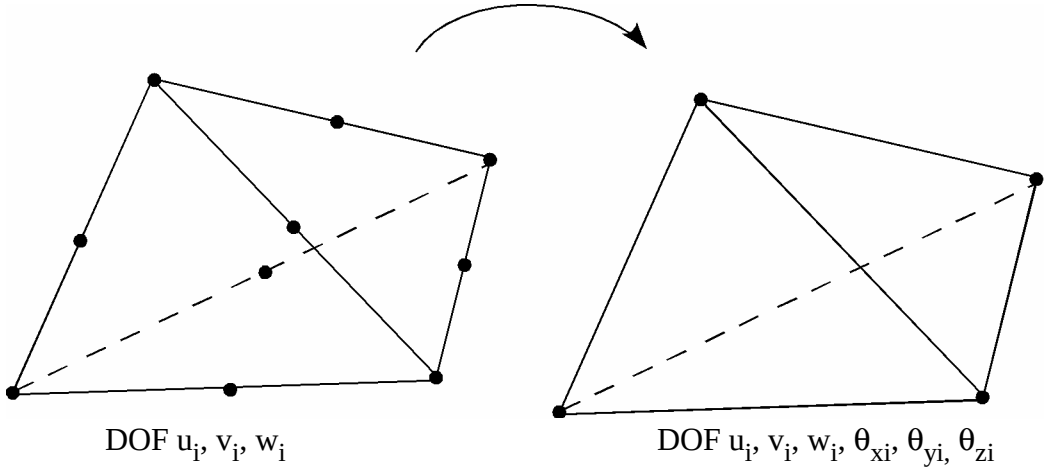


Figure 3.7. Twenty-four degrees of freedom tetrahedron element [Yunus, Pawlak, and Cook, 1989].

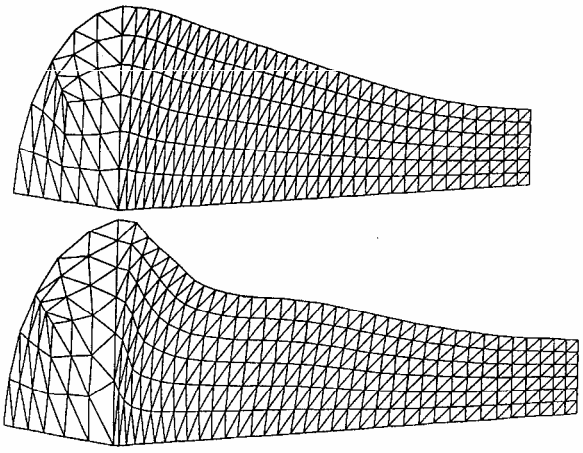


Figure 3.8. A comparison of the 12 and 24 degree-of-freedom tetrahedron elements is shown. The 12 degree-of-freedom tetrahedron element on the top displays severe volumetric locking.

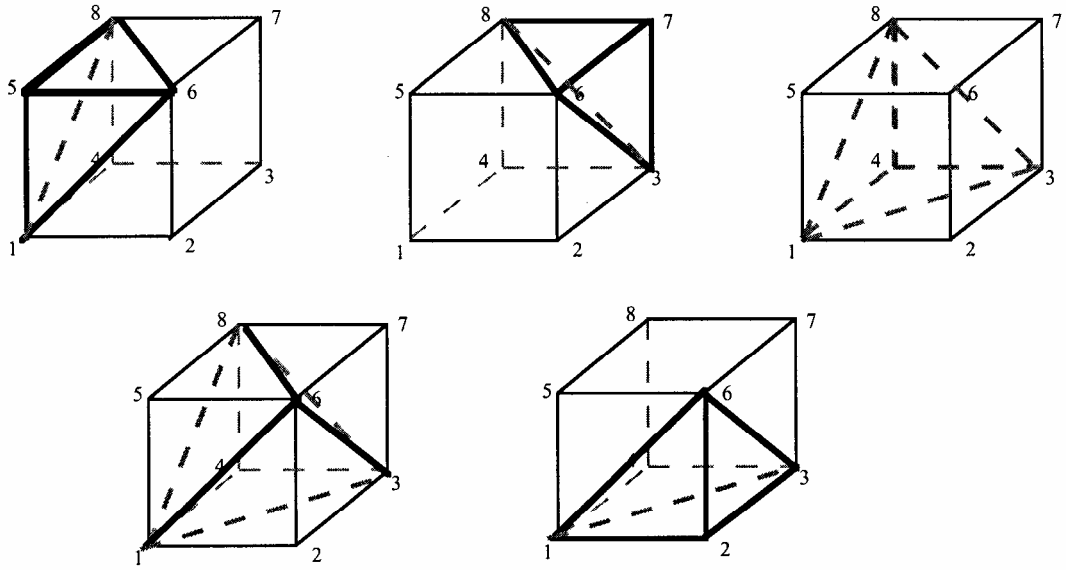


Figure 3.9. Construction of a hexahedron element with five tetrahedrons.

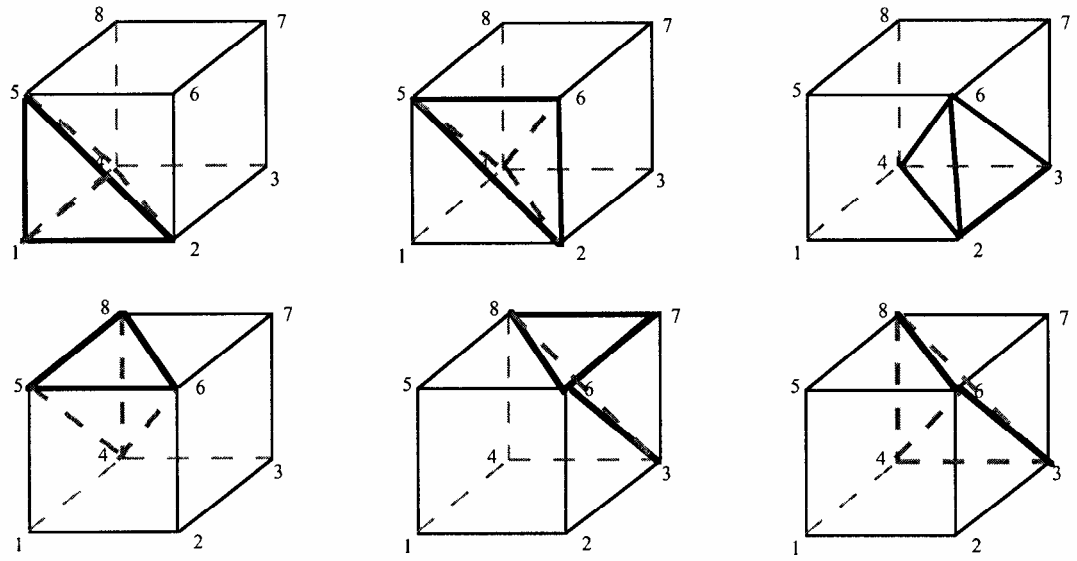


Figure 3.10. Construction of a hexahedron element with six tetrahedrons.

3.9 Integral Difference Scheme as Basis For 2D Solids

Two dimensional solid element in LS-DYNA include:

- Plane stress 2D element
- Plane strain 2D shell element
- Axisymmetric 2D Petrov-Galerkin (area weighted) element
- Axisymmetric 2D Galerkin (volume weighted) element

These elements have their origins in the integral difference method of Noh [1964] which is also used the HEMP code developed by Wilkins [1964, 1969]. In LS-DYNA, both two dimensional planar and axisymmetric geometries are defined in the xy plane. In axisymmetric geometry, however, the x axis corresponds to the radial direction and the y axis becomes the axis of symmetry. The integral difference method defines the components of the gradient of a function F in terms of the line integral about the contour S which encloses the area A:

$$\frac{\partial F}{\partial x} = \frac{\int_C F\,(n \cdot i)\,dS}{\underset{A \to 0}{Lim}\, A}$$

$$\frac{\partial F}{\partial y} = \frac{\int_C F\,(n \cdot j)\,dS}{\underset{A \to 0}{Lim}\, A} \tag{3.79}$$

Here, n is the normal vector to S and i and j are unit vectors in the x and y directions, respectively. See Figure 3.11.

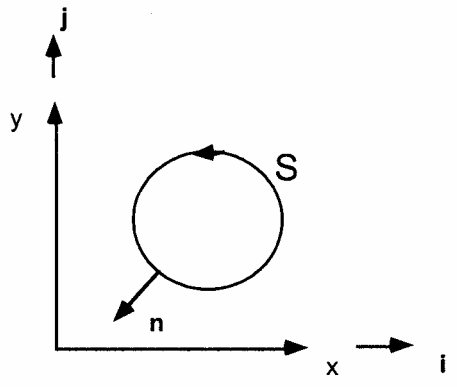


Figure 3.11. The contour S encloses an area A.

In this approach the velocity gradients which define the strain rates are element centered, and the velocities and nodal forces are node centered. See Figure 3.12.

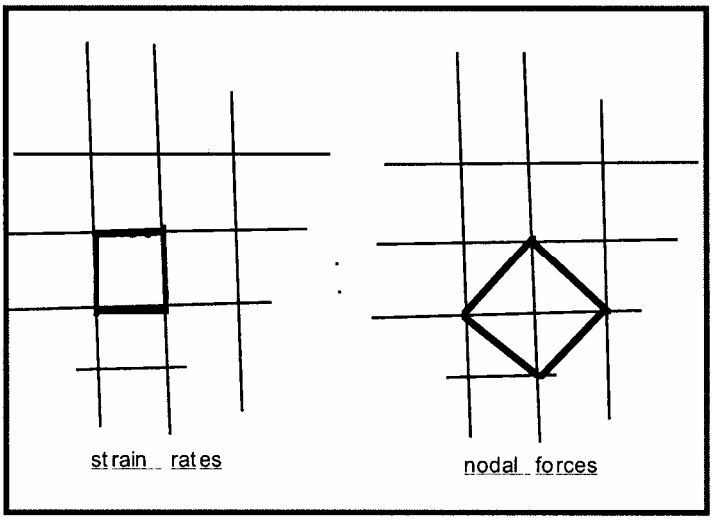


Figure 3.12. Strain rates are element centered and nodal forces are node centered.

Noting that the normal vector n is defined as:

$$n = \frac{\partial y}{\partial S} i + \frac{\partial x}{\partial S} j \tag{3.80}$$

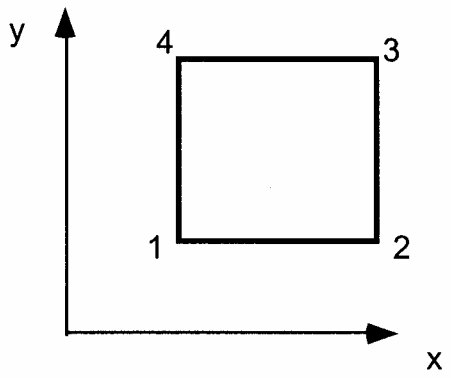


Figure 3.13. Element numbering.

and referring to Figure 3.13, we can expand the numerator in equation (3.36):

$$\int_c F\,(n\cdot i)\,dS = \int F\,\frac{\partial y}{\partial S}dS = F_{23}(y_3-y_2)+F_{34}(y_4-y_3)+F_{41}(y_1-y_4)+F_{12}(y_2-y_1) \tag{3.81}$$

where $F_{ij} = \left(F_i + F_j\right)/2$

Therefore, letting A again be the enclosed area, the following expressions are obtained:

$$\frac{\partial F}{\partial x} = \frac{F_{23}(y_3-y_2)+F_{34}(y_4-y_3)+F_{41}(y_1-y_4)+F_{12}(y_2-y_1)}{A}$$

or (3.82)

$$\frac{\partial F}{\partial x} = \frac{(F_2-F_4)(y_3-y_1)+(y_2-y_4)(F_3-F_1)}{2A}$$

Hence, the strain rates in the *x* and *y* directions become:

$$\frac{\partial F}{\partial x} = \frac{F_{23}(y_3-y_2)+F_{34}(y_4-y_3)+F_{41}(y_1-y_4)+F_{12}(y_2-y_1)}{A}$$

or (3.83a)

$$\frac{\partial F}{\partial x} = \frac{(F_2-F_4)(y_3-y_1)+(y_2-y_4)(F_3-F_1)}{2A}$$

$$\varepsilon_{yy} = \frac{\partial \dot{y}}{\partial y} = \frac{(\dot{y}_2-\dot{y}_4)(x_3-x_1)+(x_2-x_4)(\dot{y}_3-\dot{y}_1)}{2A} \tag{3.83b}$$

and the shear strain rate is given by:

$$\varepsilon_{xy} = \frac{1}{2}\left(\frac{\partial \dot{y}}{\partial x} + \frac{\partial \dot{x}}{\partial y}\right) \tag{3.83c}$$

where

$$\frac{\partial \dot{y}}{\partial x} = \frac{(\dot{y}_2 - \dot{y}_4)(y_3 - y_1) + (y_2 - y_4)(\dot{y}_3 - \dot{y}_1)}{2A} \tag{3.84a}$$

$$\frac{\partial \dot{x}}{\partial y} = \frac{(\dot{x}_2 - \dot{x}_4)(x_3 - x_1) + (x_2 - x_4)(\dot{x}_3 - \dot{x}_1)}{2A} \tag{3.84b}$$

The zero energy modes, called hourglass modes, as in the three dimensional solid elements, can be a significant problem. Consider the velocity field given by: $\dot{x}_3 = \dot{x}_1$, $\dot{x}_2 = \dot{x}_4$, $\dot{y}_3 = \dot{y}_1$, and $\dot{y}_2 = \dot{y}_4$. As can be observed from Equations (3.71) and (3.72), $\varepsilon_{xx} = \varepsilon_{yy} = \varepsilon_{xy} = 0$ and the element "hourglasses" irrespective of the element geometry. In the two-dimensional case, two modes exist versus twelve in three dimensions. The hourglass treatment for these modes is identical to the approach used for the shell elements, which are discussed later.

In two-dimensional planar geometries for plane stress and plane strain, the finite element method and the integral finite difference method are identical. The velocity strains are computed for the finite element method from the equation:

$$\dot{\varepsilon} = Bv \tag{3.85}$$

where $\dot{\varepsilon}$ is the velocity strain vector, B is the strain displacement matrix, and v is the nodal velocity vector. Equation (3.73) exactly computes the same velocity strains as the integral difference method if

$$B = B(s,t)\Big|_{s=t=0} \tag{3.86}$$

The update of the nodal forces also turns out to be identical. The momentum equations in two-dimensional planar problems are given by

$$\begin{aligned} \frac{1}{\rho}\left(\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y}\right) &= \ddot{x} \\ \frac{1}{\rho}\left(\frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y}\right) &= \ddot{y} \end{aligned} \tag{3.87}$$

Referring to Figure 3.14, the integral difference method gives Equation (3.76):

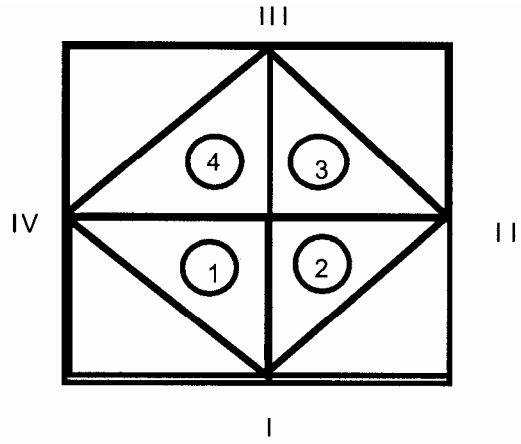


Figure 3.14. The finite difference stencil for computing nodal forces is shown.

$$\frac{1}{\rho}\frac{\partial \sigma_{xx}}{\partial x} = \frac{\sigma_{xx_1}(y_I - y_{IV}) + \sigma_{xx_2}(y_{II} - y_I) + \sigma_{xx_3}(y_{III} - y_{II}) + \sigma_{xx_4}(y_{IV} - y_{III})}{\frac{1}{2}(\rho_1 A_1 + \rho_2 A_2 + \rho_3 A_3 + \rho_4 A_4)} \tag{3.88}$$

An element wise assembly of the discretized finite difference equations is possible leading to a finite element like finite difference program. This approach is used in the DYNA2D program by Hallquist [1980].

In axisymmetric geometries additional terms arise that do not appear in planar problems:

$$\begin{aligned} \frac{1}{\rho}\left(\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{(\sigma_{xx} - \sigma_{\theta\theta})}{x}\right) &= \ddot{x} \\ \frac{1}{\rho}\left(\frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{x\theta}}{x}\right) &= \ddot{y} \end{aligned} \tag{3.89}$$

where again note that y is the axis of symmetry and x is the radial direction. The only difference between finite element approach and the finite difference method is in the treatment of the terms, which arise from the assumption of axisymmetry. In the finite difference method the radial acceleration is found from the calculation:

$$\begin{aligned} \ddot{x} = \frac{1}{2}\Bigg[&\frac{\sigma_{xx_1}(y_I - y_{IV}) + \sigma_{xx_2}(y_{II} - y_I) + \sigma_{xx_3}(y_{III} - y_{II}) + \sigma_{xx_4}(y_{IV} - y_{III})}{(\rho_1 A_1 + \rho_2 A_2 + \rho_3 A_3 + \rho_4 A_4)} - \\ &\frac{\sigma_{xy_1}(x_I - x_{IV}) + \sigma_{xy_2}(x_{II} - x_I) + \sigma_{xy_3}(x_{III} - x_{II}) + \sigma_{xy_4}(x_{IV} - x_{III})}{(\rho_1 A_1 + \rho_2 A_2 + \rho_3 A_3 + \rho_4 A_4)}\Bigg] + \beta \end{aligned} \tag{3.90}$$

where β_{fe} is found by a summation over the four surrounding elements:

$$\beta = \frac{1}{4}\sum_{i=1}^{4}\left[\frac{\left(\sigma_{xx_i} - \sigma_{\theta\theta_i}\right)}{\left(\rho x\right)_i}\right]_i, \tag{3.91}$$

x_i is the centroid of the *i*th element defined as the ratio of its volume V_i and area A_i:

$$x_i = \frac{V_i}{A_i}$$

$\sigma_{\theta\theta_i}$ is the hoop stress, and ρ_i is the *current* density.

When applying the Petrov-Galerkin finite element approach, the weighting functions are divided by the radius, r:

$$\begin{aligned}\int \frac{1}{r}\phi^t\left(\nabla\cdot\sigma + b - \rho\ddot{u}\right)dV \\ = \int \phi^t\left(\nabla\cdot\sigma + b - \rho\ddot{u}\right)dA = 0\end{aligned} \tag{3.92}$$

where the integration is over the current geometry. This is sometimes referred to as the "Area Galerkin" method. This approach leads to a time dependent mass vector. LS-DYNA also has an optional Galerkin axisymmetric element, which leads to a time independent mass vector. For structural analysis problems where pressures are low the Galerkin approach works best, but in problems of hydrodynamics where pressures are a large fraction of the elastic modulus, the Petrov-Galerkin approach is superior since the behavior along the axis of symmetry is correct.

The Petrov-Galerkin approach leads to equations similar to finite differences. The radial acceleration is given by.

$$\begin{aligned}\ddot{x} = \frac{1}{2}\Bigg[\frac{\sigma_{xx_1}\left(y_I - y_{IV}\right) + \sigma_{xx_2}\left(y_{II} - y_I\right) + \sigma_{xx_3}\left(y_{III} - y_{II}\right) + \sigma_{xx_4}\left(y_{IV} - y_{III}\right)}{\left(\rho_1 A_1 + \rho_2 A_2 + \rho_3 A_3 + \rho_4 A_4\right)} - \\ \frac{\sigma_{xy_1}\left(x_I - x_{IV}\right) + \sigma_{xy_2}\left(x_{II} - x_I\right) + \sigma_{xy_3}\left(x_{III} - x_{II}\right) + \sigma_{xy_4}\left(x_{IV} - x_{III}\right)}{\left(\rho_1 A_1 + \rho_2 A_2 + \rho_3 A_3 + \rho_4 A_4\right)}\Bigg] + \beta_{fe}\end{aligned} \tag{3.93}$$

where β_{fe} is now area weighted.

$$\beta_{fe} = \frac{1}{4\left(\rho_1 A_1 + \rho_2 A_2 + \rho_3 A_3 + \rho_4 A_4\right)}\sum_{i=1}^{4}\left[\frac{\left(\sigma_{xx_i} - \sigma_{\theta\theta_i}\right)A_i}{x_i}\right]_i \tag{3.94}$$

In LS-DYNA, the two-dimensional solid elements share the same constitutive subroutines with the three-dimensional elements. The plane stress element calls the plane stress constitutive models for shells. Similarly, the plane strain and axisymmetric elements call the full three-dimensional constitutive models for solid elements. Slight overheads exists since the strain

rate components $\dot{\varepsilon}_{yz}$ and $\dot{\varepsilon}_{zx}$ are set to zero in the two-dimensional case prior to updating the six stress component; consequently, the additional work is related to having six stresses whereas only four are needed. A slowdown of LS-DYNA compared with DYNA2D of fifteen percent has been observed; however, some of the added cost is due to the internal and hourglass energy calculations, which were not done in DYNA2D.

3.10 Rezoning With 2D Solid Elements

Lagrangian solution techniques generally function well for problems when element distortions are moderate. When distortions become excessive or when material breaks up, i.e., simply connected regions become multi-connected, these codes break down, and an Eulerian approach is a necessity. Between these two extremes, applications exist for which either approach may be appropriate but Lagrangian techniques are usually preferred for speed and accuracy. Rezoning may be used to extend the domain of application for Lagrangian codes.

Rezoning capability was added to DYNA2D in 1980 and to LS-DYNA in version 940. In the current implementation the rezoning can be done interactively and used to relocate the nodal locations within and on the boundary of parts. This method is sometimes referred to as *r-adaptive.*

The rezoning is accomplished in three steps listed below:

1. Generate nodal values for all variables to be remapped
2. Rezone one or more materials either interactively or automatically with command file.
3. Initialize remeshed regions by interpolating from nodal point values of old mesh.

In the first step each variable is approximated globally by a summation over the number of nodal points n:

$$g(r,z) = \sum_{i=1}^{n} \Phi_i(r,z) \tag{3.95}$$

where

Φ_i set of piece wise continuous global basis functions

Φ_i nodal point values

Given a variable to be remapped h(r, z), a least squares best fit is found by minimizing the functional

$$\Pi = \int (g-h)^2 dA \tag{3.96}$$

i.e.,

$$\frac{d\Pi}{dg_i} = 0 \qquad i = 1, 2, ..., n \tag{3.97}$$

This yields the set of matrix equations

$$Mg = f \tag{3.98}$$

where

$$M = \sum M^e = \sum_e \int \Phi \Phi^t dA \tag{3.99}$$

$$f = \sum f^e = \sum_e \int h \Phi dA \tag{3.100}$$

Lumping the mass makes the calculation of g trivial

$$M_i = \sum_j^n M_{ij} \tag{3.101}$$

$$g_i = \frac{f_i}{M_i} \tag{3.102}$$

In step 2, the interactive rezoning phase permits:

- Plotting of solution at current time
- Deletion of elements and slidelines
- Boundary modifications via dekinks, respacing nodes, etc.
- Mesh smoothing

A large number of interactive commands are available and are described in the Help package. Current results can be displayed by

- Color fringes
- Contour lines
- Vectors plots
- Principal stress lines
- Deformed meshes and material outlines
- Profile plots
- Reaction forces
- Interface pressures along 2D contact interfaces

Three methods are available for smoothing

- Equipotential
- Isoparametric
- Combination of equipotential and isoparametric.

In applying the relaxation, the new nodal positions are found and given by Equation (3.103)

$$x = \frac{\sum_{i=1}^{8} \xi_i \, x_i}{\sum_{i=1}^{8} \xi_i}$$

$$y = \frac{\sum_{i=1}^{8} \xi_i \, y_i}{\sum_{i=1}^{8} \xi_i} \qquad (3.103)$$

where the nodal positions relative to the node being moved are shown in the sketch in Figure 3.15.

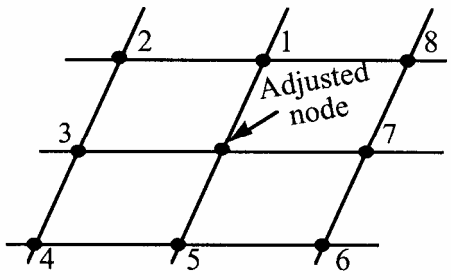


Figure 3.15. The stencil used to relax an interior nodal point.

The weights, ξ_i, for equipotential smoothing are

$$\xi_1 = \xi_5 = \frac{1}{4}\left[\left(x_7 - x_3\right)^2 + \left(y_7 - y_3\right)^2 \right]$$

$$\xi_3 = \xi_7 = \frac{1}{4}\left[\left(x_1 - x_5\right)^2 + \left(y_1 - y_5\right)^2 \right] \qquad (3.104a)$$

$$\xi_2 = \xi_6 = \frac{1}{2}\left[(x_1 - x_5)(x_7 - x_3) + (y_1 - y_5)(y_7 - y_3)\right]$$
$$\xi_4 = \xi_8 = -\xi_2 \tag{3.104b}$$

and are given by

$$\xi_1 = \xi_3 = \xi_5 = \xi_7 = .50$$
$$\xi_2 = \xi_4 = \xi_6 = \xi_8 = -.25 \tag{3.105}$$

for isoparametric smoothing. Since logical regularity is not assumed in the mesh, we construct the nodal stencil for each interior node and then relax it. The nodes are iteratively moved until convergence is obtained. In Chapter 14 of this manual, the smoothing procedures are discussed for three-dimensional applications.

The new element centered values, h^*, computed in Equation (3.106) are found by a 4 point Gauss Quadrature as illustrated in Figure 3.16.

$$h^* = \frac{\int g dA}{\int dA} \tag{3.106}$$

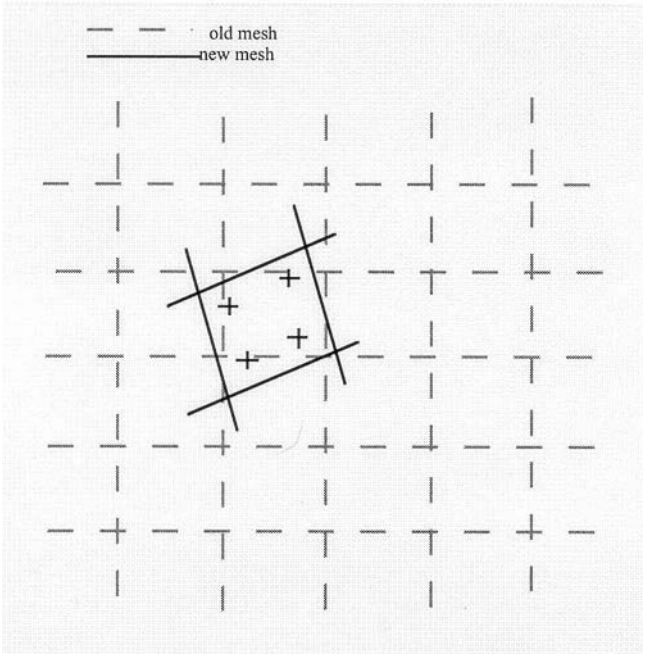


Figure 3.16. A four point Gauss quadrature rule over the new element is used to determine the new element centered value.

The Gauss point values are interpolated from the nodal values according to Equation (3.107). This is also illustrated by Figure 3.17.

$$g_a = \sum \phi_i (s_a, t_a) g_i \tag{3.107}$$

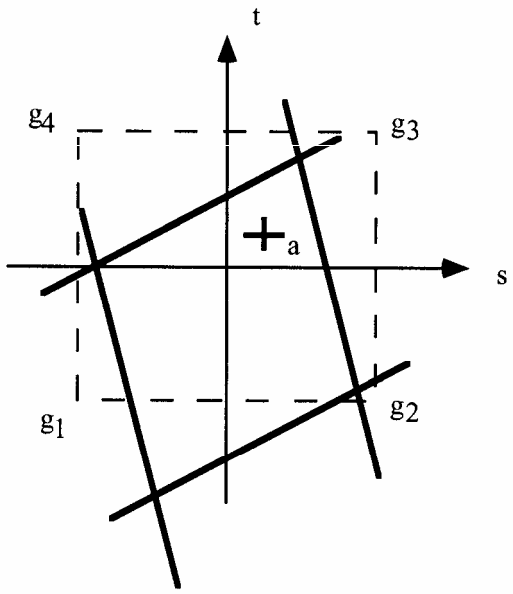


Figure 3.17. A four point Gauss quadrature rule over the new element is used to determine the new element centered value.

4. BELYTSCHKO BEAM

The Belytschko beam element formulation [Belytschko et al. 1977] is part of a family of structural finite elements, by Belytschko and other researchers that employ a 'co-rotational technique' in the element formulation for treating large rotation. This section discusses the co-rotational formulation, since the formulation is most easily described for a beam element, and then describes the beam theory used to formulate the co-rotational beam element.

4.1 Co-rotational Technique

In any large displacement formulation, the goal is to separate the deformation displacements from the rigid body displacements, as only the deformation displacements give rise to strains and the associated generation of strain energy. This separation is usually accomplished by comparing the current configuration with a reference configuration.

The current configuration is a complete description of the deformed body in its current spatial location and orientation, giving locations of all points (nodes) comprising the body. The reference configuration can be either the initial configuration of the body, i.e., nodal locations at time zero, or the configuration of the body at some other state (time). Often the reference configuration is chosen to be the previous configuration, say at time $t^n = t^{n+1} - \Delta t$.

The choice of the reference configuration determines the type of deformations that will be computed: total deformations result from comparing the current configuration with the initial configuration, while incremental deformations result from comparing with the previous configuration. In most time stepping (numerical) Lagrangian formulations, incremental deformations are used because they result in significant simplifications of other algorithms, chiefly constitutive models.

A direct comparison of the current configuration with the reference configuration does not result in a determination of the deformation, but rather provides the total (or incremental) displacements. We will use the unqualified term displacements to mean either the total displacements or the incremental displacements, depending on the choice of the reference configuration as the initial or the last state. This is perhaps most obvious if the reference configuration is the initial configuration. The direct comparison of the current configuration with the reference configuration yields displacements, which contain components due to deformations and rigid body motions. The task remains of separating the deformation and rigid body displacements. The deformations are usually found by subtracting from the displacements an estimate of the rigid body displacements. Exact rigid body displacements are usually only known for trivial cases where they are prescribed a priori as part of a displacement field. The co-rotational formulations provide one such estimate of the rigid body displacements.

The co-rotational formulation uses two types of coordinate systems: one system associated with each element, i.e., element coordinates which deform with the element, and another associated with each node, i.e., body coordinates embedded in the nodes. (The term 'body' is used to avoid possible confusion from referring to these coordinates as 'nodal' coordinates. Also, in the more general formulation presented in [Belytschko et al., 1977], the nodes could optionally be attached to rigid bodies. Thus the term 'body coordinates' refers to a system of coordinates in a rigid body, of which a node is a special case.) These two coordinate systems are shown in the upper portion of Figure 4.1(a).

The element coordinate system is defined to have the local x-axis $\hat{x}$ originating at node I and terminating at node J; the local y-axis $\hat{y}$ and, in three dimension, the local z-axis $\hat{z}$, are

constructed normal to $\hat{x}$. The element coordinate system $(\hat{x},\hat{y},\hat{z})$ and associated unit vector triad ($\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$) are updated at every time step by the same technique used to construct the initial system; thus the unit vector $\mathbf{e}_1$ deforms with the element since it always points from node I to node J.

The embedded body coordinate system is initially oriented along the principal inertial axes; either the assembled nodal mass or associated rigid body inertial tensor is used in determining the inertial principal values and directions. Although the initial orientation of the body axes is arbitrary, the selection of a principal inertia coordinate system simplifies the rotational equations of motion, i.e., no inertial cross product terms are present in the rotational equations of motion. Because the body coordinates are fixed in the node, or rigid body, they rotate and translate with the node and are updated by integrating the rotational equations of motion, as will be described subsequently.

The unit vectors of the two coordinate systems define rotational transformations between the global coordinate system and each respective coordinate system. These transformations operate on vectors with global components $\boldsymbol{A}=(A_x,A_y,A_z)$, body coordinates components $\overline{A}=\left(\overline{A}_x,\overline{A}_y,\overline{A}_z\right)$, and element coordinate components $\hat{A}=\left(\hat{A}_x,\hat{A}_y,\hat{A}_z\right)$ which are defined as:

$$A=\begin{Bmatrix}A_x\\A_y\\A_z\end{Bmatrix}=\begin{bmatrix}b_{1x}&b_{2x}&b_{3x}\\b_{1y}&b_{2y}&b_{3y}\\b_{1z}&b_{2z}&b_{3z}\end{bmatrix}\begin{Bmatrix}\overline{A}_x\\\overline{A}_y\\\overline{A}_z\end{Bmatrix}=[\lambda]\{\overline{A}\} \tag{4.1}$$

where b_{ix}, b_{iy}, b_{iz} are the global components of the body coordinate unit vectors. Similarly for the element coordinate system:

$$A=\begin{Bmatrix}A_x\\A_y\\A_z\end{Bmatrix}=\begin{bmatrix}e_{1x}&e_{2x}&e_{3x}\\e_{1y}&e_{2y}&e_{3y}\\e_{1z}&e_{2z}&e_{3z}\end{bmatrix}\begin{Bmatrix}\hat{A}_x\\\hat{A}_y\\\hat{A}_z\end{Bmatrix}=[\mu]\{\hat{A}\} \tag{4.2}$$

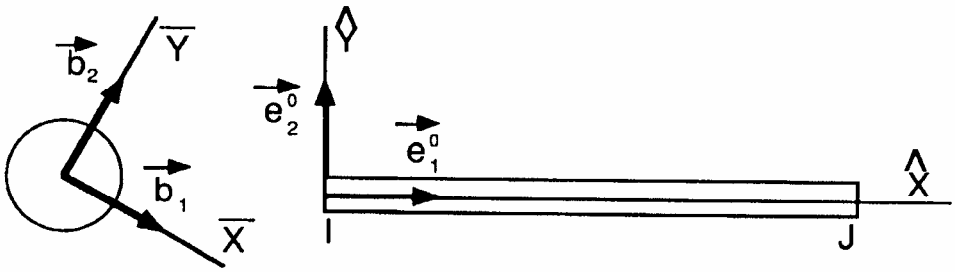
where e_{ix}, e_{iy}, e_{iz} are the global components of the element coordinate unit vectors. The inverse transformations are defined by the matrix transpose, i.e.,

$$\{\overline{A}\}=[\lambda]^T\{A\} \tag{4.3}$$

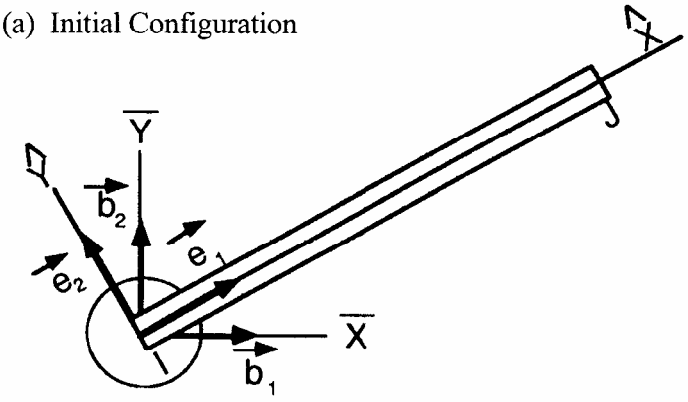
$$\{\hat{A}\}=[\mu]^T\{A\} \tag{4.4}$$

since these are proper rotational transformations.

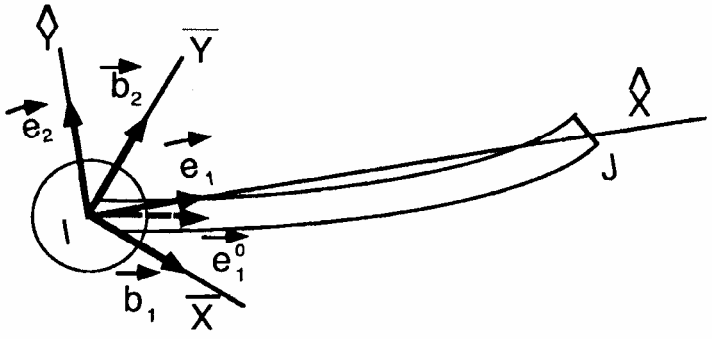
The following two examples illustrate how the element and body coordinate system are used to separate the deformations and rigid body displacements from the displacements:



(a) Initial Configuration



(b) Rigid Rotation Configuration



(c) Deformed Configuration

Figure 4.1. Co-rotational coordinate system: (a) initial configuration, (b) rigid rotational configuration and (c) deformed configuration.

Rigid Rotation. First, consider a rigid body rotation of the beam element about node I, as shown in the center of Figure 4.1(b), i.e., consider node I to be a pinned connection. Because the beam does not deform during the rigid rotation, the orientation of the unit vector $\boldsymbol{e}_1$ in the initial and rotated configuration will be the same with respect to the body coordinates. If the body coordinate components of the initial element unit vector e_1^0 were stored, they would be identical to the body coordinate components of the current element unit vector $\boldsymbol{e}_1$.

Deformation Rotation. Next, consider node I to be constrained against rotation, i.e., a clamped connection. Now node J is moved, as shown in the lower portion of Figure 4.1(c), causing the beam element to deform. The updated element unit vector $\boldsymbol{e}_1$ is constructed and its body coordinate components are compared to the body coordinate components of the original element unit vector e_1^0. Because the body coordinate system did not rotate, as node I was constrained, the original element unit vector and the current element unit vector are not colinear. Indeed, the angle between these two unit vectors is the amount of rotational deformation at node I, i.e.,

$$e_1 \times e_1^0 = \theta_\ell e_3 \tag{4.5}$$

Thus the co-rotational formulation separates the deformation and rigid body deformations by using:

- a coordinate system that deforms with the element, i.e., the element coordinates; or
- a coordinate system that rigidly rotates with the nodes, i.e., the body coordinates;

Then it compares the current orientation of the element coordinate system with the initial element coordinate system, using the rigidly rotated body coordinate system, to determine the deformations.

4.2 Belytschko Beam Element Formulation

The deformation displacements used in the Belytschko beam element formulation are:

$$\hat{d}^T = \left\{ \delta_{IJ}, \hat{\theta}_{xJI}, \hat{\theta}_{yI}, \hat{\theta}_{yJ}, \hat{\theta}_{zI}, \hat{\theta}_{zJ} \right\} \tag{4.6}$$

where

δ_{IJ}	=	length change
$\hat{\theta}_{xJI}$	=	torsional deformation
$\hat{\theta}_{yI}, \hat{\theta}_{yJ}, \hat{\theta}_{zI}, \hat{\theta}_{zJ}$	=	bending rotational deformations

The superscript ^ emphasizes that these quantities are defined in the local element coordinate system, and I and J are the nodes at the ends of the beam.

The beam deformations, defined above in Equation (4.6), are the usual small displacement beam deformations (see, for example, [Przemieniecki 1986]). Indeed, one advantage of the co-rotational formulation is the ease with which existing small displacement element formulations can be adapted to a large displacement formulation having small deformations in the element system. Small deformation theories can be easily accommodated because the definition of the local element coordinate system is independent of rigid body rotations and hence deformation displacement can be defined directly.

4.2.1 Calculation of Deformations

The elongation of the beam is calculated directly from the original nodal coordinates (X_I, Y_I, Z_I) and the total displacements (u_{xI}, u_{yI}, u_{zI}):

$$\delta_{IJ} = \frac{1}{l + l^o}\left[2\left(X_{JI}u_{xJI} + Y_{JI}u_{yJI} + Z_{JI}u_{zJI}\right) + u_{xJI}^2 + u_{yJI}^2 + u_{zJI}^2\right] \tag{4.7}$$

where

$$X_{JI} = X_J - X_I \tag{4.8}$$

$$u_{xJI} = u_{xJ} - u_{xI} \quad etc. \tag{4.9}$$

The deformation rotations are calculated using the body coordinate components of the original element coordinate unit vector along the beam axis, i.e., e_1^0, as outlined in the previous section. Because the body coordinate components of initial unit vector e_1^0 rotate with the node, in the deformed configuration it indicates the direction of the beam's axis if no deformations had occurred. Thus comparing the initial unit vector e_1^0 with its current orientation e_1 indicates the magnitude of deformation rotations. Forming the vector cross product between e_1^0 and e_1 :

$$e_1 \times e_1^0 = \hat{\theta}_y e_2 + \hat{\theta}_z e_3 \tag{4.10}$$

where

$\hat{\theta}_y$ is the incremental deformation about the local $\hat{y}$ axis

$\hat{\theta}_z$ is the incremental deformation about the local $\hat{z}$ axis

The calculation is most conveniently performed by transforming the body components of the initial element vector into the current element coordinate system:

$$\begin{Bmatrix} \hat{e}_{1x}^0 \\ \hat{e}_{1y}^0 \\ \hat{e}_{1z}^0 \end{Bmatrix} = [\mu]^T [\lambda] \begin{Bmatrix} \overline{e}_{1x}^0 \\ \overline{e}_{1y}^0 \\ \overline{e}_{1z}^0 \end{Bmatrix} \tag{4.11}$$

Substituting the above into Equation (4.10)

$$e_1 \times e_1^0 = \det \begin{bmatrix} e_1 & e_2 & e_3 \\ 1 & 0 & 0 \\ \hat{e}_{1x}^0 & \hat{e}_{1y}^0 & \hat{e}_{1z}^0 \end{bmatrix} = -\hat{e}_{1z}^0 e_2 + \hat{e}_{1y}^0 e_3 = \hat{\theta}_y e_2 + \hat{\theta}_z e_3 \tag{4.12}$$

Thus,

$$\hat{\theta}_y = -\hat{e}_{1z}^0 \tag{4.13}$$

$$\hat{\theta}_z = \hat{e}_{1y}^0 \tag{4.14}$$

The torsional deformation rotation is calculated from the vector cross product of initial unit vectors, from each node of the beam, that were normal to the axis of the beam, i.e., $\hat{e}_{2I}^0$ and $\hat{e}_{2J}^0$; note that $\hat{e}_{3I}^0$ and $\hat{e}_{3J}^0$ could also be used. The result from this vector cross product is then projected onto the current axis of the beam, i.e.,

$$\hat{\theta}_{xJI} = e_1 \cdot \left(\hat{e}_{2I}^0 \times \hat{e}_{2J}^0 \right) = e_1 \det \begin{bmatrix} e_1 & e_2 & e_3 \\ \hat{e}_{x2I}^0 & \hat{e}_{y2I}^0 & \hat{e}_{z2I}^0 \\ \hat{e}_{x2J}^0 & \hat{e}_{y2J}^0 & \hat{e}_{z2J}^0 \end{bmatrix} = \hat{e}_{y2I}^0 \hat{e}_{z2J}^0 - \hat{e}_{y2J}^0 \hat{e}_{z2I}^0 \tag{4.15}$$

Note that the body components of $\bar{e}_{2I}^0$ and $\bar{e}_{2J}^0$ are transformed into the current element coordinate system before performing the indicated vector products.

4.2.2 Calculation of Internal Forces

There are two methods for computing the internal forces for the Belytschko beam element formulation:

1. functional forms relating the overall response of the beam, e.g., moment-curvature relations,
2. direct through-the-thickness integration of the stress.

Currently only the former method, as explained subsequently, is implemented; the direct integration method is detailed in [Belytschko et al., 1977].

Axial Force. The internal axial force is calculated from the elongation of the beam δ, as given by Equation (4.7), and an axial stiffness:

$$\hat{f}_{xJ} = K^a \delta \tag{4.16}$$

where

$K^a = AE/l^0$ is the axial stiffness
A is the cross sectional area of the beam
E is Young's Modulus
l^0 is the original length of the beam

Bending Moments. The bending moments are related to the deformation rotations by

$$\begin{Bmatrix} \hat{m}_{yI} \\ \hat{m}_{yJ} \end{Bmatrix} = \frac{K_y^b}{1+\phi_y} \begin{bmatrix} 4+\phi_y & 2-\phi_y \\ 2-\phi_y & 4+\phi_y \end{bmatrix} \begin{Bmatrix} \hat{\theta}_{yI} \\ \hat{\theta}_{yJ} \end{Bmatrix} \tag{4.17}$$

$$\begin{Bmatrix} \hat{m}_{zI} \\ \hat{m}_{zJ} \end{Bmatrix} = \frac{K_z^b}{1+\phi_z} \begin{bmatrix} 4+\phi_z & 2-\phi_z \\ 2-\phi_z & 4+\phi_z \end{bmatrix} \begin{Bmatrix} \hat{\theta}_{zI} \\ \hat{\theta}_{zJ} \end{Bmatrix} \tag{4.18}$$

where Equation (4.17) is for bending in the $\hat{x}$–$\hat{z}$ plane and Equation (4.18) is for bending in the $\hat{x}$–$\hat{y}$ plane. The bending constants are given by

$$K_y^b = \frac{EI_{yy}}{l^0} \tag{4.19}$$

$$K_z^b = \frac{EI_{zz}}{l^0} \tag{4.20}$$

$$I_{yy} = \int\int \hat{z}^2 \, d\hat{y} d\hat{z} \tag{4.21}$$

$$I_{zz} = \int\int \hat{y}^2 \, d\hat{y} d\hat{z} \tag{4.22}$$

$$\phi_y = \frac{12EI_{yy}}{GA_s l^2} \qquad \phi_z = \frac{12EI_{zz}}{GA_s l^2} \tag{4.23}$$

Hence ϕ is the shear factor, G the shear modulus, and A_s is the effective area in shear.

Torsional Moment. The torsional moment is calculated from the torsional deformation rotation as

$$\hat{m}_{xJ} = K^t \hat{\theta}_{xJI} \tag{4.24}$$

where

$$K^t = \frac{GJ}{l^0} \tag{4.25}$$

$$J = \int\int \hat{y}\hat{z}\ d\hat{y}d\hat{z} \tag{4.26}$$

The above forces are conjugate to the deformation displacements given previously in Equation (4.6), i.e.,

$$\hat{d}^T = \left\{\delta_{IJ}, \hat{\theta}_{xJI}, \hat{\theta}_{yI}, \hat{\theta}_{yJ}, \hat{\theta}_{zI}, \hat{\theta}_{zJ}\right\} \tag{4.6}$$

where

$$\left\{\hat{d}\right\}^T \left\{\hat{f}\right\} = W^{\text{int}} \tag{4.27}$$

$$\hat{f}^T = \left\{\hat{f}_{xJ}, \hat{m}_{xJ}, \hat{m}_{yI}, \hat{m}_{yJ}, \hat{m}_{zI}, \hat{m}_{zJ}\right\} \tag{4.28}$$

The remaining internal force components are found from equilibrium:

$$\begin{aligned} \hat{f}_{xI} &= -\hat{f}_{xJ} & \hat{m}_{xI} &= -\hat{m}_{xJ} \\ \hat{f}_{zI} &= -\frac{\hat{m}_{yI} + \hat{m}_{yJ}}{l^0} & \hat{f}_{zI} &= -\hat{f}_{zJ} \\ \hat{f}_{yJ} &= -\frac{\hat{m}_{zI} + \hat{m}_{zJ}}{l^0} & \hat{f}_{yI} &= -\hat{f}_{yJ} \end{aligned} \tag{4.29}$$

4.2.3 Updating the Body Coordinate Unit Vectors

The body coordinate unit vectors are updated using the Newmark β-Method [Newmark 1959] with $\beta = 0$, which is almost identical to the central difference method [Belytschko 1974]. In particular, the body component unit vectors are updated using the formula

$$b_i^{j+1} = b_i^j + \Delta t \frac{db_i^j}{dt} + \frac{\Delta t^2}{2}\frac{d^2 b_i^j}{dt^2} \tag{4.30}$$

where the superscripts refer to the time step and the subscripts refer to the three unit vectors comprising the body coordinate triad. The time derivatives in the above equation are replaced by their equivalent forms from vector analysis:

$$\frac{db_i^j}{dt} = \omega \times b_i \tag{4.31}$$

$$\frac{d^2 b_i^j}{dt^2} = \omega \times (\omega \times b_i) + (\alpha_i \times b_i) \tag{4.32}$$

where ω and α are vectors of angular velocity and acceleration, respectively, obtained from the rotational equations of motion. With the above relations substituted into Equation (4.30), the update formula for the unit vectors becomes

$$b_i^{j+1} = b_i^j + \Delta t (\omega \times b_i) + \frac{\Delta t^2}{2}\left\{\left[\omega \times (\omega \times b_i) + (\alpha_i \times b_i)\right]\right\} \tag{4.33}$$

To obtain the formulation for the updated components of the unit vectors, the body coordinate system is temporarily considered to be fixed and then the dot product of Equation (4.33) is formed with the unit vector to be updated. For example, to update the $\bar{x}$ component of b_3, the dot product of Equation (4.33), with $i = 3$, is formed with b_1, which can be simplified to the relation

$$\bar{b}_{x3}^{j+1} = b_1^j \cdot b_{x3}^{j+1} = \Delta t \omega_y^j + \frac{\Delta t^2}{2}\left(\omega_x^j \omega_z^j + \alpha_y^j\right) \tag{4.34}$$

Similarly,

$$\bar{b}_{y3}^{j+1} = b_2^j \cdot b_3^{j+1} = \Delta t \omega_x^j + \frac{\Delta t^2}{2}\left(\omega_y^j \omega_z^j + \alpha_x^j\right) \tag{4.35}$$

$$\bar{b}_{y1}^{j+1} = b_1^j \cdot b_2^{j+1} = \Delta t \omega_z^j + \frac{\Delta t^2}{2}\left(\omega_x^j \omega_y^j + \alpha_z^j\right) \tag{4.36}$$

The remaining components b_3^{j+1} and b_1^{j+1} are found by using normality and orthogonality, where it is assumed that the angular velocities w are small during a time step so that the quadratic terms in the update relations can be ignored. Since b_3^{j+1} is a unit vector, normality provides the relation

$$\bar{b}_{z3}^{j+1} = \sqrt{1 - \left(\bar{b}_{x3}^{j+1}\right)^2 - \left(\bar{b}_{y3}^{j+1}\right)^2} \tag{4.37}$$

Next, if it is assumed that $\bar{b}_{x1}^{j+1} \approx 1$, orthogonality yields

$$\bar{b}_{z1}^{j+1} = -\frac{\bar{b}_{x3}^{j+1} + \bar{b}_{y1}^{j+1}\bar{b}_{y3}^{j+1}}{\bar{b}_{z3}^{j+1}} \tag{4.38}$$

The component $\bar{b}_{x1}^{j+1}$ is then found by enforcing normality:

$$\bar{b}_{x1}^{j+1} = \sqrt{1-\left(\bar{b}_{y1}^{j+1}\right)^2 - \left(\bar{b}_{z1}^{j+1}\right)^2} \tag{4.39}$$

The updated components of b_1 and b_3 are defined relative to the body coordinates at time step j. To complete the update and define the transformation matrix, Equation (4.1), at time step j+1, the updated unit vectors b_1 and b_3 are transformed to the global coordinate system, using Equation (4.1) with $[\lambda]$ defined at step j, and their vector cross product is used to form b_2

5. HUGHES-LIU BEAM

The Hughes-Liu beam element formulation, based on the shell [Hughes and Liu 1981a, 1981b] discussed later, was the first beam element we implemented. It has several desirable qualities:

- it is incrementally objective (rigid body rotations do not generate strains), allowing for the treatment of finite strains that occur in many practical applications;
- it is simple, which usually translates into computational efficiency and robustness
- it is compatible with the brick elements, because the element is based on a degenerated brick element formulation;
- it includes finite transverse shear strains. The added computations needed to retain this strain component, compare to those for the assumption of no transverse shear strain, are insignificant.

5.1 Geometry

The Hughes-Liu beam element is based on a degeneration of the isoparametric 8-node solid element, an approach originated by Ahmad et al., [1970]. Recall the solid element isoparametric mapping of the biunit cube

$$x(\xi,\eta,\zeta) = N_a(\xi,\eta,\zeta)\, x_a \tag{5.1}$$

$$N_a(\xi,\eta,\zeta) = \frac{(1+\xi_a\xi)(1+\eta_a\eta)(1+\zeta_a\zeta)}{8} \tag{5.2}$$

where x is an arbitrary point in the element, $(\xi,\ \eta,\ \zeta)$ are the parametric coordinates, x_a are the global nodal coordinates of node a, and N_a are the element shape functions evaluated at node a, i.e., (ξ_a,η_a,ζ_a) are (ξ,η,ζ) evaluated at node a.

In the beam geometry, ξ determines the location along the axis of the beam and the coordinate pair (η,ζ) defines a point on the cross section. To degenerate the 8-node brick geometry into the 2-node beam geometry, the four nodes at $\xi = -1$ and at $\xi = 1$ are combined into a single node with three translational and three rotational degrees of freedom. Orthogonal, inextensible nodal fibers are defined at each node for treating the rotational degrees of freedom. Figure 5.1 shows a schematic of the biunit cube and the beam element. The mapping of the biunit cube into the beam element is separated into three parts:

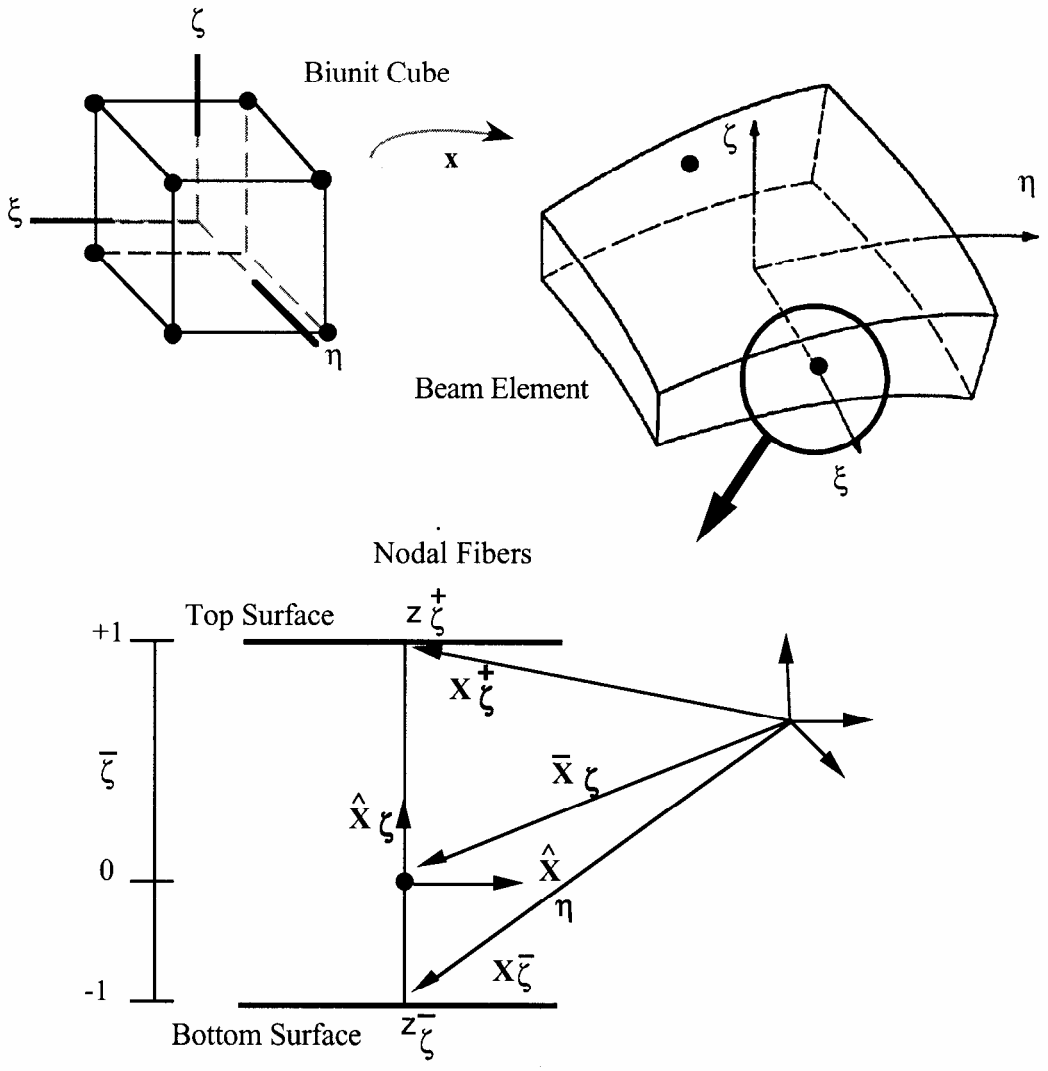


Figure 5.1 Hughes-Liu beam element.

$$x(\xi,\eta,\zeta) = \bar{x}(\xi) + X(\xi,\eta,\zeta) = \bar{x}(\xi) + X_\zeta(\xi,\zeta) + X_\eta(\xi,\eta) \tag{5.3}$$

where $\bar{x}$ denotes a position vector to a point on the reference axis of the beam, and X_ζ and X_η are position vectors at point $\bar{x}$ on the axis that define the fiber directions through that point. In particular,

$$\bar{x}\ (\xi) = N_a\ (\xi)\ \bar{x}_a \tag{5.4}$$

$$X_\eta\ (\xi,\ \eta) = N_a\ (\xi)\ X_{\eta a}\ (\eta) \tag{5.5}$$

$$X_\zeta\ (\xi,\zeta) = N_a(\xi)\ X_{\zeta a}\ (\zeta) \tag{5.6}$$

With this description, arbitrary points on the reference line $\bar{x}$ are interpolated by the one-dimensional shape function $N(\xi)$ operating on the global position of the two beam nodes that define the reference axis, i.e., $\bar{x}_a$. Points off the reference axis are further interpolated by using a one-dimensional shape function along the fiber directions, i.e., $X_{\eta}a(\eta)$ and $X_{\zeta\ a}(\zeta)$ where

$$X_{\eta a}(\eta) = z_{\eta}(\eta)\,\hat{X}_{\eta a} \tag{5.7a}$$

$$z_{\eta}(\eta) = N_{+}(\eta)\,z^{+}_{\eta a} + N_{-}(\eta)\,z^{-}_{\eta a} \tag{5.7b}$$

$$N_{+}(\eta) = \frac{(1+\eta)}{2} \tag{5.7c}$$

$$N_{-}(\eta) = \frac{(1-\eta)}{2} \tag{5.7d}$$

$$X_{\zeta a}(\zeta) = z_{\zeta}(\zeta)\,\hat{X}_{\zeta a} \tag{5.8a}$$

$$z_{\zeta}(\zeta) = N_{+}(\zeta)\,z^{+}_{\zeta a} + N_{-}(\zeta)\,z^{-}_{\zeta a} \tag{5.8b}$$

$$N_{+}(\zeta) = \frac{(1+\zeta)}{2} \tag{5.8c}$$

$$N_{-}(\zeta) = \frac{(1-\zeta)}{2} \tag{5.8d}$$

where $z_{\zeta}(\zeta)$ and $z_{\eta}(\eta)$ are "thickness functions".

The Hughes-Liu beam formulation uses four position vectors, in addition to ξ, to locate the reference axis and define the initial fiber directions. Consider the two position vectors $x^{+}_{\zeta a}$ and $x^{-}_{\zeta a}$ located on the top and bottom surfaces, respectively, at node a. Then

$$\bar{x}_{\zeta a} = \frac{1}{2}\left(1-\bar{\zeta}\right)x^{-}_{\zeta a} + \left(1+\bar{\zeta}\right)x^{+}_{\zeta a} \tag{5.9a}$$

$$\hat{X}_{\zeta a} = \frac{\left(x^{+}_{\zeta a} - x^{-}_{\zeta a}\right)}{\left\|x^{+}_{\zeta a} - x^{-}_{\zeta a}\right\|} \tag{5.9b}$$

$$z^{+}_{\zeta a} = \frac{1}{2}\left(1-\bar{\zeta}\right)\cdot\left\|x^{+}_{\zeta a} - x^{-}_{\zeta a}\right\| \tag{5.9c}$$

$$z_{\zeta a}^{-}=-\frac{1}{2}\left(1+\bar{\zeta}\right)\cdot\left\|x_{\zeta a}^{+}-x_{\zeta a}^{-}\right\| \tag{5.9d}$$

$$\bar{x}_{\eta a}=\frac{1}{2}\left(1-\bar{\zeta}\right)x_{\eta a}^{-}+\left(1+\bar{\zeta}\right)x_{\eta a}^{+} \tag{5.10a}$$

$$\hat{X}_{\eta a}=\frac{\left(x_{\eta a}^{+}-x_{\eta a}^{-}\right)}{\left\|x_{\eta a}^{+}-x_{\eta a}^{-}\right\|} \tag{5.10b}$$

$$z_{\eta a}^{+}=\frac{1}{2}\left(1-\bar{\eta}\right)\cdot\left\|x_{\eta a}^{+}-x_{\eta a}^{-}\right\| \tag{5.10c}$$

$$z_{\eta a}^{-}=-\frac{1}{2}\left(1+\bar{\eta}\right)\cdot\left\|x_{\eta a}^{+}-x_{\eta a}^{-}\right\| \tag{5.10d}$$

where $\left\|\cdot\right\|$ is the Euclidean norm. The reference surface may be located at the midsurface of the beam or offset at the outer surfaces. This capability is useful in several practical situations involving contact surfaces, connection of beam elements to solid elements, and offsetting elements such as for beam stiffeners in stiffened shells. The reference surfaces are located within the beam element by specifying the value of the parameters $\bar{\eta}$ and $\bar{\zeta}$, (see lower portion of Figure 5.1). When these parameters take on the values –1 or +1, the reference axis is located on the outer surfaces of the beam. If they are set to zero, the reference axis is at the center.

The same parametric representation used to describe the geometry of the beam elements is used to interpolate the beam element displacements, i.e., an isoparametric representation. Again the displacements are separated into the reference axis displacements and rotations associated with the fiber directions:

$$u\left(\xi,\eta,\zeta\right)=\bar{u}\left(\xi\right)+U\left(\xi,\eta,\zeta\right)=\bar{u}\left(\xi\right)+U_{\zeta}\left(\xi,\zeta\right)+U_{\eta}\left(\xi,\eta\right) \tag{5.11a}$$

$$\bar{u}\left(\xi\right)=N_{a}\left(\xi\right)\bar{u}_{a} \tag{5.11b}$$

$$U_{\eta}\left(\xi,\eta\right)=N_{a}\left(\xi\right)U_{\eta a}\left(\eta\right) \tag{5.11c}$$

$$U_{\zeta}\left(\xi,\zeta\right)=N_{a}\left(\xi\right)U_{\zeta a}\left(\zeta\right) \tag{5.11d}$$

$$U_{\eta a}\left(\eta\right)=z_{\eta a}\left(\eta\right)\hat{U}_{\eta a} \tag{5.11e}$$

$$U_{\zeta a}\left(\zeta\right)=z_{\zeta a}\left(\zeta\right)\hat{U}_{\zeta a} \tag{5.11f}$$

where u is the displacement of a generic point, $\bar{u}$ is the displacement of a point on the reference surface, and U is the 'fiber displacement' rotations. The motion of the fibers can be interpreted as either displacements or rotations as will be discussed.

Hughes and Liu introduced the notation that follows, and the associated schematic shown in Figure 5.2, to describe the current deformed configuration with respect to the reference configuration:

$$y = \bar{y} + Y \tag{5.12a}$$

$$\bar{y} = \bar{x} + \bar{u} \tag{5.12b}$$

$$\bar{y}_a = \bar{x}_a + \bar{u}_a \tag{5.12c}$$

$$Y = X + U \tag{5.12d}$$

$$Y_a = X_a + U_a \tag{5.12e}$$

$$\hat{Y}_{\eta a} = \hat{X}_{\eta a} + \hat{U}_{\eta a} \tag{5.12f}$$

$$\hat{Y}_{\zeta a} = \hat{X}_{\zeta a} + \hat{U}_{\zeta a} \tag{5.12g}$$

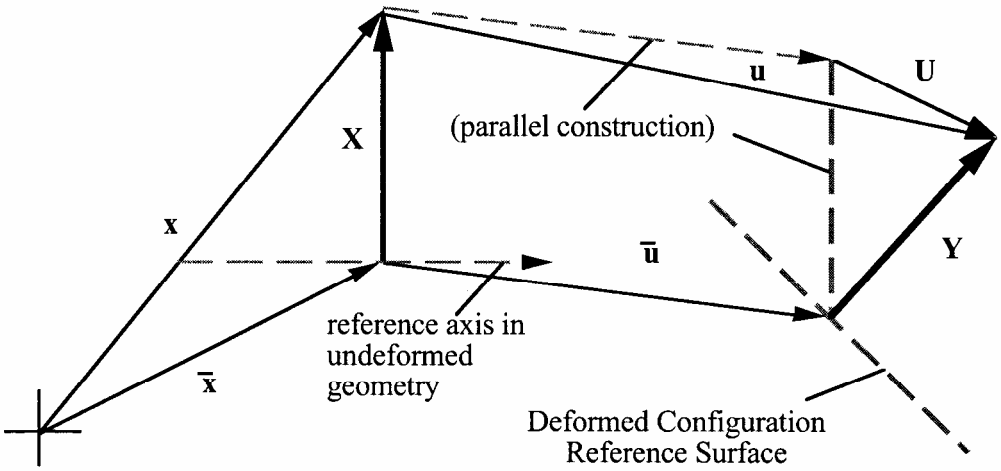


Figure 5.2. Schematic of deformed configuration displacements and position vectors.

In the above relations, and in Figure 5.2, the **x** quantities refer to the reference configuration, the **y** quantities refer to the updated (deformed) configuration and the **u** quantities are the displacements. The notation consistently uses a superscript bar $(\bar{\cdot})$ to indicate reference surface quantities, a superscript caret $(\hat{\cdot})$ to indicate unit vector quantities, lower case letter for translational displacements, and upper case letters for fiber displacements. Thus to update to the deformed configuration, two vector quantities are needed: the reference surface displacement $\bar{u}$

and the associated nodal fiber displacement $\boldsymbol{U}$. The nodal fiber displacements are defined in the fiber coordinate system, described in the next subsection.

5.2 Fiber Coordinate System

For a beam element, the known quantities will be the displacements of the reference surface $\bar{u}$ obtained from the translational equations of motion and the rotational quantities at each node obtained from the rotational equations of motion. What remains to complete the kinematics is a relation between nodal rotations and fiber displacements $\boldsymbol{U}$. The linearized relationships between the incremental components $\Delta\hat{U}$ the incremental rotations are given by

$$\begin{Bmatrix} \Delta\hat{U}_{\eta 1} \\ \Delta\hat{U}_{\eta 2} \\ \Delta\hat{U}_{\eta 3} \end{Bmatrix} = \begin{bmatrix} 0 & \hat{Y}_{\eta 3} & -\hat{Y}_{\eta 2} \\ -\hat{Y}_{\eta 3} & 0 & \hat{Y}_{\eta 1} \\ \hat{Y}_{\eta 2} & -\hat{Y}_{\eta 1} & 0 \end{bmatrix} \begin{Bmatrix} \Delta\theta_1 \\ \Delta\theta_2 \\ \Delta\theta_3 \end{Bmatrix} = h_\eta \Delta\theta \tag{5.13a}$$

$$\begin{Bmatrix} \Delta\hat{U}_{\zeta 1} \\ \Delta\hat{U}_{\zeta 2} \\ \Delta\hat{U}_{\zeta 3} \end{Bmatrix} = \begin{bmatrix} 0 & \hat{Y}_{\zeta 3} & -\hat{Y}_{\zeta 2} \\ -\hat{Y}_{\zeta 3} & 0 & \hat{Y}_{\zeta 1} \\ \hat{Y}_{\zeta 2} & -\hat{Y}_{\zeta 1} & 0 \end{bmatrix} \begin{Bmatrix} \Delta\theta_1 \\ \Delta\theta_2 \\ \Delta\theta_3 \end{Bmatrix} = h_\zeta \Delta\theta \tag{5.13b}$$

Equations (5.13) are used to transform the incremental fiber tip displacements to rotational increments in the equations of motion. The second-order accurate rotational update formulation due to Hughes and Winget [1980] is used to update the fiber vectors:

$$\hat{Y}_{\eta i}^{n+1} = R_{ij}\left(\Delta\theta\right)\hat{Y}_{\eta i}^{n} \tag{5.14a}$$

$$\hat{Y}_{\zeta i}^{n+1} = R_{ij}\left(\Delta\theta\right)\hat{Y}_{\zeta i}^{n} \tag{5.14b}$$

then

$$\Delta\hat{U}_{\eta a} = \hat{Y}_{\eta a}^{n+1} - \hat{Y}_{\eta a}^{n} \tag{5.15a}$$

$$\Delta\hat{U}_{\zeta a} = \hat{Y}_{\zeta a}^{n+1} - \hat{Y}_{\zeta a}^{n} \tag{5.15b}$$

where

$$R_{ij}\left(\Delta\theta\right) = \delta_{ij} + \frac{\left(2\delta_{ij} + \Delta S_{ik}\right)\Delta S_{ik}}{2D} \tag{5.16a}$$

$$\Delta S_{ij} = e_{ikj}\,\Delta\theta_k \tag{5.16b}$$

$$2D = 2 + \frac{1}{2}\left(\Delta\theta_1^2 + \Delta\theta_2^2 + \Delta\theta_3^2\right) \tag{5.16c}$$

Here δ_{ij} is the Kronecker delta and e_{ikj} is the permutation tensor.

5.2.1 Local Coordinate System

In addition to the above described fiber coordinate system, a local coordinate system is needed to enforce the zero normal stress conditions transverse to the axis. The orthonormal basis with two directions $\hat{e}_2$ and $\hat{e}_3$ normal to the axis of the beam is constructed as follows:

$$\hat{e}_1 = \frac{\overline{y}_2 - \overline{y}_1}{\left\|\overline{y}_2 - \overline{y}_1\right\|} \tag{5.17}$$

$$e_2' = \frac{\hat{Y}_{\eta 1} + \hat{Y}_{\eta 2}}{\left\|\hat{Y}_{\eta 1} + \hat{Y}_{\eta 2}\right\|} \tag{5.18}$$

From the vector cross product of these local tangents.

$$\hat{e}_3 = \hat{e}_1 \times e_2' \tag{5.19}$$

and to complete this orthonormal basis, the vector

$$\hat{e}_2 = \hat{e}_3 \times \hat{e}_1 \tag{5.20}$$

is defined. This coordinate system rigidly rotates with the deformations of the element.

The transformation of vectors from the global to the local coordinate system can now be defined in terms of the basis vectors as

$$\hat{A} = \begin{Bmatrix} \hat{A}_x \\ \hat{A}_y \\ \hat{A}_z \end{Bmatrix} = \begin{bmatrix} e_{1x} & e_{2x} & e_{3x} \\ e_{1y} & e_{2y} & e_{3y} \\ e_{1z} & e_{2z} & e_{3z} \end{bmatrix}^T \begin{Bmatrix} A_x \\ A_y \\ A_z \end{Bmatrix} = [q]\{A\} \tag{5.21}$$

where e_{ix}, e_{iy}, e_{iz} are the global components of the local coordinate unit vectors, $\hat{A}$ is a vector in the local coordinates, and $\boldsymbol{A}$ is the same vector in the global coordinate system.

5.3 Strains and Stress Update

5.3.1 Incremental Strain and Spin Tensors

The strain and spin increments are calculated from the incremental displacement gradient

$$G_{ij} = \frac{\partial \Delta u_i}{\partial y_j} \tag{5.22}$$

where Δu_i are the incremental displacements and y_j are the deformed coordinates. The incremental strain and spin tensors are defined as the symmetric and skew-symmetric parts, respectively, of G_{ij} :

$$\Delta \varepsilon_{ij} = \frac{1}{2}\left(G_{ij} + G_{ji}\right) \tag{5.23}$$

$$\Delta \omega_{ij} = \frac{1}{2}\left(G_{ij} - G_{ji}\right) \tag{5.24}$$

The incremental spin tensor $\Delta \omega_{ij}$ is used as an approximation to the rotational contribution of the Jaumann rate of the stress tensor; in an implicit implementation [Hallquist 1981b] the more accurate Hughes-Winget [1980] transformation matrix is used, Equation (5.16), with the incremental spin tensor for the rotational update. Here the Jaumann rate update is approximated as

$$\underline{\sigma}_{ij} = \sigma_{ij}^n + \sigma_{ip}^n \Delta \omega_{pj} + \sigma_{jp}^n \Delta \omega_{pi} \tag{5.25}$$

where the superscripts on the stress tensor refer to the updated $(n+1)$ and reference (n) configurations. This update of the stress tensor is applied before the constitutive evaluation, and the stress and strain are stored in the global coordinate system.

5.3.2 Stress Update

To evaluate the constitutive relation, the stresses and strain increments are rotated from the global to the local coordinate system using the transformation defined previously in Equation (5.21), viz.

$$\sigma_{ij}^{l^n} = q_{ik} \underline{\sigma}_{kn} q_{jn} \tag{5.26a}$$

$$\Delta \varepsilon_{ij}^{l} = q_{ik} \Delta \varepsilon_{kn} q_{jn} \tag{5.26b}$$

where the superscript l indicates components in the local coordinate system. The stress is updated incrementally:

$$\sigma_{ij}^{l^{n+1}} = \sigma_{ij}^{l^{n}} + \Delta\sigma_{ij}^{l^{n+\frac{1}{2}}} \tag{5.27}$$

and rotated back to the global system:

$$\sigma_{ij}^{n+1} = q_{ki}\sigma_{kn}^{l^{n+1}} q_{nj} \tag{5.28}$$

before computing the internal force vector.

5.3.3 Incremental Strain-Displacement Relations

After the constitutive evaluation is completed, the fully updated stresses are rotated back to the global coordinate system. These global stresses are then used to update the internal force vector

$$f_a^{\text{int}} = \int B_a^T \sigma d\upsilon \tag{5.29}$$

where f_a^{int} are the internal forces at node a and B_a is the strain-displacement matrix in the global coordinate system associated with the displacements at node a. The B matrix relates six global strain components to eighteen incremental displacements [three translational displacements per node and the six incremental fiber tip displacements of Equation (5.15)]. It is convenient to partition the B matrix:

$$B = \left[B_1, B_2\right] \tag{5.30}$$

Each B_a sub matrix is further partitioned into a portion due to strain and spin with the following sub matrix definitions:

$$B_a = \begin{bmatrix} B_1 & 0 & 0 & B_4 & 0 & 0 & B_7 & 0 & 0 \\ 0 & B_2 & 0 & 0 & B_5 & 0 & 0 & B_8 & 0 \\ 0 & 0 & B_3 & 0 & 0 & B_6 & 0 & 0 & B_9 \\ B_2 & B_1 & 0 & B_5 & B_4 & 0 & B_8 & B_7 & 0 \\ 0 & B_3 & B_2 & 0 & B_6 & B_5 & 0 & B_9 & B_8 \\ B_3 & 0 & B_1 & B_6 & 0 & B_4 & B_9 & 0 & B_7 \end{bmatrix} \tag{5.31}$$

where

$$B_i = \begin{cases} N_{a,i} = \dfrac{\partial N_a}{\partial y_i} & \text{for } i = 1,2,3 \\ \left(N_a z_{\eta a}\right)_{,i-3} = \dfrac{\partial \left(N_a z_{\eta a}\right)}{\partial y_{i-3}} & \text{for } i = 4,5,6 \\ \left(N_a z_{\zeta a}\right)_{,i-6} = \dfrac{\partial \left(N_a z_{\zeta a}\right)}{\partial y_{i-6}} & \text{for } i = 7,8,9 \end{cases} \tag{5.32}$$

With respect to the strain-displacement relations, note that:

- the derivative of the shape functions are taken with respect to the global coordinates;
- the B matrix is computed on the cross-section located at the mid-point of the axis;
- the resulting B matrix is a 6×18 matrix.

The internal force, f, given by

$$f' = T^t f_a^{\text{int}} \tag{5.33}$$

is assembled into the global right hand side internal force vector. T is defined as (also see Equation (5.13)):

$$T = \begin{bmatrix} I & 0 \\ 0 & h_\eta \\ 0 & h_\zeta \end{bmatrix} \tag{5.34}$$

where I is a 3×3 identity matrix.

5.3.4 Spatial Integration

The integration of Equation (5.29) for the beam element is performed with one- point integration along the axis and multiple points in the cross section. For rectangular cross sections, a variety of choices are available as is shown in Figure 5.3. The beam has no zero energy or locking modes.

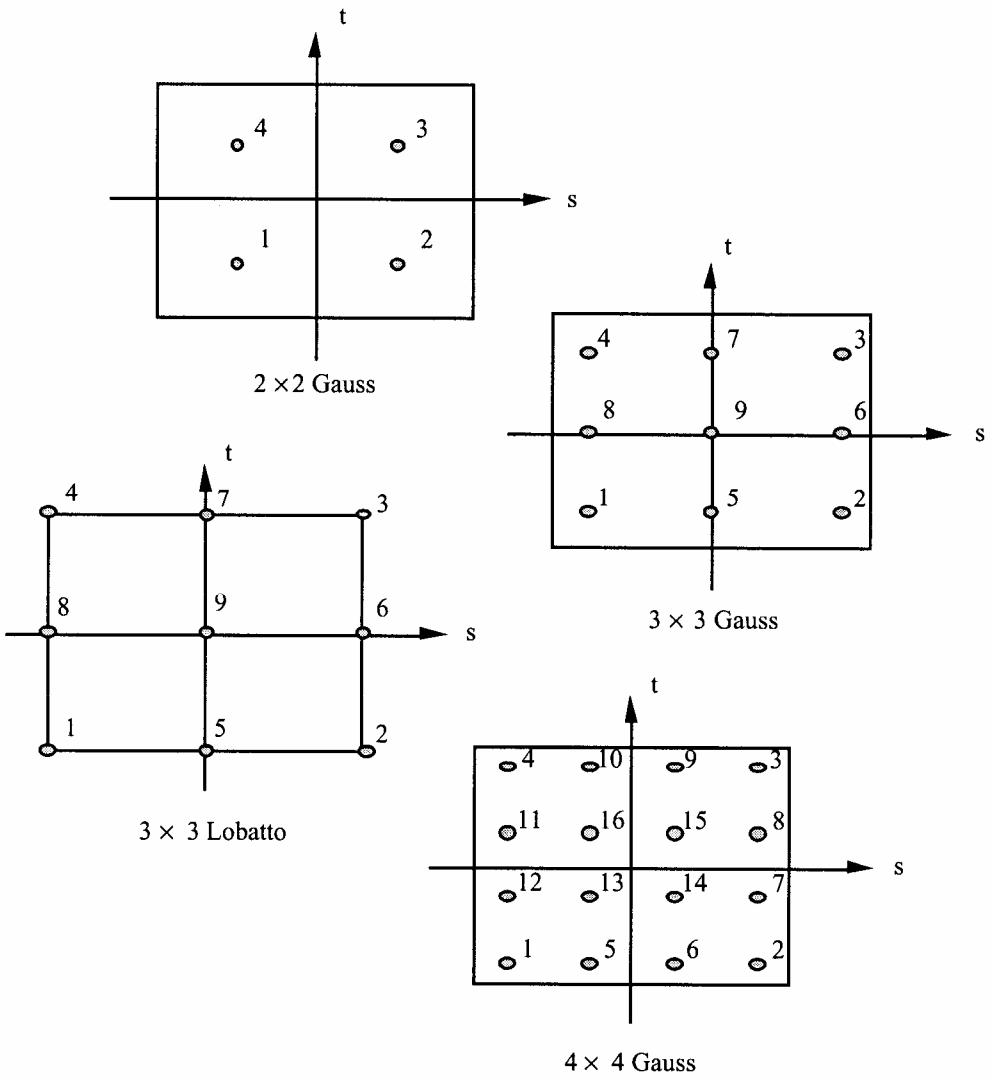


Figure 5.3. Integration possibilities for rectangular cross sections in the Hughes-Liu beam element.

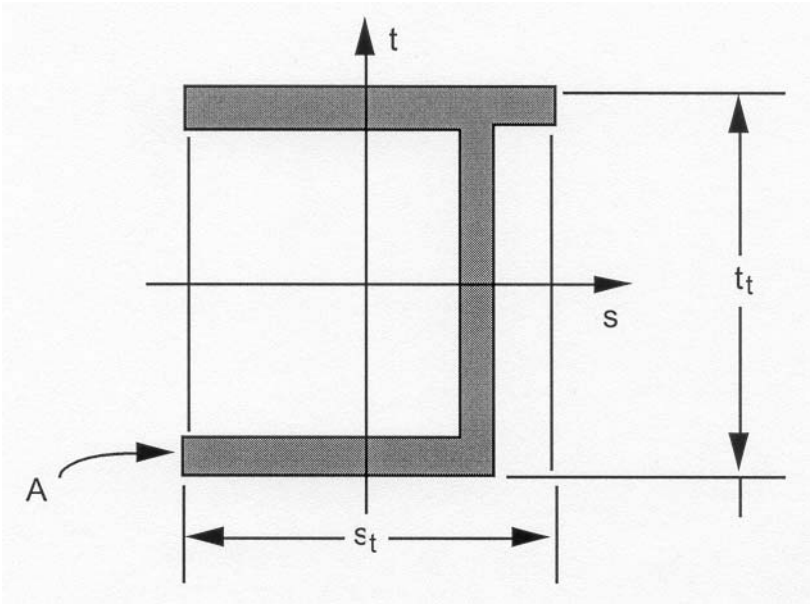


Figure 5.4. Specification of the nodal thickness, s_t and t_t, for a beam with an arbitrary cross-section.

For the user defined rule, it is necessary to specify the number of integration points and the relative area for the total cross section:

$$A_r = \frac{A}{s_t \cdot t_t}$$

where s_t and t_t are the beam thickness specified on either the cross section or beam element cards. The rectangular cross-section which contains s_t and t_t should completely contain the cross-sectional geometry. Figure 5.4 illustrates this for a typical cross-section. In Figure 5.5, the area is broken into twelve integration points. For each integration point, it is necessary to define the s and t parametric coordinates, (s_i, t_i), of the centroid of the ith integration point and the relative area associated with the point

$$A_{ri} = \frac{A_i}{A}$$

where A_i is the ratio of the area of the integration point and the actual area of the cross-section, A.

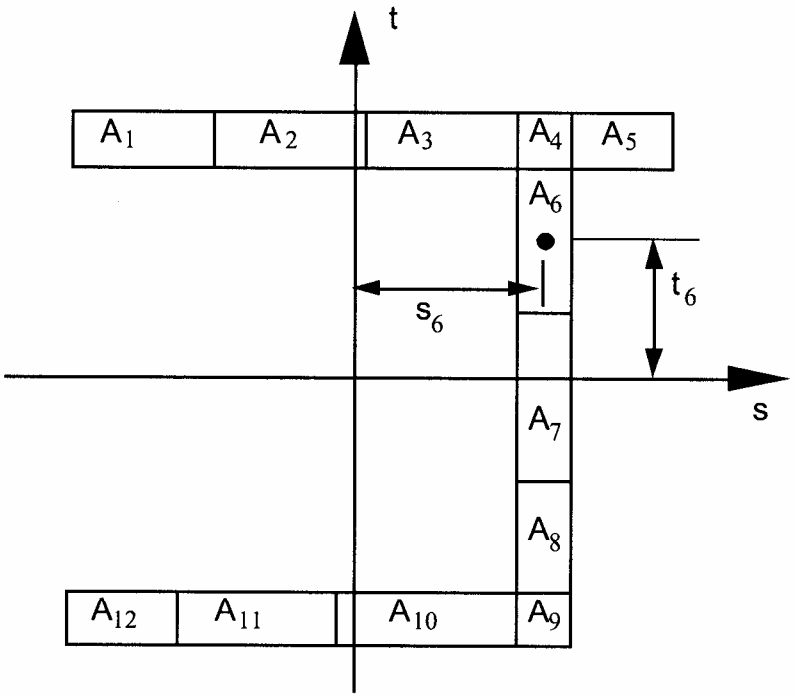


Figure 5.5. A breakdown of the cross section geometry in Figure 5.4 into twelve integration points.

6. WARPED BEAM ELEMENTS IN LS-DYNA

6.1 Resultant Warped Beam

6.1.1 Green-Lagrange Strains in Terms of Deformational Displacements

All quantities in this section are referred to the local element coordinate system $e_i,\ i=1,2,3$. The origin of the local system is taken at node 1, with e_1 directed along the line of centroids, while e_2, and e_3 are directed along the principal axes of the cross-section.

With respect to the local system, the Green-Lagrange strain tensor can be written as:

$$\varepsilon_{ij} = e_{ij} + \eta_{ij} \tag{6.1}$$

where,

$$\begin{aligned} e_{ij} &= 0.5\left(u_{i,j} + u_{j,i}\right) \\ \eta_{ij} &= 0.5 u_{k,i} u_{k,j} \end{aligned} \tag{6.2}$$

The geometric assumption of infinite in-plane rigidity implies $\varepsilon_{22} = \varepsilon_{33} = \gamma_{23} = 0$. Then the non-zero strain components which contribute to the strain energy are:

$$\begin{aligned} \varepsilon_{11} &= u_{1,1} + \tfrac{1}{2}\left(u_{1,1}^2 + u_{2,1}^2 + u_{3,1}^2\right) \\ 2\varepsilon_{12} &= u_{1,2} + u_{2,1} + u_{1,1,} u_{1,2} + u_{2,1} u_{2,2} + u_{3,1} u_{3,2} \\ 2\varepsilon_{13} &= u_{1,3} + u_{3,1} + u_{1,1} u_{1,3} + u_{2,1} u_{2,3} + u_{3,1} u_{3,3} \end{aligned} \tag{6.3}$$

6.1.2 Deformational Displacements After Large Rotations

The position vectors of an arbitrary point *P* in the initial and current local configurations are:

$$x_P^0 = x_C^0 + \begin{pmatrix} e_1 & e_2 & e_3 \end{pmatrix} \begin{pmatrix} 0 \\ x_2 \\ x_3 \end{pmatrix} \tag{6.4}$$

$$x_P = x_C + \begin{pmatrix} e_1' & e_2' & e_3' \end{pmatrix} \begin{pmatrix} \varpi\phi \\ x_2 \\ x_3 \end{pmatrix} \tag{6.5}$$

respectively, with

$$\begin{pmatrix} e_1' & e_2' & e_3' \end{pmatrix} = \left(I + \theta + \tfrac{1}{2}\theta^2\right) \begin{pmatrix} e_1 & e_2 & e_3 \end{pmatrix} \tag{6.6}$$

where

$$\theta = \begin{bmatrix} 0 & -\theta_3 & \theta_2 \\ \theta_3 & 0 & -\theta_1 \\ -\theta_2 & \theta_1 & 0 \end{bmatrix} \tag{6.7}$$

and ϖ is the Saint-Venant warping function about the centroid *C*. By the transfer theorem, the following relation holds:

$$\varpi = \omega + c_2 x_3 - c_3 x_2 \tag{6.8}$$

where ω refers to the shear center *S*, and c_2 and c_3 are the coordinates of *S*.

Subtracting Equation (6.4) from Equation (6.5) and neglecting third-order terms, the displacements vector of point *P* can be computed:

$$\begin{aligned} u_1 &= \bar{u}_1 - x_2\theta_3 + x_3\theta_2 + \tfrac{1}{2}x_2\theta_1\theta_2 + \tfrac{1}{2}x_3\theta_1\theta_3 + \varpi\phi \\ u_2 &= \bar{u}_2 - x_3\theta_1 - \tfrac{1}{2}x_2\left(\theta_1^2 + \theta_3^2\right) + \tfrac{1}{2}x_3\theta_2\theta_3 + \varpi\theta_3\phi \\ u_3 &= \bar{u}_3 + x_2\theta_1 - \tfrac{1}{2}x_3\left(\theta_1^2 + \theta_2^2\right) + \tfrac{1}{2}x_2\theta_2\theta_3 - \varpi\theta_2\phi \end{aligned} \tag{6.9}$$

where $\bar{u}_1$, $\bar{u}_2$, and $\bar{u}_3$ are the displacements of the centroid *C*.

6.1.3 Green-Lagrange Strains in terms of Centroidal Displacements and Angular Rotations

From Equations (6.3) and (6.9), a second-order approximation of the Green-Lagrange strains can be evaluated. Neglecting term $\frac{1}{2}u_{1,1}^2$ and the nonlinear strain components generated by warping, the strain components are simplified as

$$\begin{aligned} \varepsilon_{11} &= \varepsilon_0 + x_2\kappa_2 + x_3\kappa_3 + \tfrac{1}{2}\left(x_2^2 + x_3^2\right)\theta_{1,1}^2 + \omega\phi_{,1} \\ 2\varepsilon_{12} &= \gamma_{12} + \varpi_{,2}\phi - x_3\kappa_1 \\ 2\varepsilon_{13} &= \gamma_{13} + \varpi_{,3}\phi + x_2\kappa_1 \end{aligned} \tag{6.10}$$

with

$$\begin{aligned} \varepsilon_0 &= \bar{u}_{1,1} + \tfrac{1}{2}\left(\bar{u}_{2,1}^2 + \bar{u}_{3,1}^2\right) \\ \kappa_1 &= \theta_{1,1} + \tfrac{1}{2}\left(\theta_{2,1}\theta_3 - \theta_{3,1}\theta_2\right) \\ \kappa_2 &= -\theta_{3,1} + \tfrac{1}{2}\left(\theta_1\theta_{2,1} + \theta_{1,1}\theta_2\right) + \bar{u}_{3,1}\theta_{1,1} - c_3\phi_{,1} \\ \kappa_3 &= \theta_{2,1} + \tfrac{1}{2}\left(\theta_1\theta_{3,1} + \theta_{1,1}\theta_3\right) - \bar{u}_{2,1}\theta_{1,1} + c_2\phi_{,1} \\ \gamma_{12} &= \bar{u}_{2,1} - \theta_3 + \tfrac{1}{2}\theta_1\theta_2 + \bar{u}_{3,1}\theta_1 - \bar{u}_{1,1}\theta_3 \\ \gamma_{13} &= \bar{u}_{3,1} + \theta_2 + \tfrac{1}{2}\theta_1\theta_3 - \bar{u}_{2,1}\theta_1 + \bar{u}_{1,1}\theta_2 \end{aligned} \tag{6.11}$$

Numerical testing has shown that neglecting the nonlinear terms in the curvatures κ_1, κ_2, κ_3 and bending shear strains γ_{12}, γ_{13} has little effect on the accuracy of the results. Therefore, Equation (6.11) can be simplified to

$$\begin{aligned} &\kappa_1 = \theta_{1,1}, \quad \kappa_2 = -\theta_{3,1} - c_3\phi_{,1}, \quad \kappa_3 = \theta_{2,1} + c_2\phi_{,1} \\ &\varepsilon_0 = \bar{u}_{1,1} + \tfrac{1}{2}\left(\bar{u}_{2,1}^2 + \bar{u}_{3,1}^2\right), \quad \gamma_{12} = \bar{u}_{2,1} - \theta_3, \quad \gamma_{13} = \bar{u}_{3,1} + \theta_2 \end{aligned} \tag{6.12}$$

Adopting Bernoulli's assumption $\left(\gamma_{12} = \gamma_{13} = 0\right)$ and Vlasov's assumption $\left(\phi = \theta_{1,1}\right)$, Equation (6.10) can be rewritten as:

$$\begin{aligned} \varepsilon_{11} &= \varepsilon_0 + x_2\kappa_2 + x_3\kappa_3 + \tfrac{1}{2}r^2\kappa_1^2 + \omega\theta_{1,11} \\ 2\varepsilon_{12} &= \left(\varpi_{,2} - x_3\right)\kappa_1 \\ 2\varepsilon_{13} &= \left(\varpi_{,3} - x_2\right)\kappa_1 \end{aligned} \tag{6.13}$$

where

$$r^2 = x_2^2 + x_3^2, \quad \kappa_1 = \theta_{1,1}, \quad \kappa_2 = -\bar{u}_{2,11} - c_3\theta_{1,11}, \quad \kappa_3 = -\bar{u}_{3,11} + c_2\theta_{1,11} \tag{6.14}$$

To avoid membrane locking, ε_{11} in Equation (6.13) is reformulated as

$$\varepsilon_{11} = \varepsilon_a + x_2\kappa_2 + x_3\kappa_3 + \tfrac{1}{2}\left(r^2 - \tfrac{I_o}{A}\right)\kappa_1^2 + \omega\theta_{1,11} \tag{6.15}$$

where

$$\varepsilon_a = \frac{1}{l}\int_0^1 \left[\bar{u}_{1,1} + \tfrac{1}{2}\left(\bar{u}_{2,1}^2 + \bar{u}_{3,1}^2 + \tfrac{I_o}{A}\kappa_1^2\right)\right] dx_1 \tag{6.16}$$

6.1.4 Strain Energy

Assuming material is linear elastic, the strain energy can be evaluated from:

$$U = \int_0^L \left(\frac{1}{2}E\int_A \varepsilon_{11}^2 dA + \frac{1}{2}G\int_A \left[\left(2\varepsilon_{12}\right)^2 + \left(2\varepsilon_{13}\right)^2\right] dA\right) dx_1 \tag{6.17}$$

The following relations are used in integrating the previous equations:
(1) Since the reference frame is located at centroid *C* with e_2 and e_3 directed along the principal axes,

$$\int_A x_2 dA = 0, \quad \int_A x_3 dA = 0, \quad \int_A x_2 x_3 dA = 0 \tag{6.18}$$

(2) Since sectorial area ω refers to shear center *S*,

$$\int_A \omega dA = 0, \quad \int_A x_2 \omega dA = 0, \quad \int_A x_3 \omega dA = 0 \tag{6.19}$$

Integration through the cross-section gives:

$$\int_A \varepsilon_{11}^2 dA + A\varepsilon_a^2 + I_{22}\kappa_2^2 + I_{33}\kappa_3^2 + I_\omega \theta_{1,11}^2 + I_{2r}\kappa_2\kappa_1^2 + I_{3r}\kappa_3\kappa_1^2 + I_{\omega r}\theta_{1,11}\kappa_1^2 + \tfrac{1}{4}\left(I_{rr} - \tfrac{I_O^2}{A}\right)\kappa_1^4 \tag{6.20}$$

$$\int_A \left[\left(2\varepsilon_{12}\right)^2 + \left(2\varepsilon_{13}\right)^2 \right] dA = J\kappa_1^2 \tag{6.21}$$

with

$$\begin{aligned} &I_{22} = \int_A x_2^2 dA, \qquad I_{33} = \int_A x_3^2 dA, \qquad I_O = I_{22} + I_{33} \\ &I_{2r} = \int_A x_2 r^2 dA, \quad I_{3r} = \int_A x_3 r^2 dA, \quad I_{rr} = \int_A r^4 dA \\ &I_\omega = \int_A \omega^2 dA, \qquad I_{\omega r} = \int_A \omega r^2 dA, \quad J = \int_A \left[\left(\bar{\omega}_{,3} + x_2\right)^2 + \left(\bar{\omega}_{,2} - x_3\right)^2 \right] dA \end{aligned} \tag{6.22}$$

6.1.5 Displacement Field

Linear interpolation is used for axial displacement $\bar{u}$, whereas Hermitian interpolations are used for u_2, u_3, and θ_1, considering the following relations used in deriving the final expression of strain energy:

$$\theta_2 = -u_{3,1} \qquad \theta_3 = u_{2,1} \qquad \phi = \theta_{1,1} \tag{6.23}$$

The nodal displacement field is constructed by

$$\bar{u}_1 = \boldsymbol{N}_1 \boldsymbol{d} \tag{6.24}$$

$$\begin{Bmatrix} \bar{u}_2 \\ \bar{u}_3 \\ \theta_1 \end{Bmatrix} = \boldsymbol{N}_2 \boldsymbol{d} \tag{6.25}$$

where

$$\boldsymbol{d}^T = \begin{bmatrix} 0 & 0 & 0 & 0 & \theta_{2I} & \theta_{3I} & \phi_I & \bar{u}_{1JI} & 0 & 0 & \theta_{1JI} & \theta_{2J} & \theta_{3J} & \phi_J \end{bmatrix}^T \tag{6.26}$$

$$\boldsymbol{N}_1 = \left[1-\xi \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \middle| \quad \xi \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot\right]$$
$$\boldsymbol{N}_2 = \left[\begin{array}{ccccccc|ccccccc} \cdot & f & \cdot & \cdot & \cdot & g & \cdot & \cdot & 1-f & \cdot & \cdot & \cdot & h & \cdot \\ \cdot & \cdot & f & \cdot & -g & \cdot & \cdot & \cdot & \cdot & 1-f & \cdot & -h & \cdot & \cdot \\ \cdot & \cdot & \cdot & f & \cdot & \cdot & g & \cdot & \cdot & \cdot & 1-f & \cdot & \cdot & h \end{array}\right] \tag{6.27}$$

with

$$f = 1-3\xi^2+2\xi^3 \qquad g = l\left(\xi-2\xi^2+\xi^3\right) \qquad h = l\left(\xi^3-\xi^2\right) \tag{6.28}$$

Equations (6.25) and (6.27) also imply

$$\theta_{,1} = \boldsymbol{N}_3\boldsymbol{d} \tag{6.29}$$

where

$$\boldsymbol{N}_3 = \left[\begin{array}{ccccccc|ccccccc} \cdot & \cdot & f_{,1} & \cdot & \cdot & g_{,1} & \cdot & \cdot & \cdot & -f_{,1} & \cdot & \cdot & h_{,1} \end{array}\right] \tag{6.30}$$

6.1.6 Strain Energy in Matrix Form

The strain energy due to the average strain ε_a defined in Equation (6.16) can be expressed in matrix form as

$$U_1 = \frac{1}{2}EAl\left[\int_0^1\left(\boldsymbol{N}_{1,1}\boldsymbol{d}\right)d\xi + \frac{1}{2}\boldsymbol{d}^T\int_0^1\left(\boldsymbol{N}_{2,1}^T\,\boldsymbol{D}\boldsymbol{N}_{2,1}\right)d\xi\boldsymbol{d}\right]^2 \tag{6.31}$$

where

$$\boldsymbol{D} = \text{diag}\left(1,\ 1,\ \frac{I_o}{A}\right) \tag{6.32}$$

The strain energy due to the second through fourth terms is

$$U_2 = \frac{1}{2}El\ \boldsymbol{d}^T\int_0^1\boldsymbol{N}_{2,11}^T\boldsymbol{H}\boldsymbol{N}_{2,11}d\xi\boldsymbol{d} \tag{6.33}$$

where

$$\boldsymbol{H} = \begin{bmatrix} I_{22} & & I_{22}c_3 \\ & I_{33} & -I_{33}c_2 \\ I_{22}c_3 & -I_{33}c_2 & I'_\omega \end{bmatrix},\quad I'_\omega = I_\omega + I_{22}c_3^2 + I_{33}c_2^2 \tag{6.34}$$

The strain energy due to the fifth through seventh terms is

$$U_3 = \frac{1}{2} El \int_0^1 \left(\boldsymbol{N}_3 \boldsymbol{d} \right)^2 \boldsymbol{\upsilon} \boldsymbol{N}_{2,11} d\xi \boldsymbol{d} \tag{6.35}$$

where

$$\upsilon = \begin{pmatrix} -I_{2r} & -I_{3r} & I'_{\omega r} \end{pmatrix}, \quad I'_{\omega r} = I_{\omega r} - c_3 I_{2r} + c_2 I_{3r} \tag{6.36}$$

The strain energy due to the eighth and ninth terms is

$$U_4 = \frac{1}{8} El \left(I_{rr} - \frac{I_o^2}{A} \right) \int_0^1 \left(\boldsymbol{N}_3 \boldsymbol{d} \right)^4 d\xi + \frac{1}{2} GJl \int_0^1 \left(\boldsymbol{N}_3 \boldsymbol{d} \right)^2 d\xi \tag{6.37}$$

6.1.7 Internal Nodal Force Vector

The internal force can be evaluated from

$$f_e = EAl \left(\frac{\overline{u}_{1JL}}{l} + \frac{1}{2} \boldsymbol{d}^T \boldsymbol{Q} \right) \left(\boldsymbol{P} + \boldsymbol{Q} \right) + E \left(\boldsymbol{R} + \frac{1}{2} \boldsymbol{S} + \boldsymbol{T} + \frac{1}{2} \boldsymbol{V} \right) + \boldsymbol{GW} \tag{6.38}$$

where

$$\begin{aligned} &\boldsymbol{P} = \int_0^1 \boldsymbol{N}_{1,1}^T d\xi \quad \boldsymbol{Q} = \int_0^1 \boldsymbol{N}_{2,1}^T \boldsymbol{D} \boldsymbol{N}_{2,1} d\xi \boldsymbol{d} \quad \boldsymbol{R} = l \int_0^1 \boldsymbol{N}_{2,11}^T \boldsymbol{H} \boldsymbol{N}_{2,11} d\xi \boldsymbol{d} \\ &\boldsymbol{S} = l \int_0^1 \left(\boldsymbol{N}_3 \boldsymbol{d} \right)^2 \boldsymbol{N}_{2,11}^T d\xi \boldsymbol{\upsilon}^T \quad \boldsymbol{T} = l \int_0^1 \left(\boldsymbol{N}_3 \boldsymbol{d} \right) \left(\boldsymbol{\upsilon} \boldsymbol{N}_{2,11} \boldsymbol{d} \right) \boldsymbol{N}_3^T d\xi \\ &\boldsymbol{V} = \left(I_{rr} - \frac{I_o^2}{A} \right) l \int_0^1 \left(\boldsymbol{N}_3 \boldsymbol{d} \right)^3 \boldsymbol{N}_3^T d\xi \quad \boldsymbol{W} = Jl \int_0^1 \left(\boldsymbol{N}_3 \boldsymbol{d} \right) \boldsymbol{N}_3^T d\xi \quad \lambda = \frac{\overline{u}_{1JI}}{l} + \frac{1}{2} \boldsymbol{d}^T \boldsymbol{Q} \end{aligned} \tag{6.39}$$

With respect to the local coordinate system, there are totally eight independent components in the nodal force vector, in correspondence to the eight nodal displacement components.

Other forces can be calculated by:

$$F_1 = -F_8 \quad F_2 \frac{F_6 + F_{13}}{l} \quad F_3 = -\frac{F_5 + F_{12}}{l} \quad F_4 = -F_{11} \quad F_9 = -F_2 \quad F_{10} = -F_3 \tag{6.41}$$

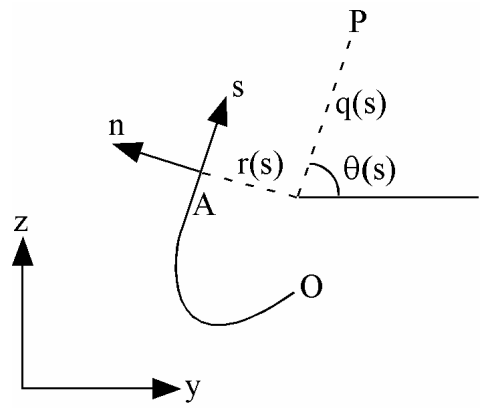


Figure 6.1. Definition of coordinates in thin-walled open section

6.2 Integrated Warped Beam

6.2.1 Kinematics

We introduce three coordinate systems that are mutually interrelated. The first coordinate system is the orthogonal Cartesian coordinate system (x, y, z), for which the y and z axes lie in the plane of the cross-section and the x-axis parallel to the longitudinal axis of the beam. The second coordinate system is the local plate coordinate system (x, s, n) as shown in Figure 6.1, wherein the n-axis is normal to the middle surface of a plate element, the s-axis is tangent to the middle surface and is directed along the contour line of the cross-section. The (x, s, n) and (x, y, z) coordinate systems are related through an angle of orientation θ as defined in Figure 40.1. The third coordinate set is the contour coordinate s along the profile of the section with its origin at some point O on the profile section. Point P is called the pole through which the axis parallel to the x-axis is called the pole axis. To derive the analytical model for a thin-walled beam, the following two assumptions are made:

1. The contour of the thin wall does not deform in its own plane.
2. The shear strain γ_{sx} of the middle surface is zero.

According to assumption 1, the midsurface displacement components v and w with respect to the (x, s, n) coordinate system at a point A can be expressed in terms of displacements V and W of the pole P in the (x, y, z) coordinate system and the rotation angle ϕ_x about the pole axis

$$\begin{aligned} v(x,s) &= V(x)\cos\theta(s) + W(x)\sin\theta(s) - r(s)\phi_x(x) \\ w(x,s) &= -V(x)\sin\theta(s) + W(x)\cos\theta(s) - q(s)\phi_x(x) \end{aligned} \tag{6.55}$$

These equations apply to the whole contour. The out-of-plane displacement u can now be found from assumption 2. On the middle surface

$$\frac{\partial u}{\partial s}+\frac{\partial v}{\partial x}=0 \tag{6.56}$$

which can be written

$$\frac{\partial u}{\partial s}=-\frac{\partial v}{\partial x}=-V'(x)\cos\theta(s)-W'(x)\sin\theta(s)+r(s)\phi'_x(x) \tag{6.57}$$

Integrating this relation from point O to an arbitrary point on the contour yields (using t as a dummy for s)

$$\int_0^s \frac{\partial u}{\partial t}dt=-V'(x)\int_0^s \cos\theta(t)dt-W'(x)\int_0^s \sin\theta(t)dt+\phi'_x(x)\int_0^s r(t)dt\,. \tag{6.58}$$

Noting that

$$\begin{aligned} dy&=\cos\theta(t)dt\\ dz&=\sin\theta(t)dt \end{aligned} \tag{6.59}$$

we end up with

$$\begin{aligned} u(x,s)&=\underbrace{u(x,0)+V'(x)y(0)+W'(x)z(0)+\phi'_x(x)\varpi}_{=:U(x)}\\ &\quad-\underbrace{V'(x)}_{=:\phi_z(x)}y(s)-\underbrace{W'(x)}_{=:-\phi_y(x)}z(s)+\phi'_x(x)(\omega(s)-\varpi)\\ &=U(x)-\phi_z(x)y+\phi_y(x)z+\phi'_x(x)(\omega(s)-\varpi) \end{aligned} \tag{6.60}$$

where U denotes the average out-of-plane displacement over the section, ϕ_y and ϕ_z denote the rotation angle about the y and z axis[1], respectively, ω is the sectorial area defined as

$$\omega(s)=\int_0^s r(t)dt \tag{6.61}$$

and ϖ is the average of the sectorial area over the section.

The expression for the displacements in the (x,y,z) coordinate system is

[1] The substitution of $V'(x)$ for $\phi_z(x)$ and $W'(x)$ for $-\phi_y(x)$ can be seen as a conversion from an Euler-Bernoulli kinematic assumption to that of Timoschenko.

$$
\begin{aligned}
u(x,y,z) &= U(x) - \phi_z(x)y + \phi_y(x)z + \vartheta(x)\omega(y,z) \\
v(x,y,z) &= V(x) - \phi_x(x)z \\
w(x,y,z) &= W(x) + \phi_x(x)y
\end{aligned}
\tag{6.62}
$$

where we have introduced ϑ to represent the twist constrained by the condition

$$
\vartheta(x) = \phi_x(x) \tag{6.63}
$$

and ω denotes the sectorial coordinate that is adjusted for zero average over the section.

6.2.2 Kinetics

The kinetic energy of the beam can be written

$$
T = \frac{1}{2}\int_V \rho\left\{\dot{u}^2 + \dot{v}^2 + \dot{w}^2\right\}dV \tag{6.64}
$$

Taking the variation of this expression leads to

$$
\begin{aligned}
\delta T &= \int_V \rho\left\{\dot{u}\delta\dot{u} + \dot{v}\delta\dot{v} + \dot{w}\delta\dot{w}\right\}dV \\
&= \int_V \rho\left\{\dot{U} - \dot{\phi}_z y + \dot{\phi}_y z + \dot{\vartheta}\omega\right\}\left\{\delta\dot{U} - \delta\dot{\phi}_z y + \delta\dot{\phi}_y z + \delta\dot{\vartheta}\omega\right\}dV + \\
&\quad \int_V \rho\left\{\dot{V} - \dot{\phi}_x z\right\}\left\{\delta\dot{V} - \delta\dot{\phi}_x z\right\}dV + \int_V \rho\left\{\dot{W} + \dot{\phi}_x y\right\}\left\{\delta\dot{W} + \delta\dot{\phi}_x y\right\}dV \\
&= \int_V \rho\left\{\dot{U}\delta\dot{U} + y^2\dot{\phi}_z\delta\dot{\phi}_z - y\omega\dot{\phi}_z\delta\dot{\vartheta} + z^2\dot{\phi}_y\delta\dot{\phi}_y\right\}dV + \\
&\quad \int_V \rho\left\{z\omega\dot{\phi}_y\delta\dot{\vartheta} - y\omega\dot{\vartheta}\delta\dot{\phi}_z + z\omega\dot{\vartheta}\delta\dot{\phi}_y + \omega^2\dot{\vartheta}\delta\dot{\vartheta}\right\}dV + \\
&\quad \int_V \rho\left\{\dot{V}\delta\dot{V} + z^2\dot{\phi}_x\delta\dot{\phi}_x + \dot{W}\delta\dot{W} + y^2\dot{\phi}_x\delta\dot{\phi}_x\right\}dV \\
&= \rho A\int_l \left\{\dot{U}\delta\dot{U} + \dot{V}\delta\dot{V} + \dot{W}\delta\dot{W}\right\}dV + \rho I_{zz}\int_l \left\{\dot{\phi}_z\delta\dot{\phi}_z + \dot{\phi}_x\delta\dot{\phi}_x\right\}dV + \rho I_{\acute{y}y}\int_l \left\{\dot{\phi}_y\delta\dot{\phi}_y + \dot{\phi}_x\delta\dot{\phi}_x\right\}dV + \\
&\quad \rho I_{y\omega}\int_l \left\{\dot{\phi}_y\delta\dot{\vartheta} + \dot{\vartheta}\delta\dot{\phi}_y\right\}dV - \rho I_{z\omega}\int_l \left\{\dot{\vartheta}\delta\dot{\phi}_z + \dot{\phi}_z\delta\dot{\vartheta}\right\}dV + \rho I_{\omega\omega}\int_l \dot{\vartheta}\delta\dot{\vartheta}dV
\end{aligned}
$$

from which the consistent mass matrix can be read out. Here A is the cross sectional area, I_{zz} and I_{yy} are the second moments of area with respect to the z and y axes, respectively, $I_{\omega\omega}$ is the sectorial second moment and $I_{z\omega}$ and $I_{y\omega}$ are the sectorial product moments. An approximation of this mass matrix can be made by neglecting the off diagonal components. The diagonal components are

$$
\begin{aligned}
m_{TRNS} &= \rho A l/3 \\
m_{RTx} &= \rho(I_{yy}+I_{zz})l/3 \\
m_{RTy} &= \rho I_{yy} l/3 \\
m_{RTz} &= \rho I_{zz} l/3 \\
m_{TWST} &= \rho I_{\omega\omega} l/3
\end{aligned}
\tag{6.65}
$$

With E as Young's modulus and G as the shear modulus, the strain energy can be written

$$
\Pi = \frac{1}{2}(E\varepsilon_{xx}^2 + G\gamma_{xy}^2 + G\gamma_{xz}^2) \tag{6.66}
$$

where the infinitesimal strain components are (neglecting the derivatives of sectorial area)

$$
\begin{aligned}
\varepsilon_{xx} &= U' + \phi_y' z - \phi_z' y + \vartheta'\omega \\
\gamma_{xy} &= V' - \phi_x' z - \phi_z \\
\gamma_{xz} &= W' + \phi_x' y + \phi_y
\end{aligned}
\tag{6.67}
$$

and the variation of the same can be written

$$
\begin{aligned}
\delta\varepsilon_{xx} &= \delta U' + \delta\phi_y' z - \delta\phi_z' y + \delta\vartheta'\omega \\
\delta\gamma_{xy} &= \delta V' - \delta\phi_x' z - \delta\phi_z \\
\delta\gamma_{xz} &= \delta W' + \delta\phi_x' y + \delta\phi_y
\end{aligned}
\tag{6.68}
$$

The variation of the strain energy is

$$
\begin{aligned}
\delta\Pi &= \int_V \left\{ E\varepsilon_{xx}\delta\varepsilon_{xx} + G\gamma_{xy}\delta\gamma_{xy} + G\gamma_{xz}\delta\gamma_{xz} \right\} dV \\
&= EA\int_l U'\delta U' dl + GA\int_l V'\delta V' dl + GA\int_l W'\delta W' dl + G(I_{yy}+I_{zz})\int_l \phi_x'\delta\phi_x' dl + \\
&\quad EI_{yy}\int_l \phi_y'\delta\phi_y' dl + GA\int_l \phi_y\delta\phi_y dl + EI_{zz}\int_l \phi_z'\delta\phi_z' dl + GA\int_l \phi_z\delta\phi_z dl + EI_{\omega\omega}\int_l \vartheta'\delta\vartheta' dl - \\
&\quad GA\int_l (V'\delta\phi_z + \phi_z\delta V') dl + GA\int_l (W'\delta\phi_y + \phi_y\delta W') dl + \\
&\quad EI_{y\omega}\int_l (\phi_y'\delta\vartheta' + \vartheta'\delta\phi_y') dl - EI_{z\omega}\int_l (\phi_z'\delta\vartheta' + \vartheta'\delta\phi_z') dl
\end{aligned}
$$

where the stiffness matrix can be read. Again the diagonal components are

$$
\begin{aligned}
k_{TRNS} &= EA/l \\
k_{SHR} &= GA/l \\
k_{RTx} &= G(I_{yy}+I_{zz})/l \\
k_{RTy} &= EI_{yy}/l + GAl/3 \\
k_{RTz} &= EI_{zz}/l + GAl/3 \\
k_{TWST} &= EI_{\omega\omega}/l
\end{aligned}
\tag{6.69}
$$

From the expressions of the mass and stiffness matrix, the frequencies of the most common modes can be estimated. These are

1. The tensile and twisting modes with frequency $\omega = \frac{\sqrt{3}}{l}\sqrt{\frac{E}{\rho}}$.
2. The transverse shear and torsional mode with frequency $\omega = \frac{\sqrt{3}}{l}\sqrt{\frac{G}{\rho}}$
3. The bending modes with frequencies $\omega = \sqrt{\frac{3E}{\rho l^2}+\frac{GA}{\rho I_{yy}}}$ and $\omega = \sqrt{\frac{3E}{\rho l^2}+\frac{GA}{\rho I_{zz}}}$.

Which one of these four that is the highest depends on the geometry of the beam element. In LS-DYNA the first of these frequencies is used for calculating a stable time step. We have found no reason for changing approach regarding this element.

6.2.3 Penalty on Twist

The twist is constrained using a penalty that is introduced in the strain energy as

$$
\Pi_P = \frac{PEA}{2}\int_l (\phi_x' - \vartheta)^2\, dl \tag{6.70}
$$

and the corresponding variation is

$$
\delta\Pi = PEA\int_l (\phi_x' - \vartheta)(\delta\phi_x' - \delta\vartheta)dl\ . \tag{6.71}
$$

The diagonal of the stiffness matrix is modified as follows

$$
\begin{aligned}
k_{RTx} &= G(I_{yy}+I_{zz})/l + PEA/l \\
k_{TWST} &= EI_{\omega\omega}/l + PEAl/3
\end{aligned}
\tag{6.72}
$$

This increases the twist mode frequency to $\sqrt{\frac{3E}{\rho l^2}+\frac{PEA}{\rho I_{\omega\omega}}}$ and the torsional mode to

$$\frac{\sqrt{3}}{l}\sqrt{\frac{G(I_{yy}+I_{zz})+PEA}{\rho}}\,. \tag{6.73}$$

Even though this gives an indication of a frequency increase we have made no modifications on the computation of the critical time step in an explicit analysis. We have used $P=1$ in the implementation. This decision may have to be reconsidered depending on the choice of the parameter μ, in the end it will come down to trial and error from numerical simulations.

6.3 Generalization to Large Displacements

A generalization of the small displacement theory to nonlinear theory is quite straightforward. We have used a corotational formulation where the small strains in the linear theory are used directly as strain rates in the element system. We emphasize that the nonlinear beam formulation is obtained by simply replacing displacements for velocities and strains with strain rates in the previous section.

The nodal velocities for a beam element in the local system is written

$$\mathbf{v}=\begin{pmatrix} v_x^1 & v_y^1 & v_z^1 & \omega_x^1 & \omega_y^1 & \omega_z^1 & \vartheta^1 & v_x^2 & v_y^2 & v_z^2 & \omega_x^2 & \omega_y^2 & \omega_z^2 & \vartheta^2 \end{pmatrix}^T \tag{6.74}$$

where the superscript refers to the local node number. These are obtained by transforming the translational velocities and rotational velocities using the local to global transformation matrix q_{ij}. The strain rate – velocity matrix in the local system can be written

$$\mathbf{B}_0=\begin{bmatrix} -l_0^{-1} & 0 & 0 & 0 & -l_0^{-1}z & l_0^{-1}y & -l_0^{-1}\omega & l_0^{-1} & 0 & 0 & 0 & l_0^{-1}z & -l_0^{-1}y & l_0^{-1}\omega \\ 0 & -l_0^{-1} & 0 & l_0^{-1}z & 0 & -\frac{1}{2} & 0 & 0 & l_0^{-1} & 0 & -l_0^{-1}z & 0 & -\frac{1}{2} & 0 \\ 0 & 0 & -l_0^{-1} & -l_0^{-1}y & \frac{1}{2} & 0 & 0 & 0 & 0 & l_0^{-1} & l_0^{-1}y & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & -l_0^{-1} & 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 & l_0^{-1} & 0 & 0 & -\frac{1}{2} \end{bmatrix}$$

where l_0 is the beam length in the reference configuration, i.e., beginning of the time step. A corresponding matrix w.r.t. the current configuration is

$$\mathbf{B}=\begin{bmatrix} -l^{-1} & 0 & 0 & 0 & -l^{-1}z & l^{-1}y & -l^{-1}\omega & l^{-1} & 0 & 0 & 0 & l^{-1}z & -l^{-1}y & l^{-1}\omega \\ 0 & -l^{-1} & 0 & l^{-1}z & 0 & -\frac{1}{2} & 0 & 0 & l^{-1} & 0 & -l^{-1}z & 0 & -\frac{1}{2} & 0 \\ 0 & 0 & -l^{-1} & -l^{-1}y & \frac{1}{2} & 0 & 0 & 0 & 0 & l^{-1} & l^{-1}y & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & -l^{-1} & 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 & l^{-1} & 0 & 0 & -\frac{1}{2} \end{bmatrix}$$

where we use the current length of the beam. These matrices are evaluated in each integration point (x,y) of the cross section. To compute the strain rate in the local system we simply apply

$$\dot{\boldsymbol{\varepsilon}} = \mathbf{B}_0 \mathbf{v} \tag{6.75}$$

which is then used to update the local stresses $\boldsymbol{\sigma}$. The internal force vector is then assembled as

$$\mathbf{f} = \mathbf{B}^T \boldsymbol{\sigma} . \tag{6.76}$$

Finally the internal force is transformed to the global system using the transformation matrix.
To compute the stiffness matrix for implicit we neglect the geometric contribution and just apply

$$\mathbf{K} = \mathbf{B}^T \mathbf{C} \mathbf{B} \tag{6.77}$$

where $\mathbf{C}$ is the material tangent modulus. Again the matrix must be transformed to the global system before used in the implicit solver.

7. BELYTSCHKO-LIN-TSAY SHELL

The Belytschko-Lin-Tsay shell element ([Belytschko and Tsay 1981], [Belytschko et al., 1984a]) was implemented in LS-DYNA as a computationally efficient alternative to the Hughes-Liu shell element. For a shell element with five through thickness integration points, the Belytschko-Lin-Tsay shell elements requires 725 mathematical operations compared to 4050 operations for the under integrated Hughes-Liu element. The selectively reduced integration formulation of the explicit Hughes-Liu element requires 35,350 mathematical operations. Because of its computational efficiency, the Belytschko-Lin-Tsay shell element is usually the shell element formulation of choice. For this reason, it has become the default shell element formulation for explicit calculations.

The Belytschko-Lin-Tsay shell element is based on a combined co-rotational and velocity-strain formulation. The efficiency of the element is obtained from the mathematical simplifications that result from these two kinematical assumptions. The co-rotational portion of the formulation avoids the complexities of nonlinear mechanics by embedding a coordinate system in the element. The choice of velocity-strain or rate-of-deformation in the formulation facilitates the constitutive evaluation, since the conjugate stress is the physical Cauchy stress. We closely follow the notation of Belytschko, Lin, and Tsay in the following development.

7.1 Co-rotational Coordinates

The midsurface of the quadrilateral shell element, or reference surface, is defined by the location of the element's four corner nodes. An embedded element coordinate system (see Figure 7.1) that deforms with the element is defined in terms of these nodal coordinates. Then the procedure for constructing the co-rotational coordinate system begins by calculating a unit vector normal to the main diagonal of the element:

$$\hat{e}_3 = \frac{s_3}{\|s_3\|} \tag{7.1a}$$

$$\|s_3\| = \sqrt{s_{31}^2 + s_{32}^2 + s_{33}^2} \tag{7.1b}$$

$$s_3 = r_{31} \times r_{42} \tag{7.1c}$$

where the superscript caret $(\hat{\cdot})$ is used to indicate the local (element) coordinate system.

It is desired to establish the local x axis $\hat{x}$ approximately along the element edge between nodes 1 and 2. This definition is convenient for interpreting the element stresses, which are defined in the local $\hat{x}-\hat{y}$ coordinate system. The procedure for constructing this unit vector is to define a vector s_1 that is nearly parallel to the vector r_{21}, viz.

$$s_1 = r_{21} - (r_{21} \cdot \hat{e}_3)\hat{e}_3 \tag{7.2a}$$

$$\hat{e}_1 = \frac{s_1}{\|s_1\|} \tag{7.2b}$$

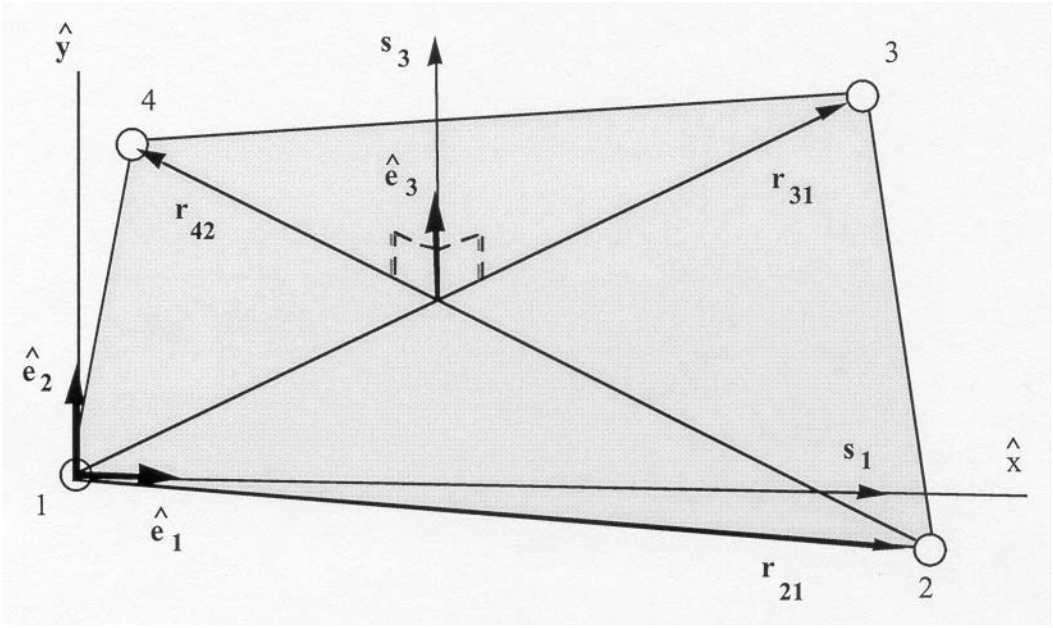


Figure 7.1. Construction of element coordinate system is shown.

The remaining unit vector is obtained from the vector cross product

$$\hat{e}_2 = \hat{e}_3 \times \hat{e}_1 \tag{7.3}$$

If the four nodes of the element are coplanar, then the unit vectors $\hat{e}_1$ and $\hat{e}_2$ are tangent to the midplane of the shell and $\hat{e}_3$ is in the fiber direction. As the element deforms, an angle may develop between the actual fiber direction and the unit normal $\hat{e}_3$. The magnitude of this angle may be characterized as

$$\left|\hat{e}_3 \cdot f - 1\right| < \delta \tag{7.4}$$

where f is the unit vector in the fiber direction and the magnitude of δ depends on the magnitude of the strains. According to Belytschko et al., for most engineering applications, acceptable values of δ are on the order of 10^{-2} and if the condition presented in Equation (7.4) is met, then the difference between the rotation of the co-rotational coordinates $\hat{e}$ and the material rotation should be small.

The global components of this co-rotational triad define a transformation matrix between the global and local element coordinate systems. This transformation operates on vectors with global components $\boldsymbol{A} = (A_x, A_y, A_z)$ and element coordinate components $\hat{A} = \left(\hat{A}_x, \hat{A}_y, \hat{A}_z\right)$, and is defined as:

$$\{A\} = \begin{Bmatrix} A_x \\ A_y \\ A_z \end{Bmatrix} = \begin{bmatrix} e_{1x} & e_{2x} & e_{3x} \\ e_{1y} & e_{2y} & e_{3y} \\ e_{1z} & e_{2z} & e_{3z} \end{bmatrix} \begin{Bmatrix} \hat{A}_x \\ \hat{A}_y \\ \hat{A}_z \end{Bmatrix} = [\mu]\{\hat{A}\} = [q]^T \{\hat{A}\} \tag{7.5a}$$

where e_{ix}, e_{iy}, e_{iz} are the global components of the element coordinate unit vectors. The inverse transformation is defined by the matrix transpose, i.e.,

$$\{\hat{A}\} = [\mu]^T \{A\} \tag{7.5b}$$

7.2 Velocity-Strain Displacement Relations

The above small rotation condition, Equation (7.4), does not restrict the magnitude of the element's rigid body rotations. Rather, the restriction is placed on the out-of-plane deformations, and, thus, on the element strain. Consistent with this restriction on the magnitude of the strains, the velocity-strain displacement relations used in the Belytschko-Lin-Tsay shell are also restricted to small strains.

As in the Hughes-Liu shell element, the displacement of any point in the shell is partitioned into a midsurface displacement (nodal translations) and a displacement associated with rotations of the element's fibers (nodal rotations). The Belytschko-Lin-Tsay shell element uses the Mindlin [1951] theory of plates and shells to partition the velocity of any point in the shell as:

$$v = v^m - \hat{z}\, e_3 \times \theta \tag{7.6}$$

where v^m is the velocity of the mid-surface, θ is the angular velocity vector, and $\hat{z}$ is the distance along the fiber direction (thickness) of the shell element. The corresponding co-rotational components of the velocity strain (rate of deformation) are given by

$$\hat{d}_{ij} = \frac{1}{2}\left(\frac{\partial \hat{v}_i}{\partial \hat{x}_j} + \frac{\partial \hat{v}_j}{\partial \hat{x}_i} \right) \tag{7.7}$$

Substitution of Equation (7.6) into the above yields the following velocity-strain relations:

$$\hat{d}_x = \frac{\partial \hat{v}_x^m}{\partial \hat{x}} + \hat{z}\frac{\partial \hat{\theta}_y}{\partial \hat{x}} \tag{7.8a}$$

$$\hat{d}_y = \frac{\partial \hat{v}_y^m}{\partial \hat{y}} - \hat{z}\frac{\partial \hat{\theta}_x}{\partial \hat{y}} \tag{7.8b}$$

$$2\hat{d}_{xy} = \frac{\partial \hat{v}_x^m}{\partial \hat{y}} + \frac{\partial \hat{v}_y^m}{\partial \hat{x}} + \hat{z}\left(\frac{\partial \hat{\theta}_y}{\partial \hat{y}} - \frac{\partial \hat{\theta}_x}{\partial \hat{x}} \right) \tag{7.8c}$$

$$2\hat{d}_{yz} = \frac{\partial \hat{v}_z^m}{\partial \hat{y}} - \hat{\theta}_x \tag{7.8d}$$

$$2\hat{d}_{xz} = \frac{\partial \hat{v}_z^m}{\partial \hat{x}} + \hat{\theta}_y \tag{7.8e}$$

The above velocity-strain relations need to be evaluated at the quadrature points within the shell. Standard bilinear nodal interpolation is used to define the mid-surface velocity, angular velocity, and the element's coordinates (isoparametric representation). These interpolations relations are given by

$$v^m = N_I(\xi,\eta)\, v_I \tag{7.9a}$$

$$\theta^m = N_I(\xi,\eta)\, \theta_I \tag{7.9b}$$

$$x^m = N_I(\xi,\eta)\, x_I \tag{7.9c}$$

where the subscript I is summed over all the nodes of the element and the nodal velocities are obtained by differentiating the nodal coordinates with respect to time, i.e., $v_I = \dot{x}_I$. The bilinear shape functions are

$$N_1 = \frac{1}{4}(1-\xi)(1-\eta) \tag{7.10a}$$

$$N_2 = \frac{1}{4}(1+\xi)(1-\eta) \tag{7.10b}$$

$$N_3 = \frac{1}{4}(1+\xi)(1+\eta) \tag{7.10c}$$

$$N_4 = \frac{1}{4}(1-\xi)(1+\eta) \tag{7.10d}$$

The velocity-strains at the center of the element, i.e., at $\xi = 0$, and $\eta = 0$, are obtained by substitution of the above relations into the previously defined velocity-strain displacement relations, Equations (7.8a) and (7.8e). After some algebra, this yields

$$\hat{d}_x = B_{1I}\hat{v}_{xI} + \hat{z}B_{1I}\hat{\theta}_{yI} \tag{7.11a}$$

$$\hat{d}_y = B_{2I}\hat{v}_{yI} - \hat{z}B_{2I}\hat{\theta}_{xI} \tag{7.11b}$$

$$2\hat{d}_{xy} = B_{2I}\,\hat{v}_{xI} + B_{1I}\hat{v}_{yI} + \hat{z}\left(B_{2I}\hat{\theta}_{yI} - B_{1I}\hat{\theta}_{xI}\right) \tag{7.11c}$$

$$2\hat{d}_{xz} = B_{1I}\hat{v}_{zI} + N_I\hat{\theta}_{yI} \tag{7.11d}$$

$$2\,\hat{d}_{yz} = B_{2I}\,\hat{v}_{zI} - N_I\,\hat{\theta}_{xI} \tag{7.11e}$$

where

$$B_{1I} = \frac{\partial N_I}{\partial \hat{x}} \tag{7.12a}$$

$$B_{2I} = \frac{\partial N_I}{\partial \hat{y}} \tag{7.12b}$$

The shape function derivatives B_{aI} are also evaluated at the center of the element, i.e., at $\xi = 0$, and $\eta = 0$.

7.3 Stress Resultants and Nodal Forces

After suitable constitutive evaluations using the above velocity-strains, the resulting stresses are integrated through the thickness of the shell to obtain local resultant forces and moments. The integration formula for the resultants are

$$\hat{f}^R_{\alpha\beta} = \int \hat{\sigma}_{\alpha\beta}\, d\,\hat{z} \tag{7.13a}$$

$$\hat{m}^R_{\alpha\beta} = -\int \hat{z}\hat{\sigma}_{\alpha\beta} d\hat{z} \tag{7.13b}$$

where the superscript, R, indicates a resultant force or moment, and the Greek subscripts emphasize the limited range of the indices for plane stress plasticity.

The above element-centered force and moment resultants are related to the local nodal forces and moments by invoking the principle of virtual power and integrating with a one-point quadrature. The relations obtained in this manner are

$$\hat{f}_{xI} = A\left(B_{1I}\,\hat{f}^R_{xx} + B_{2I}\,\hat{f}^R_{xy}\right) \tag{7.14a}$$

$$\hat{f}_{yI} = A\left(B_{2I}\,\hat{f}^R_{yy} + B_{1I}\,\hat{f}^R_{xy}\right) \tag{7.14b}$$

$$\hat{f}_{zI} = A\kappa\left(B_{1I}\,\hat{f}^R_{xz} + B_{2I}\,\hat{f}^R_{yz}\right) \tag{7.14c}$$

$$\hat{m}_{xI} = A\left(B_{2I}\hat{m}^R_{yy} + B_{1I}\hat{m}^R_{xy} - \frac{\kappa}{4}\hat{f}^R_{yz} \right) \tag{7.14d}$$

$$\hat{m}_{yI} = -A\left(B_{1I}\hat{m}^R_{xx} + B_{2I}\hat{m}^R_{xy} - \frac{\kappa}{4}\hat{f}^R_{xz} \right) \tag{7.14e}$$

$$\hat{m}_{zI} = 0 \tag{7.14f}$$

where A is the area of the element, and κ is the shear factor from the Mindlin theory. In the Belytschko-Lin-Tsay formulation, κ is used as a penalty parameter to enforce the Kirchhoff normality condition as the shell becomes thin.

The above local nodal forces and moments are then transformed to the global coordinate system using the transformation relations given previously as Equation (7.5a). The global nodal forces and moments are then appropriately summed over all the nodes and the global equations of motion are solved for the next increment in nodal accelerations.

7.4 Hourglass Control (Belytschko-Lin-Tsay)

In part, the computational efficiency of the Belytschko-Lin-Tsay and the under integrated Hughes-Liu shell elements are derived from their use of one-point quadrature in the plane of the element. To suppress the hourglass deformation modes that accompany one-point quadrature, hourglass viscosity stresses are added to the physical stresses at the local element level. The discussion of the hourglass control that follows pertains to the Hughes-Liu and the membrane elements as well.

The hourglass control used by Belytschko et al., extends an earlier derivation by Flanagan and Belytschko [1981], (see also Kosloff and Frazier [1978], Belytschko and Tsay [1983]). The hourglass shape vector, τ_I, is defined as

$$\tau_I = h_I - \left(h_J \hat{x}_{aJ}\right) B_{aI} \tag{7.15}$$

where

$$h = \begin{bmatrix} +1 \\ -1 \\ +1 \\ -1 \end{bmatrix} \tag{7.16}$$

is the basis vector that generates the deformation mode that is neglected by one-point quadrature. In Equation (7.15) and the reminder of this subsection, the Greek subscripts have a range of 2, e.g., $\hat{x}_{aI} = \left(\hat{x}_{1I}, \hat{x}_{2I}\right) = \left(\hat{x}_I, \hat{y}_I\right)$.

The hourglass shape vector then operates on the generalized displacements, in a manner similar to Equations (7.11a - e), to produce the generalized hourglass strain rates

$$\dot{q}^B_\alpha = \tau_I \hat{\theta}_{\alpha I} \tag{7.17a}$$

$$\dot{q}_3^B = \tau_I \, \hat{v}_{zI} \tag{7.17b}$$

$$\dot{q}_\alpha^M = \tau_I \, \hat{v}_{\alpha I} \tag{7.17c}$$

where the superscripts B and M denote bending and membrane modes, respectively. The corresponding hourglass stress rates are then given by

$$\dot{Q}_\alpha^B = \frac{r_\theta E t^3 A}{192} B_{\beta I} B_{\beta I} \dot{q}_\alpha^B \tag{7.18a}$$

$$\dot{Q}_3^B = \frac{r_w \kappa G t^3 A}{12} B_{\beta I} B_{\beta I} \dot{q}_3^B \tag{7.18b}$$

$$\dot{Q}_\alpha^M = \frac{r_m E t A}{8} B_{\beta I} B_{\beta I} \dot{q}_\alpha^M \tag{7.18c}$$

where t is the shell thickness and the parameters, r_θ, r_w, and r_m are generally assigned values between 0.01 and 0.05.

Finally, the hourglass stresses, which are updated from the stress rates in the usual way, i.e.,

$$Q^{n+1} = Q^n + \Delta t \dot{Q} \tag{7.19}$$

and the hourglass resultant forces are then

$$\hat{m}_{\alpha I}^H = \tau_I Q_\alpha^B \tag{7.20a}$$

$$\hat{f}_{3I}^H = \tau_I Q_3^B \tag{7.20b}$$

$$\hat{f}_{\alpha I}^H = \tau_I Q_\alpha^M$$

where the superscript H emphasizes that these are internal force contributions from the hourglass deformations. These hourglass forces are added directly to the previously determined local internal forces due to deformations Equations (7.14a - f). These force vectors are orthogonalized with respect to rigid body motion.

7.5 Hourglass Control (Englemann and Whirley)

Englemann and Whirley [1991] developed an alternative hourglass control, which they implemented in the framework of the Belytschko, Lin, and Tsay shell element. We will briefly highlight their procedure here that has proven to be cost effective-only twenty percent more expensive than the default control.

In the hourglass procedure, the in-plane strain field (subscript p) is decomposed into the one point strain field plus the stabilization strain field:

$$\dot{\bar{\varepsilon}}_p = \dot{\bar{\varepsilon}}^0_p + \dot{\bar{\varepsilon}}^s_p \tag{7.21}$$

where the stabilization strain field, which is obtained from the assumed strain fields of Pian and Sumihara [1984], is given in terms of the hourglass velocity field as

$$\dot{\bar{\varepsilon}}^s_p = W_m \dot{q}_m + z W_b \dot{q}_b \tag{7.22}$$

Here, W_m and W_b play the role of stabilization strain velocity operators for membrane and bending:

$$W_m = \begin{bmatrix} f_1^p(\xi,\eta) & f_4^p(\xi,\eta) \\ f_2^p(\xi,\eta) & f_5^p(\xi,\eta) \\ f_3^p(\xi,\eta) & f_6^p(\xi,\eta) \end{bmatrix} \tag{7.23}$$

$$W_b = \begin{bmatrix} -f_4^p(\xi,\eta) & f_1^p(\xi,\eta) \\ -f_5^p(\xi,\eta) & f_2^p(\xi,\eta) \\ -f_6^p(\xi,\eta) & f_3^p(\xi,\eta) \end{bmatrix} \tag{7.24}$$

where the terms $f_i^p(\xi,\eta)$ $i = 1,2,...,6$, are rather complicated and the reader is referred to the reference [Englemann and Whirley, 1991].

To obtain the transverse shear assumed strain field, the procedure given in [Bathe and Dvorkin, 1984] is used. The transverse shear strain field can again be decomposed into the one point strain field plus the stabilization field:

$$\dot{\bar{\varepsilon}}_s = \dot{\bar{\varepsilon}}^0_s + \dot{\bar{\varepsilon}}^s_s \tag{7.25}$$

that is related to the hourglass velocities by

$$\dot{\bar{\varepsilon}}^s_s = W_s \dot{q}_s \tag{7.26}$$

where the transverse shear stabilization strain-velocity operator W_s is given by

$$W_s = \begin{bmatrix} f_1^s(\xi,\eta) & -g_1^s\xi & g_2^s\eta & g_3^s\xi & g_3^s\eta \\ f_2^s(\xi,\eta) & g_4^s\xi & g_4^s\eta & -g_2^s\xi & g_1^s\eta \end{bmatrix} \tag{7.27}$$

Again, the coefficients $f_1^s(\xi,\eta)$ and g_1^s are defined in the reference.

In their formulation, the hourglass forces are related to the hourglass velocity field through an incremental hourglass constitutive equation derived from an additive decomposition of the stress into a "one-point stress," plus a "stabilization stress." The integration of the

stabilization stress gives a resultant constitutive equation relating hourglass forces to hourglass velocities.

The in-plane and transverse stabilization stresses are updated according to:

$$\tau_s^{s,n+1} = \tau_s^{s,n} + \Delta t c_s C_s \dot{\bar{\varepsilon}}_s^s$$
$$\tau_s^{s,n+1} = \tau_s^{s,n} + \Delta t c_s C_s \dot{\bar{\varepsilon}}_s^s \tag{7.28}$$

where the tangent matrix is the product of a matrix C, which is constant within the shell domain, and a scalar c that is constant in the plane but may vary through the thickness.

The stabilization stresses can now be used to obtain the hourglass forces:

$$Q_m = \int_{-\frac{h}{2}}^{\frac{h}{2}} \int_A W_m^T \tau_p^s dA dz$$
$$Q_b = \int_{-\frac{h}{2}}^{\frac{h}{2}} \int_A W_b^T \tau_p^s dA dz \tag{7.29}$$
$$Q_s = \int_{-\frac{h}{2}}^{\frac{h}{2}} \int_A W_s^T \tau_s^s dA dz$$

7.6 Belytschko-Wong-Chiang Improvements

Since the Belytschko-Tsay element is based on a perfectly flat geometry, warpage is not considered. Although this generally poses no major difficulties and provides for an efficient element, incorrect results in the twisted beam problem, See Figure 7.2, are obtained where the nodal points of the elements used in the discretization are not coplanar. The Hughes-Liu shell element considers non-planar geometry and gives good results on the twisted beam, but is relatively expensive. The effect of neglecting warpage in typical a application cannot be predicted beforehand and may lead to less than accurate results, but the latter is only speculation and is difficult to verify in practice. Obviously, it would be better to use shells that consider warpage if the added costs are reasonable and if this unknown effect is eliminated. In this section we briefly describe the simple and computationally inexpensive modifications necessary in the Belytschko-Tsay shell to include the warping stiffness. The improved transverse shear treatment is also described which is necessary for the element to pass the Kirchhoff patch test. Readers are directed to the references [Belytschko, Wong, and Chang 1989, 1992] for an in depth theoretical background.

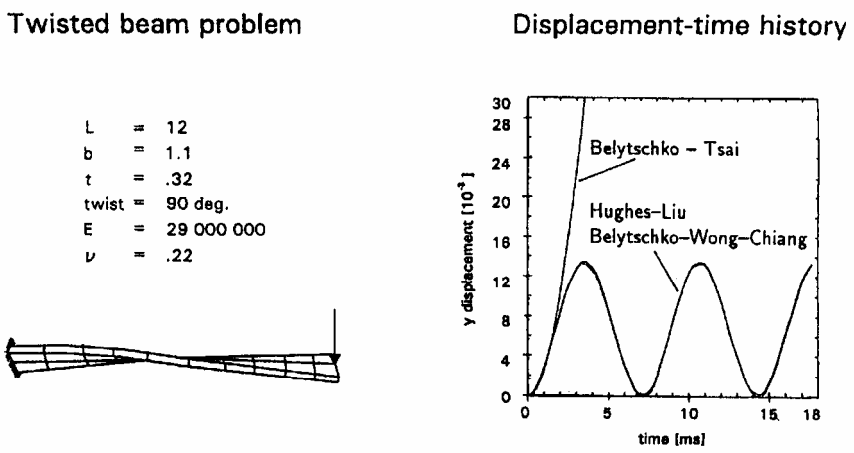


Figure 7.2. The twisted beam problem fails with the Belytschko-Tsay shell element.

In order to include warpage in the formulation it is convenient to define nodal fiber vectors as shown in Figure 7.3. The geometry is interpolated over the surface of the shell from:

$$x = x^m + \bar{\zeta}\, p = \left(x_I + \bar{\zeta}\, p_I\right) N_I\left(\xi, \eta\right) \tag{7.30}$$

where

$$\bar{\zeta} = \frac{\zeta h}{2}$$

and ζ is a parametric coordinate which varies between -1 to +1.

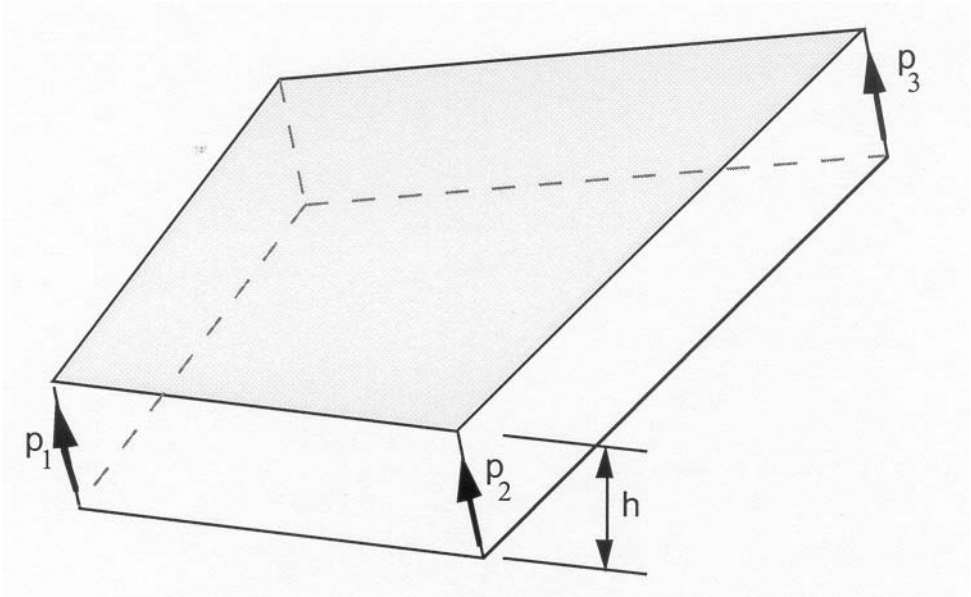


Figure 7.3. Nodal fiber vectors p_1, p_2, and p_3, where h is the thickness.

The in plane strain components are given by:

$$d_{xx} = b_{xI}\hat{v}_{xI} + \overline{\zeta}\left(b^c_{xI}\hat{v}_{xI} + b_{xI}\dot{p}_{xI}\right) \tag{7.31a}$$

$$d_{yy} = b_{yI}\hat{v}_{yI} + \overline{\zeta}\left(b^c_{yI}\hat{v}_{yI} + b_{yI}\dot{p}_{yI}\right) \tag{7.31b}$$

$$d_{xy} = \frac{1}{2}b_{xI}\hat{v}_{yI} + b_{yI}\hat{v}_{xI} + \overline{\zeta}\left(b^c_{xI}\hat{v}_{yI} + b_{xI}\dot{p}_{yI} + b^c_{yI}\hat{v}_{xI} + b_{yI}\dot{p}_{xI}\right) \tag{7.31c}$$

The coupling terms are come in through b^c_{iI} : which is defined in terms of the components of the fiber vectors as:

$$\begin{Bmatrix} b^c_{xI} \\ b^c_{yI} \end{Bmatrix} = \begin{bmatrix} p_{\hat{y}2} - p_{\hat{y}4} & p_{\hat{y}3} - p_{\hat{y}1} & p_{\hat{y}4} - p_{\hat{y}2} & p_{\hat{y}1} - p_{\hat{y}3} \\ p_{\hat{x}2} - p_{\hat{x}4} & p_{\hat{x}3} - p_{\hat{x}1} & p_{\hat{x}4} - p_{\hat{x}2} & p_{\hat{x}1} - p_{\hat{x}3} \end{bmatrix} \tag{7.32}$$

For a flat geometry the normal vectors are identical and no coupling can occur. Two methods are used by Belytschko for computing b^c_{iI} and the reader is referred to his papers for the details. Both methods have been tested in LS-DYNA and comparable results were obtained.

The transverse shear strain components are given as

$$\hat{\gamma}_{xz} = -N_I(\xi,\eta)\overline{\theta}_{\hat{y}I} \tag{7.33a}$$

$$\hat{\gamma}_{yz} = -N_I(\xi,\eta)\overline{\theta}_{\hat{x}I} \tag{7.33b}$$

where the nodal rotational components are defined as:

$$\overline{\theta}_{\hat{x}I} = \left(e^I_n \cdot e_{\hat{x}}\right)\overline{\theta}^I_n + \left(e^K_n \cdot e_{\hat{x}}\right)\overline{\theta}^K_n \tag{7.34a}$$

$$\overline{\theta}_{\hat{y}I} = \left(e^I_n \cdot e_{\hat{y}}\right)\overline{\theta}^I_n + \left(e^K_n \cdot e_{\hat{y}}\right)\overline{\theta}^K_n \tag{7.34b}$$

The rotation $\overline{\theta}^I_n$ comes from the nodal projection

$$\overline{\theta}^I_n = \frac{1}{2}\left(\theta^I_{nI} + \theta^I_{nJ}\right) + \frac{1}{L^{IJ}}\left(\hat{v}_{zJ} - \hat{v}_{zJ}\right) \tag{7.35}$$

where the subscript n refers to the normal component of side I as seen in Figure 7.3 and L^{IJ} is the length of side IJ .

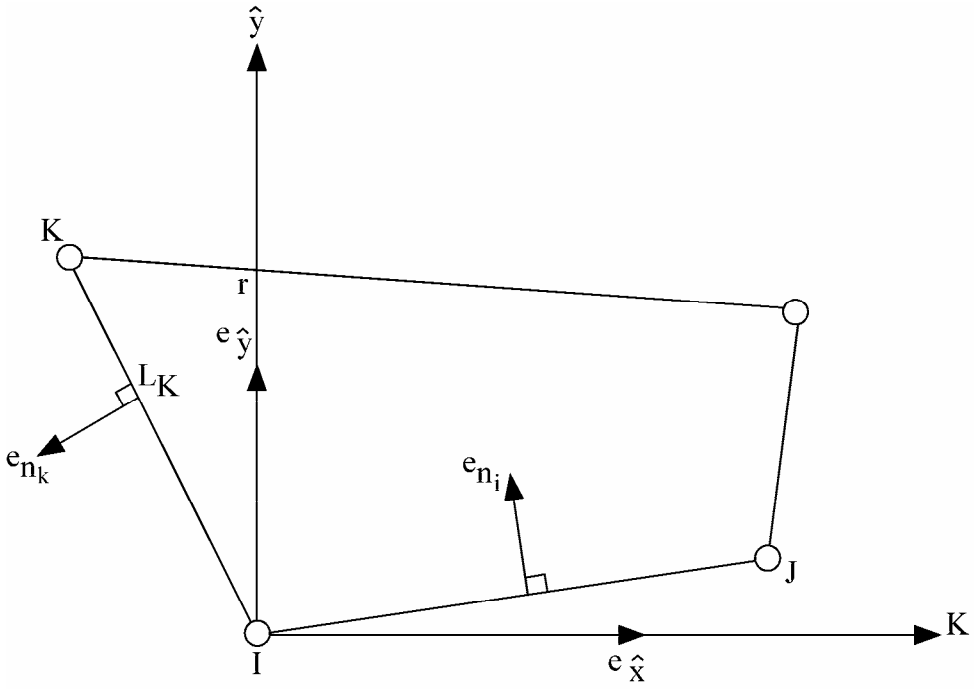


Figure 7.4. Vector and edge definitions for computing the transverse shear strain components.

8. C^0 TRIANGULAR SHELL

The C^0 shell element due to Kennedy, Belytschko, and Lin [1986] has been implemented as a computationally efficient triangular element complement to the Belytschko-Lin-Tsay quadrilateral shell element ([Belytschko and Tsay 1981], [Belytschko et al., 1984a]). For a shell element with five through-the-thickness integration points, the element requires 649 mathematical operations (the Belytschko-Lin-Tsay quadrilateral shell element requires 725 mathematical operations) compared to 1417 operations for the Marchertas-Belytschko triangular shell [Marchertas and Belytschko 1974] (referred to as the BCIZ [Bazeley, Cheung, Irons, and Zienkiewicz 1965] triangular shell element in the DYNA3D user's manual).

Triangular shell elements are offered as optional elements primarily for compatibility with local user grid generation and refinement software. Many computer aided design (CAD) and computer aided manufacturing (CAM) packages include finite element mesh generators, and most of these mesh generators use triangular elements in the discretization. Similarly, automatic mesh refinement algorithms are typically based on triangular element discretization. Also, triangular shell element formulations are not subject to zero energy modes inherent in quadrilateral element formulations.

The triangular shell element's origins are based on the work of Belytschko et al., [Belytschko, Stolarski, and Carpenter 1984b] where the linear performance of the shell was demonstrated. Because the triangular shell element formulations parallels closely the formulation of the Belytschko-Lin-Tsay quadrilateral shell element presented in the previous section (Section 7), the following discussion is limited to items related specifically to the triangular shell element.

8.1 Co-rotational Coordinates

The mid-surface of the triangular shell element, or reference surface, is defined by the location of the element's three nodes. An embedded element coordinate system (see Figure 8.1) that deforms with the element is defined in terms of these nodal coordinates. The procedure for constructing the co-rotational coordinate system is simpler than the corresponding procedure for the quadrilateral, because the three nodes of the triangular element are guaranteed coplanar.

The local x-axis $\hat{x}$ is directed from node 1 to 2. The element's normal axis $\hat{z}$ is defined by the vector cross product of a vector along $\hat{x}$ with a vector constructed from node 1 to node 3. The local y-axis $\hat{y}$ is defined by a unit vector cross product of $\hat{e}_3$ with $\hat{e}_1$, which are the unit vectors in the $\hat{z}$ directions, respectively. As in the case of the quadrilateral element, this triad of co-rotational unit vectors defines a transformation between the global and local element coordinate systems (see Equations (7.5 a, b)).

8.2 Velocity-Strain Relations

As in the Belytschko-Lin-Tsay quadrilateral shell element, the displacement of any point in the shell is partitioned into a mid-surface displacement (nodal translations) and a displacement associated with rotations of the element's fibers (nodal rotations). The Kennedy-Belytschko-Lin triangular shell element also uses the Mindlin [Mindlin 1951] theory of plates and shells to partition the velocity of any point in the shell (recall Equation (7.6)):

$$v = v^m - \hat{z}\, e_3 \times \theta \tag{8.1}$$

where v^m is the velocity of the mid-surface, θ is the angular velocity vector, and $\hat{z}$ is the distance along the fiber direction (thickness) of the shell element. The corresponding co-rotational components of the velocity strain (rate of deformation) were given previously in Equation (7.11 a - e).

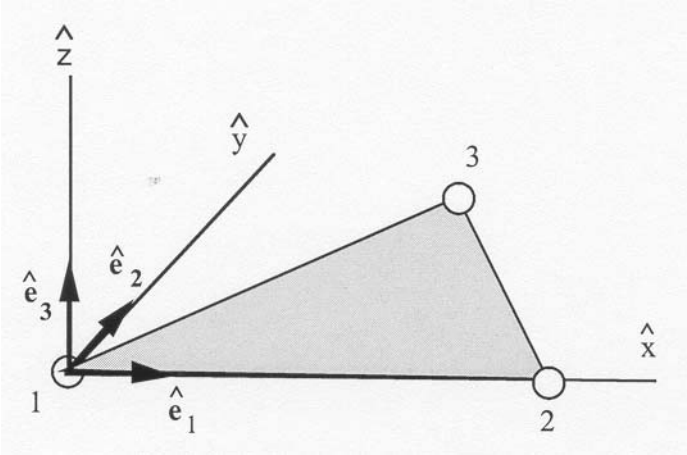


Figure 8.1. Local element coordinate system for C^0 shell element.

Standard linear nodal interpolation is used to define the midsurface velocity, angular velocity, and the element's coordinates (isoparametric representation). These interpolation functions are the area coordinates used in triangular element formulations. Substitution of the nodally interpolated velocity fields into the velocity-strain relations (see Belytschko et al., for details), leads to strain rate-velocity relations of the form

$$\hat{d} = B\,\hat{v} \tag{8.2}$$

where $\hat{d}$ are the velocity strains (strain rates), the elements of B are derivatives of the nodal interpolation functions, and the $\hat{v}$ are the nodal velocities and angular velocities.

It is convenient to partition the velocity strains and the B matrix into membrane and bending contributions. The membrane relations are given by

$$\begin{Bmatrix} \hat{d}_x \\ \hat{d}_y \\ 2\hat{d}_{xy} \end{Bmatrix}^M = \frac{1}{\hat{x}_2\hat{y}_3} \begin{bmatrix} \hat{y}_3 & 0 & \hat{y}_3 & 0 & 0 & 0 \\ 0 & \hat{x}_3-\hat{x}_2 & 0 & -\hat{x}_3 & 0 & \hat{x}_2 \\ \hat{x}_3-\hat{x}_2 & -\hat{y}_3 & -\hat{x}_3 & \hat{y}_3 & \hat{x}_2 & 0 \end{bmatrix} \begin{Bmatrix} \hat{v}_{x1} \\ \hat{v}_{y1} \\ \hat{v}_{x2} \\ \hat{v}_{y2} \\ \hat{v}_{x3} \\ \hat{v}_{y3} \end{Bmatrix} \tag{8.3}$$

or

$$\hat{d}^{\,M} = B_M\,\hat{v} \tag{8.4}$$

The bending relations are given by

$$\begin{Bmatrix} \hat{\kappa}_x \\ \hat{\kappa}_y \\ 2\hat{\kappa}_{xy} \end{Bmatrix} = \frac{-1}{\hat{x}_2\hat{y}_3} \begin{bmatrix} 0 & -\hat{y}_3 & 0 & \hat{y}_3 & 0 & 0 \\ \hat{x}_3-\hat{x}_2 & 0 & \hat{x}_3 & 0 & -\hat{x}_2 & 0 \\ \hat{y}_3 & \hat{x}_3-\hat{x}_2 & -\hat{y}_3 & -\hat{x}_3 & 0 & \hat{x}_2 \end{bmatrix} \begin{Bmatrix} \hat{\theta}_{x1} \\ \hat{\theta}_{y1} \\ \hat{\theta}_{x2} \\ \hat{\theta}_{y2} \\ \hat{\theta}_{x3} \\ \hat{\theta}_{y3} \end{Bmatrix} \tag{8.5}$$

or

$$\hat{\kappa}^M = B_M \hat{\theta}^{\text{def}} \tag{8.6}$$

The local element velocity strains are then obtained by combining the above two relations:

$$\begin{Bmatrix} \hat{d}_x \\ \hat{d}_y \\ 2\hat{d}_{xy} \end{Bmatrix} = \begin{Bmatrix} \hat{d}_x \\ \hat{d}_y \\ 2\hat{d}_{xy} \end{Bmatrix}^M - \hat{z} \begin{Bmatrix} \hat{\kappa}_x \\ \hat{\kappa}_y \\ 2\hat{\kappa}_{xy} \end{Bmatrix} = \hat{d}^M - \hat{z}\hat{\kappa} \tag{8.7}$$

The remaining two transverse shear strain rates are given by

$$\begin{Bmatrix} 2\hat{d}_{xz} \\ 2\hat{d}_{yz} \end{Bmatrix} = \frac{1}{6\hat{x}_2\hat{y}_3}$$

$$\begin{bmatrix} -\hat{y}_3^2 & \hat{y}_3(2\hat{x}_2+\hat{x}_3) & \hat{y}_3^2 & \hat{y}_3(3\hat{x}_2-\hat{x}_3) & 0 & \hat{x}_2\hat{y}_3 \\ \hat{y}_3(\hat{x}_2-2\hat{x}_2) & \hat{x}_2^2-\hat{x}_3^2 & -\hat{y}_3{}^2(\hat{x}_2+\hat{x}_3) & \hat{x}_3(\hat{x}_3-2\hat{x}_2) & -3\hat{x}_2\hat{y}_3 & \hat{x}_2(2\hat{x}_3-\hat{x}_2) \end{bmatrix}$$

$$\begin{Bmatrix} \hat{\theta}_{x1} \\ \hat{\theta}_{y1} \\ \hat{\theta}_{x2} \\ \hat{\theta}_{y2} \\ \hat{\theta}_{x3} \\ \hat{\theta}_{y3} \end{Bmatrix}^{\text{def}} \tag{8.8}$$

or

$$\hat{d}^{\,S} = B_S \hat{\theta}^{\,\mathrm{def}} \tag{8.9}$$

All of the above velocity-strain relations have been simplified by using one-point quadrature.

In the above relations, the angular velocities $\hat{\theta}^{\,\mathrm{def}}$ are the deformation component of the angular velocity $\hat{\theta}$ obtained by subtracting the portion of the angular velocity due to rigid body rotation, i.e.,

$$\hat{\theta}^{\,\mathrm{def}} = \hat{\theta} - \hat{\theta}^{\,\mathrm{rig}} \tag{8.10}$$

The two components of the rigid body angular velocity are given by

$$\hat{\theta}_y^{\mathrm{rig}} = \frac{\hat{v}_{z1} - \hat{v}_{z2}}{\hat{x}_2} \tag{8.11a}$$

$$\hat{\theta}_x^{\mathrm{rig}} = \frac{(\hat{v}_{z3} - \hat{v}_{z1})\,\hat{x}_2 - (\hat{v}_{z2} - \hat{v}_{z1})\,\hat{x}_3}{\hat{x}_2 \hat{y}_3} \tag{8.11b}$$

The first of the above two relations is obtained by considering the angular velocity of the local x-axis about the local y-axis. Referring to Figure 8.1, by construction nodes 1 and 2 lie on the local x-axis and the distance between the nodes is $\hat{x}_2$ i.e., the $\hat{x}$ distance from node 2 to the local coordinate origin at node 1. Thus the difference in the nodal $\hat{z}$ velocities divided by the distance between the nodes is an average measure of the rigid body rotation rate about the local y-axis.

The second relation is conceptually identical, but is implemented in a slightly different manner due to the arbitrary location of node 3 in the local coordinate system. Consider the two local element configurations shown in Figure 8.2. For the leftmost configuration, where node 3 is the local y-axis, the rigid body rotation rate about the local x-axis is given by

$$\hat{\theta}_{x-left}^{\mathrm{rig}} = \frac{\hat{v}_{z3} - \hat{v}_{z1}}{\hat{y}_3} \tag{8.12}$$

and for the rightmost configuration the same rotation rate is given by

$$\hat{\theta}_{x-right}^{\mathrm{rig}} = \frac{\hat{v}_{z3} - \hat{v}_{z2}}{\hat{y}_3} \tag{8.13}$$

Although both of these relations yield the average rigid body rotation rate, the selection of the correct relation depends on the configuration of the element, i.e., on the location of node 3. Since every element in the mesh could have a configuration that is different in general from either of the two configurations shown in Figure 8.2, a more robust relation is needed to determine the average rigid body rotation rate about the local x-axis. In most typical grids, node 3 will be located somewhere between the two configurations shown in Figure 8.2. Thus a linear interpolation between these two rigid body rotation rates was devised using the distance $\hat{x}_3$ as the interpolant:

$$\hat{\theta}_x^{\text{rig}} = \hat{\theta}_{x-left}^{\text{rig}} \left(1 - \frac{\hat{x}_3}{\hat{x}_2} \right) + \hat{\theta}_{x-right}^{\text{rig}} \left(\frac{\hat{x}_3}{\hat{x}_2} \right) \tag{8.14}$$

Substitution of Equations (8.12) and (8.13) into (8.14) and simplifying produces the relations given previously as Equation (8.11b).

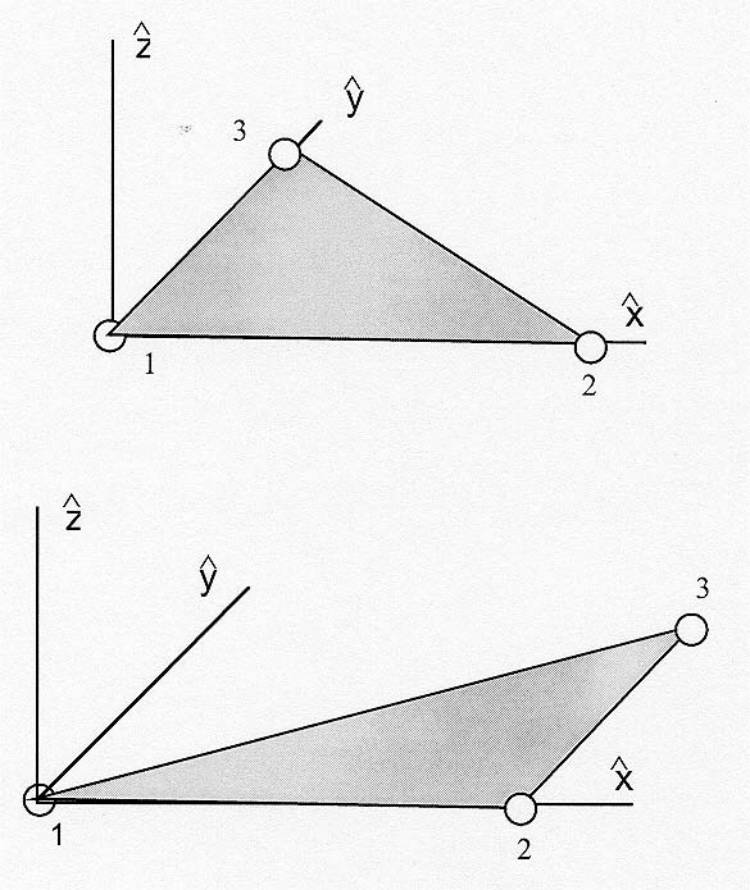


Figure 8.2. Element configurations with node 3 aligned with node 1 (left) and node 3 aligned with node 2 (right).

8.3 Stress Resultants and Nodal Forces

After suitable constitutive evaluation using the above velocity strains, the resulting local stresses are integrated through the thickness of the shell to obtain local resultant forces and moments. The integration formulae for the resultants are

$$\hat{f}_{\alpha\beta}^R = \int \hat{\sigma}_{\alpha\beta} d\,\hat{z} \tag{8.15a}$$

$$\hat{m}_{\alpha\beta}^R = -\int \hat{z}\,\hat{\sigma}_{\alpha\beta} d\,\hat{z} \tag{8.15b}$$

where the superscript R indicates a resultant force or moment and the Greek subscripts emphasize the limited range of the indices for plane stress plasticity.

The above element-centered force and moment resultant are related to the local nodal forces and moments by invoking the principle of virtual power and performing a one-point quadrature. The relations obtained in this manner are

$$\begin{Bmatrix} \hat{f}_{x1} \\ \hat{f}_{y1} \\ \hat{f}_{x2} \\ \hat{f}_{y2} \\ \hat{f}_{x3} \\ \hat{f}_{y3} \end{Bmatrix} = A B_M^T \begin{Bmatrix} \hat{f}_{xx}^R \\ \hat{f}_{yy}^R \\ \hat{f}_{xy}^R \end{Bmatrix} \tag{8.16a}$$

$$\begin{Bmatrix} \hat{m}_{x1} \\ \hat{m}_{y1} \\ \hat{m}_{x2} \\ \hat{m}_{y2} \\ \hat{m}_{x3} \\ \hat{m}_{y3} \end{Bmatrix} = A B_M^T \begin{Bmatrix} \hat{m}_{xx}^R \\ \hat{m}_{yy}^R \\ \hat{m}_{xy}^R \end{Bmatrix} + A B_S^T \begin{Bmatrix} \hat{f}_{xz}^R \\ \hat{f}_{yz}^R \end{Bmatrix} \tag{8.16b}$$

where A is the area of the element $\left(2A = \hat{x}_2 \hat{y}_3\right)$.

The remaining nodal forces, the $\hat{z}$ component of the force $\left(\hat{f}_{z3}, \hat{f}_{z2}, \hat{f}_{z1}\right)$, are determined by successively solving the following equilibration equations

$$\hat{m}_{x1} + \hat{m}_{x2} + \hat{m}_{x3} + \hat{y}_3 \hat{f}_{z3} = 0 \tag{8.17a}$$

$$\hat{m}_{y1} + \hat{m}_{y2} + \hat{m}_{y3} - \hat{x}_3 \hat{f}_{z3} - \hat{x}_2 \hat{f}_{z2} = 0 \tag{8.17b}$$

$$\hat{f}_{z1} + \hat{f}_{z2} + \hat{f}_{z3} = 0 \tag{8.17c}$$

which represent moment equilibrium about the local x-axis, moment equilibrium about the local y-axis, and force equilibrium in the local z-direction, respectively.

9. MARCHERTAS-BELYTSCHKO TRIANGULAR SHELL

The Marchertas-Belytschko [1974] triangular shell element, or the BCIZ triangular shell element as it is referred to in the LS-DYNA user's manual, was developed in the same time period as the Belytschko beam element [Belytschko, Schwer, and Klein, 1977], see Section 4, forming the first generation of co-rotational structural elements developed by Belytschko and co-workers. This triangular shell element became the first triangular shell implemented in DYNA3D. Although the Marchertas-Belytschko shell element is relatively expensive, i.e., the C^0 triangular shell element with five through-the-thickness integration points requires 649 mathematical operations compared to 1,417 operations for the Marchertas-Belytschko triangular shell, it is maintained in LS-DYNA for compatibility with earlier user models. However, as the LS-DYNA user community moves to application of the more efficient shell element formulations, the use of the Marchertas-Belytschko triangular shell element will decrease.

As mentioned above, the Marchertas-Belytschko triangular shell has a common co-rotational formulation origin with the Belytschko beam element. The interested reader is referred to the beam element description, see Section 4, for details on the co-rotational formulation. In the next subsection a discussion of how the local element coordinate system is identical for the triangular shell and beam elements. The remaining subsections discuss the triangular element's displacement interpolants, the strain displacement relations, and calculations of the element nodal forces. In the report [1974], much greater detail is provided.

9.1 Element Coordinates

Figure 9.1(a) shows the element coordinate system, $(\hat{x}, \hat{y}, \hat{z})$ originating at Node 1, for the Marchertas-Belytschko triangular shell. The element coordinate system is associated with a triad of unit vectors (e_1, e_2, e_3) the components of which form a transformation matrix between the global and local coordinate systems for vector quantities. The nodal or body coordinate system unit vectors (b_1, b_2, b_3) are defined at each node and are used to define the rotational deformations in the element, see Section 9.4.

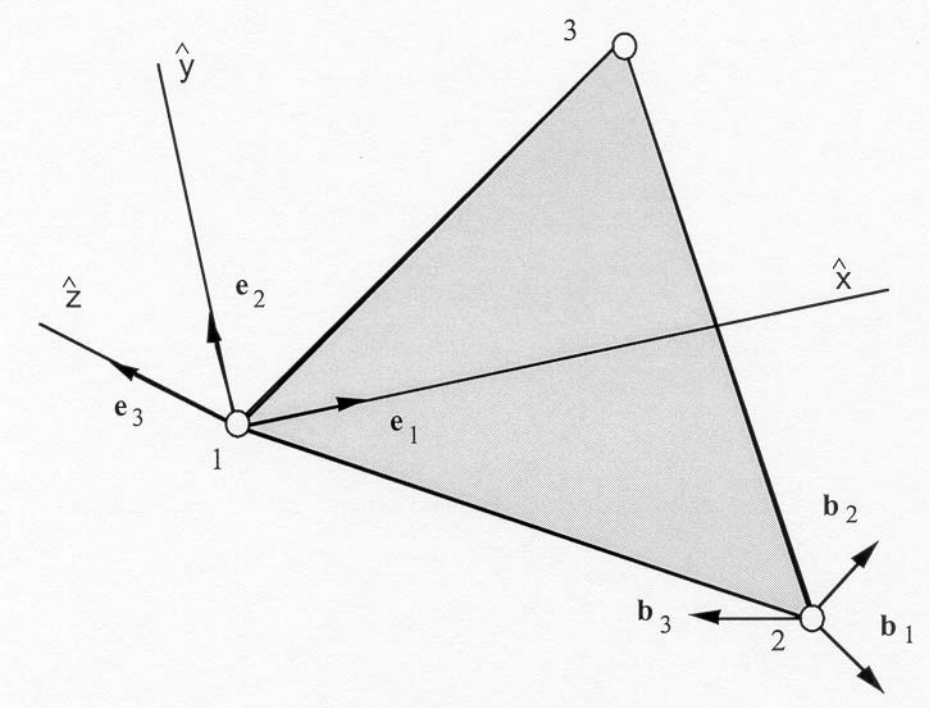
The unit normal to the shell element e_3 is formed from the vector cross product

$$e_3 = l_{21} \times l_{31} \tag{9.1}$$

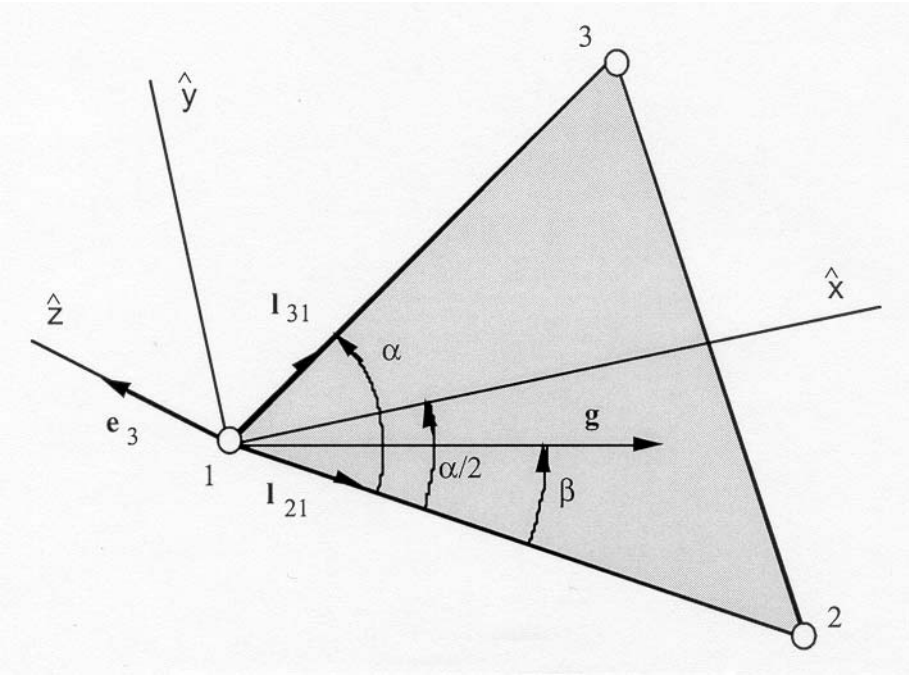
where l_{21} and l_{31} are unit vectors originating at Node 1 and pointing towards Nodes 2 and 3 respectively, see Figure 9.1(b).

Next a unit vector $\boldsymbol{g}$, see Figure 9.1(b), is assumed to be in the plane of the triangular element with its origin at Node 1 and forming an angle β with the element side between Nodes 1 and 2, i.e., the vector l_{21}. The direction cosines of this unit vector are represented by the symbols (g_x, g_y, g_z). Since $\boldsymbol{g}$ is the unit vector, its direction cosines will satisfy the equation

$$g_x^2 + g_y^2 + g_z^2 = 1 \tag{9.2}$$



(a) Element and Body Coordinates



(b) Construction of Element Coordinates

Figure 9.1. Construction of local element coordinate system.

Next a unit vector $\boldsymbol{g}$, see Figure 9.1(b), is assumed to be in the plane of the triangular element with its origin at Node 1 and forming an angle β with the element side between Nodes 1 and 2, i.e., the vector l_{21}. The direction cosines of this unit vector are represented by the symbols (g_x, g_y, g_z). Since $\boldsymbol{g}$ is the unit vector, its direction cosines will satisfy the equation

$$g_x^2 + g_y^2 + g_z^2 = 1 \tag{9.2}$$

Also, since $\boldsymbol{g}$ and e_3 are orthogonal unit vectors, their vector dot product must satisfy the equation

$$e_{3x} g_x + e_{3y} g_y + e_{3z} g_z = 0 \tag{9.3}$$

In addition, the vector dot product of the co-planar unit vectors $\boldsymbol{g}$ and l_{21} satisfies the equation

$$I_{21xgx} + I_{21ygy} + I_{21zgz} = \cos \beta \tag{9.4}$$

where $(l_{21x}, + l_{21y}, + l_{21z})$ are the direction cosines of l_{21}.

Solving this system of three simultaneous equation, i.e., Equation (9.2), (9.3), and (9.4), for the direction cosines of the unit vector g yields

$$\begin{aligned} g_x &= l_{21x} \cos \beta + (e_{3y} l_{21z} - e_{3z} l_{21y}) \sin \beta \\ g_y &= l_{21y} \cos \beta + (e_{3z} l_{21x} - e_{3x} l_{21z}) \sin \beta \\ g_z &= l_{21z} \cos \beta + (e_{3x} l_{21y} - e_{3y} l_{21x}) \sin \beta \end{aligned} \tag{9.5}$$

These equations provide the direction cosines for any vector in the plane of the triangular element that is oriented at an angle β from the element side between Nodes 1 and 2. Thus the unit vector components of e_1, and e_2 are obtained by setting $\beta = \alpha/2$ and $\beta = (\pi + \alpha)/2$ in Equation (9.5), respectively. The angle α is obtained from the vector dot product of the unit vectors l_{21} and l_{31},

$$\cos \alpha = l_{21} \cdot l_{31} \tag{9.6}$$

9.2 Displacement Interpolation

As with the other large displacement and small deformation co-rotational element formulations, the nodal displacements are separated into rigid body and deformation displacements,

$$\mathbf{u} = \mathbf{u}^{\text{rigid}} + \mathbf{u}^{\text{def}} \tag{9.7}$$

where the rigid body displacements are defined by the motion of the local element coordinate system, i.e., the co-rotational coordinates, and the deformation displacement are defined with respect to the co-rotational coordinates. The deformation displacement are defined by

$$\begin{Bmatrix} \hat{u}_x \\ \hat{u}_y \\ --- \\ \hat{u}_z \end{Bmatrix}^{def} = \begin{bmatrix} \phi_x^m \\ \phi_y^m \\ --- \\ \phi_z^f \end{bmatrix} \begin{Bmatrix} \delta \\ --- \\ \hat{\theta} \end{Bmatrix} \tag{9.8}$$

where

$$\{\delta\}^T = \{\delta_{12}\ \delta_{23}\ \delta_{31}\} \tag{9.9}$$

are the edge elongations and

$$\{\hat{\theta}\} = \{\hat{\theta}_{1x}\hat{\theta}_{1y}\hat{\theta}_{2x}\hat{\theta}_{2y}\hat{\theta}_{3x}\hat{\theta}_{3y}\} \tag{9.10}$$

are the local nodal rotation with respect to the co-rotational coordinates.

The matrices ϕ_x^m, ϕ_y^m and ϕ_z^f are the membrane and flexural interpolation functions, respectively. The element's membrane deformation is defined in terms of the edge elongations. Marchertas and Belytschko adapted this idea from Argyris et al., [1964], where incremental displacements are used, by modifying the relations for total displacements,

$$\delta_{ij} = \frac{2\left(x_{ji}u_{jix} + y_{ji}u_{jiy} + z_{ji}u_{jiz}\right) + u_{jix}^2 + u_{jiy}^2 + u_{jiz}^2}{l_{ij}^0 + l_{ij}} \tag{9.11}$$

where $x_{ji} = x_j - x_i$, etc.

The non-conforming shape functions used for interpolating the flexural deformations, ϕ_z^f were originally derived by Bazeley, Cheung, Irons, and Zienkiewicz [1965]; hence the LS-DYNA reference to the BCIZ element. Explicit expressions for ϕ_z^f are quite tedious and are not given here. The interested reader is referred to Appendix G in the original work of Marchertas and Belytschko [1974].

The local nodal rotations, which are interpolated by these flexural shape functions, are defined in a manner similar to those used in the Belytschko beam element. The current components of the original element normal are obtained from the relation

$$e_3^0 = \mu^T \lambda \bar{e}_3^0 \tag{9.12}$$

where μ and λ are the current transformations between the global coordinate system and the element (local) and body coordinate system, respectively. The vector $\bar{e}_3^0$ is the original element

unit normal expressed in the body coordinate system. The vector cross product between this current-original unit normal and the current unit normal,

$$e_3 \times e_3^0 = \hat{\theta}_x e_1 + \hat{\theta}_y e_2 \tag{9.13}$$

define the local nodal rotations as

$$\hat{\theta}_x = -\hat{e}_{3y}^0 \tag{9.14}$$

$$\hat{\theta}_y = \hat{e}_{3x}^0 \tag{9.15}$$

Note that at each node the corresponding λ transformation matrix is used in Equation (9.12).

9.3 Strain-Displacement Relations

Marchertas-Belytschko impose the usual Kirchhoff assumptions that normals to the midplane of the element remain straight and normal, to obtain

$$e_{xx} = \frac{\partial u_x}{\partial x} - z\frac{\partial^2 u_z}{\partial x^2} \tag{9.16a}$$

$$e_{yy} = \frac{\partial u_y}{\partial y} - z\frac{\partial^2 u_z}{\partial y^2} \tag{9.16b}$$

$$2e_{xy} = \frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} - 2z\frac{\partial^2 u_z}{\partial x \partial y} \tag{9.16c}$$

where it is understood that all quantities refer to the local element coordinate system.

Substitution of Equations (9.8) into the above strain-displacement relations yields

$$\{\varepsilon\} = \left[E^m\right]\{\delta\} - z\left[E^f\right]\{\hat{\theta}\} \tag{9.17}$$

where

$$\{\varepsilon\}^t = \{\varepsilon_{xx}, \varepsilon_{yy}, 2\varepsilon_{xy}\} \tag{9.18}$$

with

$$\left[E^m\right] = \begin{bmatrix} \dfrac{\partial \phi_{xi}^m}{\partial x} \\ \dfrac{\partial \phi_{yi}^m}{\partial y} \\ \dfrac{\partial \phi_{xi}^m}{\partial y} + \dfrac{\partial \phi_{yi}^m}{\partial x} \end{bmatrix} \tag{9.19a}$$

and

$$\left[E^f\right]=\begin{bmatrix}\dfrac{\partial^2\phi_{zi}^f}{\partial x^2}\\ \dfrac{\partial^2\phi_{zi}^f}{\partial y^2}\\ 2\dfrac{\partial^2\phi_{zi}^f}{\partial x\partial y}\end{bmatrix} \tag{9.19b}$$

Again, the interested reader is referred to Appendices F and G in the original work of Marchertas and Belytschko [1974] for explicit expressions of the above two matrices.

9.4 Nodal Force Calculations

The local element forces and moments are found by integrating the local element stresses through the thickness of the element. The local nodal forces are given by

$$\hat{f}=\int\left[E^m\right]^t\hat{\sigma}\,dV \tag{9.20}$$

where

$$\hat{f}^T=\{f_{12}\,f_{23}\,f_{31}\} \tag{9.21}$$

$$\hat{\sigma}^T=\{\sigma_{xx}\sigma_{yy}\sigma_{xy}\} \tag{9.22}$$

where the side forces and stresses are understood to all be in the local convected coordinate system.

Similarly, the local moments are given by

$$\hat{m}=-\int z\left[E^f\right]^t\hat{\sigma}\,dV \tag{9.23}$$

where

$$\hat{m}^T=\{\hat{m}_{1x}\hat{m}_{1y}\hat{m}_{2x}\hat{m}_{2y}\hat{m}_{3x}\hat{m}_{3y}\} \tag{9.24}$$

The through-the-thickness integration portions of the above local force and moment integrals are usually performed with a 3- or 5-point trapezoidal integration. A three-point inplane integration is also used; it is inpart this three-point inplane integration that increases the operation count for this element over the C^0 shell, which used one-point inplane integration with hourglass stabilization.

The remaining transverse nodal forces are obtained from element equilibrium considerations. Moment equilibrium requires

$$\begin{Bmatrix}\hat{f}_{2z}\\ \hat{f}_{3z}\end{Bmatrix}=\frac{1}{2A}\begin{bmatrix}-\hat{x}_3 & \hat{y}_3\\ \hat{x}_2 & -\hat{y}_2\end{bmatrix}\begin{Bmatrix}\hat{m}_{1x}+\hat{m}_{2x}+\hat{m}_{3x}\\ \hat{m}_{1y}+\hat{m}_{2y}+\hat{m}_{3y}\end{Bmatrix} \tag{9.25}$$

where A is the area of the element. Next transverse force equilibrium provides

$$\hat{f}_{1z} = -\hat{f}_{2z} - \hat{f}_{3z} \tag{9.26}$$

The corresponding global components of the nodal forces are obtained from the following transformation

$$\begin{Bmatrix} f_{ix} \\ f_{iy} \\ f_{iz} \end{Bmatrix} = \frac{f_{ij}}{l_{ij}} \begin{Bmatrix} x_{ij} + u_{ijx} \\ y_{ij} + u_{ijy} \\ z_{ij} + u_{ijz} \end{Bmatrix} + \frac{f_{ik}}{l_{ik}} \begin{Bmatrix} x_{ik} + u_{ikx} \\ y_{ik} + u_{iky} \\ z_{ik} + u_{ikz} \end{Bmatrix} + \hat{f}_{iz} \begin{Bmatrix} e_{3x} \\ e_{3y} \\ e_{3z} \end{Bmatrix} \tag{9.27}$$

Finally, the local moments are transformed to the body coordinates using the relation

$$\begin{Bmatrix} \bar{m}_{ix} \\ \bar{m}_{iy} \\ \bar{m}_{iz} \end{Bmatrix} = \lambda^T \mu \begin{Bmatrix} \hat{m}_{ix} \\ \hat{m}_{iy} \\ \hat{m}_{iz} \end{Bmatrix} \tag{9.28}$$

10. HUGHES-LIU SHELL

The Hughes-Liu shell element formulation ([Hughes and Liu 1981a, b], [Hughes et al., 1981], [Hallquist et al., 1985]) was the first shell element implemented in LS-DYNA. It was selected from among a substantial body of shell element literature because the element formulation has several desirable qualities:

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

The remainder of this section reviews the Hughes-Liu shell element (referred to by Hughes and Liu as the U1 element) which is a four-node shell with uniformly reduced integration, and summarizes the modifications to their theory as it is implemented in LS-DYNA. A detailed discussion of these modifications, as well as those associated with the implementation of the Hughes-Liu shell element in NIKE3D, are presented in an article by Hallquist and Benson [1986].

10.1 Geometry

The Hughes-Liu shell element is based on a degeneration of the standard 8-node brick element formulation, an approach originated by Ahmad et al. [1970]. Recall from the discussion of the solid elements the isoparametric mapping of the biunit cube:

$$x(\xi,\eta,\zeta) = N_a(\xi,\eta,\zeta)x_a \tag{10.1}$$

$$N_a(\xi,\eta,\zeta) = \frac{(1+\xi_a\xi)(1+\eta_a\eta)(1+\zeta_a\zeta)}{8} \tag{10.2}$$

where x is an arbitrary point in the element, (ξ, η, ζ) are the parametric coordinates, x_a are the global nodal coordinates of node a, and N_a are the element shape functions evaluated at node a, i.e., (ξ_a, η_a, ζ_a) are (ξ, η, ζ) evaluated at node a.

In the shell geometry, planes of constant ζ will define the lamina or layers of the shell and fibers are defined by through-the-thickness lines when both ξ and η are constant (usually only defined at the nodes and thus referred to as 'nodal fibers'). To degenerate the 8-node brick geometry into the 4-node shell geometry, the nodal pairs in the ζ direction (through the shell thickness) are combined into a single node, for the translation degrees of freedom, and an

inextensible nodal fiber for the rotational degrees of freedom. Figure 10.1 shows a schematic of the bi-unit cube and the shell element.

The mapping of the bi-unit cube into the shell element is separated into two parts

$$\mathrm{x}(\xi,\eta,\zeta) = \bar{\mathrm{x}}(\xi,\eta) + \mathrm{X}(\xi,\eta,\zeta) \tag{10.3}$$

where $\bar{\mathrm{x}}$ denotes a position vector to a point on the reference surface of the shell and X is a position vector, based at point $\bar{\mathrm{x}}$ on the reference, that defines the fiber direction through that point. In particular, if we consider one of the four nodes which define the reference surface, then

$$\bar{\mathrm{x}}(\xi,\eta) = N_a(\xi,\eta)\bar{\mathrm{x}}_a \tag{10.4}$$

$$X(\xi,\eta,\zeta) = N_a(\xi,\eta)\mathrm{X}_a(\zeta) \tag{10.5}$$

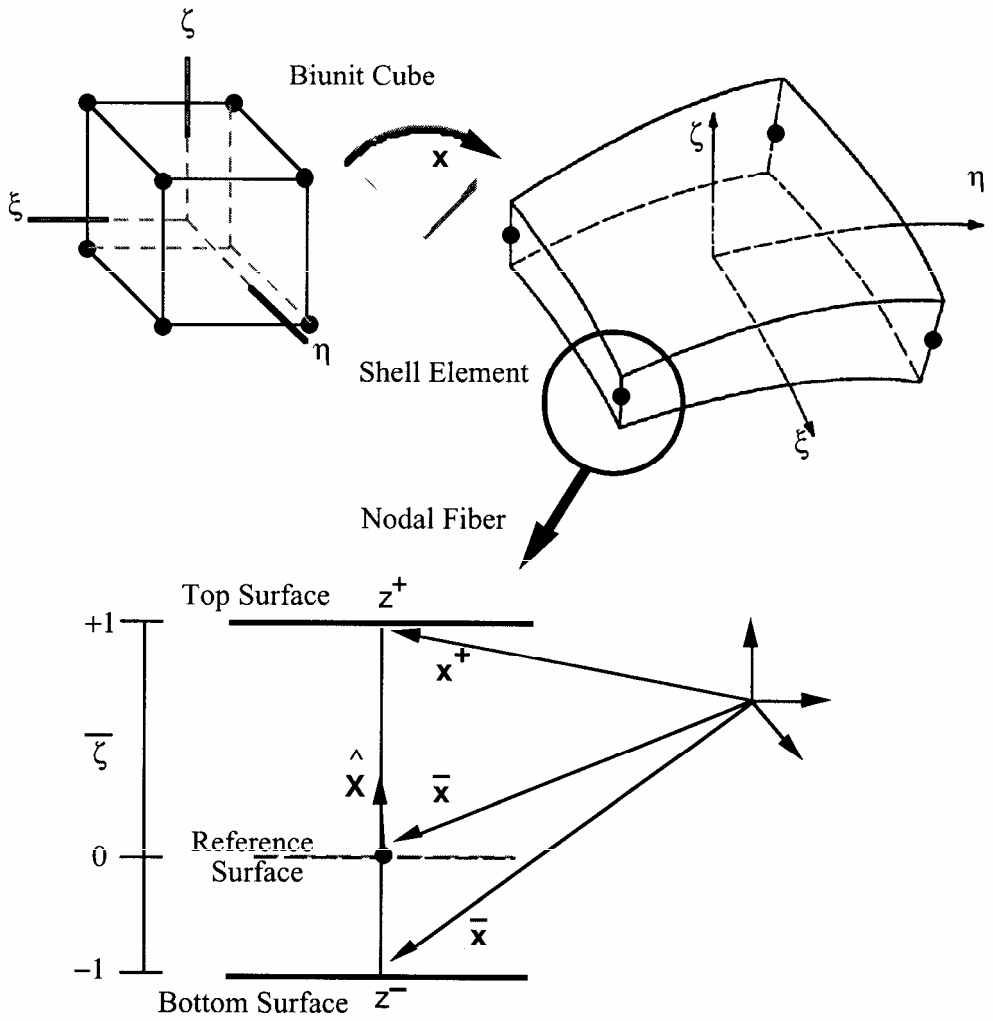


Figure 10.1. Mapping of the biunit cube into the Hughes-Liu shell element and nodal fiber nomenclature.

With this description, arbitrary points on the reference surface $\bar{\mathrm{x}}$ are interpolated by the two-dimensional shape function $N(\xi,\eta)$ operating on the global position of the four shell nodes that define the reference surfaces, i.e., $\bar{\mathrm{x}}_a$. Points off the reference surface are further interpolated by using a one-dimensional shape function along the fiber direction, i.e., $\mathrm{X}_a(\zeta)$, where

$$X_a(\zeta) = z_a(\zeta)\hat{X}_a \tag{10.6a}$$

$$z_a(\zeta) = N_+(\zeta)z_a^+ + N_-(\zeta)z_a^- \tag{10.6b}$$

$$N_+(\zeta) = \frac{(1+\zeta)}{2} \tag{10.6c}$$

$$N_-(\zeta) = \frac{(1-\zeta)}{2} \tag{10.6d}$$

As shown in the lower portion of Figure 10.1, $\hat{X}_a$ is a unit vector in the fiber direction and $z(\zeta)$ is a thickness function. (Thickness changes (see [Hughes and Carnoy 1981]) are accounted for by explicitly adjusting the fiber lengths at the completion of a time step based on the amount of straining in the fiber direction. Updates of the fiber lengths always lag one time step behind other kinematical quantities.)

The reference surface may be located at the mid-surface of the shell or at either of the shell's outer surfaces. This capability is useful in several practical situations involving contact surfaces, connection of shell elements to solid elements, and offsetting elements such as stiffeners in stiffened shells. The reference surface is located within the shell element by specifying the value of the parameter $\bar{\zeta}$ (see lower portion of Figure 10.1). When $\bar{\zeta}$ = –1, 0, +1, the reference surface is located at the bottom, middle, and top surface of the shell, respectively.

The Hughes-Liu formulation uses two position vectors, in addition to $\bar{\zeta}$, to locate the reference surface and define the initial fiber direction. The two position vectors x_a^+ and x_a^- are located on the top and bottom surfaces, respectively, at node a. From these data the following are obtained:

$$\bar{x}_a = \frac{1}{2}\left(1-\bar{\zeta}\right)x_a^- + \left(1+\bar{\zeta}\right)x_a^+ \tag{10.7a}$$

$$\hat{X}_a = \frac{\left(x_a^+ - x_a^-\right)}{h_a} \tag{10.7b}$$

$$z_a^+ = \frac{1}{2}\left(1-\bar{\zeta}\right)h_a \tag{10.7c}$$

$$z_a^- = -\frac{1}{2}\left(1+\bar{\zeta}\right)h_a \tag{10.7d}$$

$$h_a = \left\| x_a^+ - x_a^- \right\| \tag{10.7e}$$

where $\|\cdot\|$ is the Euclidean norm.

10.2 Kinematics

The same parametric representation used to describe the geometry of the shell element, i.e., reference surface and fiber vector interpolation, are used to interpolate the shell element displacement, i.e., an isoparametric representation. Again, the displacements are separated into the reference surface displacements and rotations associated with the fiber direction:

$$u(\xi,\eta,\zeta) = \bar{u}(\xi,\eta) + U(\xi,\eta,\zeta) \tag{10.8a}$$

$$\bar{u}(\xi,\eta) = N_a(\xi,\eta)\bar{u}_a \tag{10.8b}$$

$$U(\xi,\eta,\zeta) = N_a(\xi,\eta)U_a(\zeta) \tag{10.8c}$$

$$U_a(\zeta) = z_a(\zeta)\hat{U}_a \tag{10.8d}$$

where $\boldsymbol{u}$ is the displacement of a generic point; $\bar{u}$ is the displacement of a point on the reference surface, and $\boldsymbol{U}$ is the 'fiber displacement' rotations; the motion of the fibers can be interpreted as either displacements or rotations as will be discussed.

Hughes and Liu introduce the notation that follows, and the associated schematic shown in Figure 10.2, to describe the current deformed configuration with respect to the reference configuration:

$$y = \bar{y} + Y \tag{10.9a}$$

$$\bar{y} = \bar{x} + \bar{u} \tag{10.9b}$$

$$\bar{y}_a = \bar{x}_a + \bar{u}_a \tag{10.9c}$$

$$Y = X + U \tag{10.9d}$$

$$Y_a = X_a + U_a \tag{10.9e}$$

$$\hat{Y}_a = \hat{X}_a + \hat{U}_a \tag{10.9f}$$

In the above relations, and in Figure 10.2, the x quantities refer to the reference configuration, the y quantities refer to the updated (deformed) configuration and the u quantities are the displacements. The notation consistently uses a superscript bar $(\bar{\cdot})$ to indicate reference surface

quantities, a superscript caret $(\hat{\ })$ to indicate unit vector quantities, lower case letters for translational displacements, and upper case letters indicating fiber displacements. To update to the deformed configuration, two vector quantities are needed: the reference surface displacement $\bar{u}$ and the associated nodal fiber displacement $\boldsymbol{U}$. The nodal fiber displacements are defined in the fiber coordinate system, described in the next subsection.

10.2.1 Fiber Coordinate System

For a shell element with four nodes, the known quantities will be the displacements of the reference surface $\bar{u}$ obtained from the translational equations of motion and some rotational quantities at each node obtained from the rotational equations of motion. To complete the kinematics, we now need a relation between nodal rotations and fiber displacements $\boldsymbol{U}$.

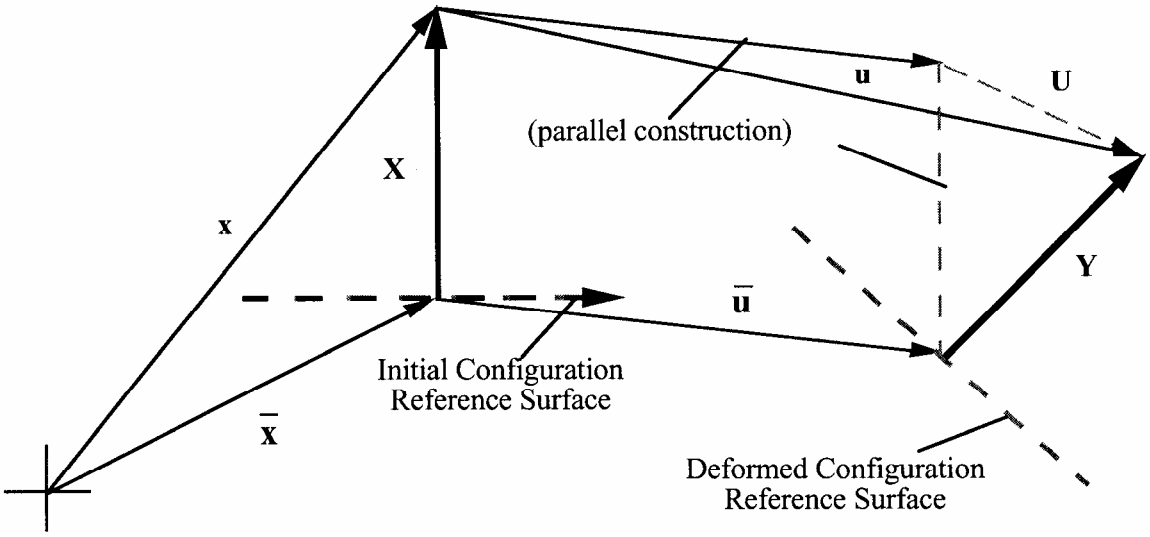


Figure 10.2. Schematic of deformed configuration displacements and position vectors.

At each node a unique local Cartesian coordinate system is constructed that is used as the reference frame for the rotation increments. The relation presented by Hughes and Liu for the nodal fiber displacements (rotations) is an incremental relation, i.e., it relates the current configuration to the last state, not to the initial configuration. Figure 10.3 shows two triads of unit vectors: $\left(b_1^f, b_2^f, b_3^f\right)$ comprising the orthonormal fiber basis in the reference configuration (where the fiber unit vector is now $\hat{Y} = b_3^f$) and (b_1, b_2, b_3) indicating the incrementally updated current configuration of the fiber vectors. The reference triad is updated by applying the incremental rotations, $\Delta\theta 1$ and $\Delta\theta 2$, obtained from the rotational equations of motion, to the fiber vectors $\left(b_1^f \text{ and } b_2^f\right)$ as shown in Figure 10.3. The linearized relationship between the components of $\Delta\hat{U}$ in the fiber system viz, $\Delta\hat{U}_1^f, \Delta\hat{U}_2^f, \Delta\hat{U}_3^f$, and the incremental rotations is given by

$$\begin{Bmatrix} \Delta\hat{U}_1^f \\ \Delta\hat{U}_2^f \\ \Delta\hat{U}_3^f \end{Bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \Delta\theta_1 \\ \Delta\theta_2 \end{Bmatrix} \tag{10.10}$$

Although the above Hughes-Liu relation for updating the fiber vector enables a reduction in the number of nodal degrees of freedom from six to five, it is not implemented in LS-DYNA because it is not applicable to beam elements.

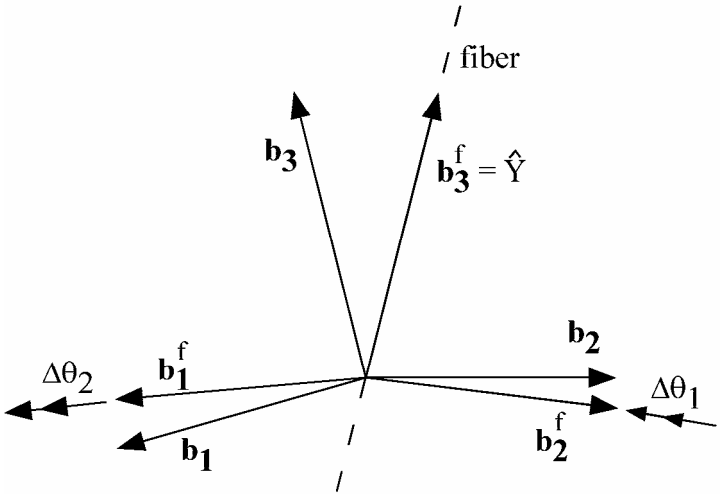


Figure 10.3. Incremental update of fiber vectors using Hughes-Liu incremental rotations.

In LS-DYNA, three rotational increments are used, defined with reference to the global coordinate axes:

$$\begin{Bmatrix} \Delta\hat{U}_1 \\ \Delta\hat{U}_2 \\ \Delta\hat{U}_3 \end{Bmatrix} = \begin{bmatrix} 0 & \hat{Y}_3 & -\hat{Y}_2 \\ -\hat{Y}_3 & 0 & \hat{Y}_1 \\ \hat{Y}_2 & -\hat{Y}_1 & 0 \end{bmatrix} \begin{Bmatrix} \Delta\theta_1 \\ \Delta\theta_2 \\ \Delta\theta_3 \end{Bmatrix} \tag{10.11}$$

Equation (10.11) is adequate for updating the stiffness matrix, but for finite rotations the error is significant. A more accurate second-order technique is used in LS-DYNA for updating the unit fiber vectors:

$$\hat{Y}_i^{n+1} = R_{ij}(\Delta\theta)\hat{Y}_i^n \tag{10.12}$$

where

$$R_{ij}(\Delta\theta) = \delta_{ij} + \frac{1}{2}\frac{(2\delta_{ij} + \Delta S_{ik})\Delta S_{ik}}{D} \tag{10.13a}$$

$$\Delta S_{ij} = e_{ikj}\Delta\theta k \tag{10.13b}$$

$$2D = 2 + \frac{1}{2}\left(\Delta\theta_1^2 + \Delta\theta_2^2 + \Delta\theta_3^2\right) \tag{10.13c}$$

Here, δ_{ij} is the Kronecker delta and e_{ijk} is the permutation tensor. This rotational update is often referred to as the Hughes-Winget formula [Hughes and Winget 1980]. An exact rotational update using Euler angles or Euler parameters could easily be substituted in Equation (10.12), but it is doubtful that the extra effort would be justified.

10.2.2 Lamina Coordinate System

In addition to the above described fiber coordinate system, a local lamina coordinate system is needed to enforce the zero normal stress condition, i.e., plane stress. Lamina are layers through the thickness of the shell that correspond to the locations and associated thicknesses of the through-the-thickness shell integration points; the analogy is that of lamina in a fibrous composite material. The orthonormal lamina basis (Figure 10.4), with one direction $\hat{e}_3$ normal to the lamina of the shell, is constructed at every integration point in the shell.

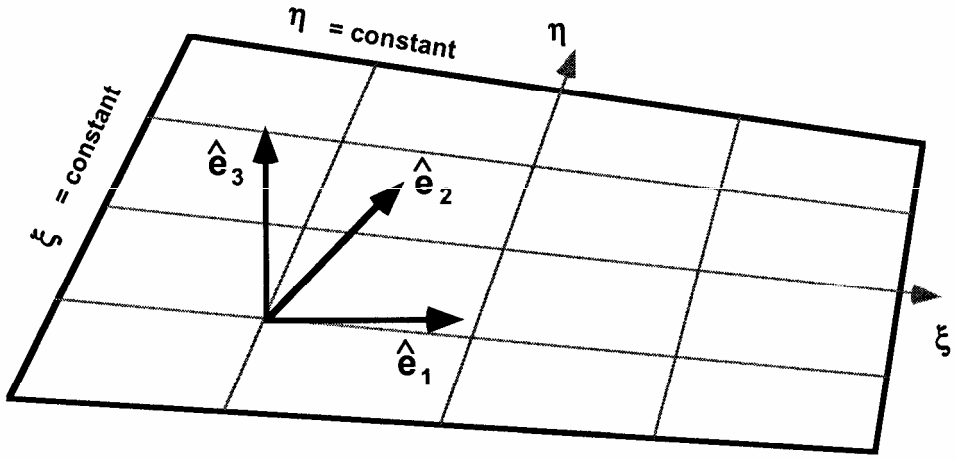


Figure 10.4. Schematic of lamina coordinate unit vectors.

The lamina basis is constructed by forming two unit vectors locally tangent to the lamina:

$$\hat{e}_1 = \frac{y_{,\xi}}{\left\| y_{,\xi} \right\|} \tag{10.14}$$

$$e_2' = \frac{y_{,\eta}}{\left\| y_{,\eta} \right\|} \tag{10.15}$$

where, as before, y is the position vector in the current configuration. The normal to the lamina at the integration point is constructed from the vector cross product of these local tangents:

$$\hat{e}_3 = \hat{e}_1 \times e_2' \tag{10.16}$$

To complete this orthonormal lamina basis, the vector

$$\hat{e}_2 = \hat{e}_3 \times \hat{e}_1 \tag{10.17}$$

is defined, because $\hat{e}_2$, although tangent to both the lamina and lines of constant ξ, may not be normal to $\hat{e}_1$ and $\hat{e}_3$. The lamina coordinate system rotates rigidly with the element.

The transformation of vectors from the global to lamina coordinate system can now be defined in terms of the lamina basis vectors as

$$\hat{A} = \begin{Bmatrix} \hat{A}_x \\ \hat{A}_y \\ \hat{A}_z \end{Bmatrix} = \begin{bmatrix} e_{1x} & e_{2x} & e_{3x} \\ e_{1y} & e_{2y} & e_{3y} \\ e_{1z} & e_{2z} & e_{3z} \end{bmatrix}^T \begin{Bmatrix} A_x \\ A_y \\ A_z \end{Bmatrix} = [q]\{A\} \tag{10.18}$$

where e_{ix}, e_{iy}, e_{iz} are the global components of the lamina coordinate unit vectors; $\hat{A}$ is a vector in the lamina coordinates, and A is the same vector in the global coordinate system.

10.3 Strains and Stress Update

10.3.1 Incremental Strain and Spin Tensors

The strain and spin increments are calculated from the incremental displacement gradient

$$G_{ij} = \frac{\partial \Delta u_i}{\partial y_j} \tag{10.19}$$

where Δu_i are the incremental displacements and y_j are the deformed coordinates. The incremental strain and spin tensors are defined as the symmetric and skew-symmetric parts, respectively, of G_{ij}:

$$\Delta \varepsilon_{ij} = \frac{1}{2}\left(G_{ij} + G_{ji}\right) \tag{10.20}$$

$$\Delta \omega_{ij} = \frac{1}{2}\left(G_{ij} - G_{ji}\right) \tag{10.21}$$

The incremental spin tensor $\Delta\omega_{ij}$ is used as an approximation to the rotational contribution of the Jaumann rate of the stress tensor; LS-DYNA implicit uses the more accurate Hughes-Winget transformation matrix (Equation (10.12)) with the incremental spin tensor for the rotational update. The Jaumann rate update is approximated as:

$$\underline{\sigma}_{ij}^{n+1} = \sigma_{ij}^n + \sigma_{ip}^n \Delta\omega_{pj} + \sigma_{jp}^n \Delta\omega_{pi} \tag{10.22}$$

where the superscripts on the stress refer to the updated $(n+1)$ and reference (n) configurations. The Jaumann rate update of the stress tensor is applied in the global configuration before the constitutive evaluation is performed. In the Hughes-Liu shell the stresses and history variables are stored in the global coordinate system.

10.3.2 Stress Update

To evaluate the constitutive relation, the stresses and strain increments are rotated from the global to the lamina coordinate system using the transformation defined previously in Equation (10.18), viz.

$$\underline{\sigma}_{ij}^{l^{n+1}} = q_{ik}\,\underline{\sigma}_{kn}^{n+1} q_{jn} \tag{10.23}$$

$$\Delta\varepsilon_{ij}^{l^{n+1/2}} = q_{ik}\Delta\varepsilon_{kn}^{n+1/2} q_{jn} \tag{10.24}$$

where the superscript l indicates components in the lamina (local) coordinate system.

The stress is updated incrementally:

$$\sigma_{ij}^{l^{n+1}} = \underline{\sigma}_{ij}^{l^{n+1}} + \Delta\sigma_{ij}^{l^{n+1/2}} \tag{10.25}$$

and rotated back to the global system:

$$\sigma_{ij}^{n+1} = q_{ki}\sigma_{kn}^{l^{n+1}} q_{nj} \tag{10.26}$$

before computing the internal force vector.

10.3.3 Incremental Strain-Displacement Relations

The global stresses are now used to update the internal force vector

$$f_a^{\text{int}} = \int T_a^T B_a^T \sigma d\upsilon \tag{10.27}$$

where f_a^{int} are the internal forces at node a, B_a is the strain-displacement matrix in the lamina coordinate system associated with the displacements at node a, and T_a is the transformation matrix relating the global and lamina components of the strain-displacement matrix. Because the $\boldsymbol{B}$ matrix relates six strain components to twenty-four displacements (six degrees of freedom at four nodes), it is convenient to partition the $\boldsymbol{B}$ matrix into four groups of six:

$$B = [B1,\ B2,\ B3,\ B4] \tag{10.28a}$$

Each $\boldsymbol{B}_a$ submatrix is further partitioned into a portion due to strain and spin:

$$B_a = \begin{bmatrix} B_a^{\varepsilon} \\ B_a^{\omega} \end{bmatrix} \tag{10.28b}$$

with the following submatrix definitions:

$$B_a^{\varepsilon} = \begin{bmatrix} B_1 & 0 & 0 & B_4 & 0 & 0 \\ 0 & B_2 & 0 & 0 & B_5 & 0 \\ \bar{B}_2 & \bar{B}_1 & 0 & \bar{B}_5 & \bar{B}_4 & 0 \\ 0 & \bar{B}_3 & \bar{B}_2 & 0 & \bar{B}_6 & \bar{B}_5 \\ \bar{B}_3 & 0 & \bar{B}_1 & \bar{B}_6 & 0 & \bar{B}_4 \end{bmatrix} \tag{10.28c}$$

$$B_a^{\omega} = \begin{bmatrix} \bar{B}_2 & -\bar{B}_1 & 0 & \bar{B}_5 & -\bar{B}_4 & 0 \\ 0 & \bar{B}_3 & -\bar{B}_2 & 0 & \bar{B}_6 & -\bar{B}_5 \\ -\bar{B}_3 & 0 & \bar{B}_1 & -\bar{B}_6 & 0 & \bar{B}_4 \end{bmatrix} \tag{10.28d}$$

where

$$B_i = \begin{cases} N_{a,i} = \dfrac{\partial N_a}{\partial y_i^l} & \text{for } i = 1,2,3 \\ \left(N_a z_a\right)_{,i-3} = \dfrac{\partial \left(N_a z_a\right)}{\partial y_{i-3}^l} & \text{for } i = 4,5,6 \end{cases}$$

Notes on strain-displacement relations:

- The derivatives of the shape functions are taken with respect to the lamina coordinate system, e.g., $y = qy$.
- The superscript bar indicates the B's are evaluated at the center of the lamina $(0, 0, \zeta)$. The strain-displacement matrix uses the 'B-Bar' $\left(\bar{B}\right)$ approach advocated by Hughes [1980]. In the NIKE3D and DYNA3D implementations, this entails replacing certain rows of the $\boldsymbol{B}$ matrix and the strain increments with their counterparts evaluated at the center of the element. In particular, the strain-displacement matrix is modified to produce constant shear and spin increments throughout the lamina.
- The resulting $\boldsymbol{B}$-matrix is a 8×24 matrix. Although there are six strain and three rotations increments, the $\boldsymbol{B}$ matrix has been modified to account for the fact that σ_{33} will be zero in the integration of Equation (10.27).

10.4 Element Mass Matrix

Hughes, Liu, and Levit [Hughes et al., 1981] describe the procedure used to form the shell element mass matrix in problems involving explicit transient dynamics. Their procedure, which scales the rotary mass terms, is used for all shell elements in LS-DYNA including those formulated by Belytschko and his co-workers. This scaling permits large critical time step sizes without loss of stability.

The consistent mass matrix is defined by

$$M = \int_{v_m} \rho \, \mathrm{N}^t \mathrm{N} \, dv_m \tag{10.29}$$

but cannot be used effectively in explicit calculations where matrix inversions are not feasible. In LS-DYNA only three and four-node shell elements are used with linear interpolation functions; consequently, we compute the translational masses from the consistent mass matrix by row summing, leading to the following mass at element node *a*:

$$M_{disp_a} = \int_v \rho \phi_a \, dv \tag{10.30}$$

The rotational masses are computed by scaling the translational mass at the node by the factor α:

$$M_{rot_a} = \propto M_{disp_a} \tag{10.31}$$

where

$$\propto = \max\{\propto_1, \propto_2\} \tag{10.32}$$

$$\propto_1 = \langle z_a \rangle^2 + \tfrac{1}{12}[z_a]^2 \tag{10.33}$$

$$\propto_2 = \frac{V}{8h} \tag{10.34}$$

$$\langle z_a \rangle = \frac{\left(z_a^+ + z_a^-\right)}{2} \tag{10.35}$$

$$[z_a] = z_a^+ - z_a^- \tag{10.36}$$

and V and h are the volume and the thickness of the element, respectively.

10.5 Accounting for Thickness Changes

Hughes and Carnoy [1981] describe the procedure used to update the shell thickness due to large membrane stretching. Their procedure with any necessary modifications is used across all shell element types in LS-DYNA. One key to updating the thickness is an accurate calculation of the normal strain component $\Delta\varepsilon_{33}$. This strain component is easily obtained for elastic materials but can require an iterative algorithm for nonlinear material behavior. In LS-DYNA we therefore default to an iterative plasticity update to accurately determine $\Delta\varepsilon_{33}$.

Hughes and Carnoy integrate the strain tensor through the thickness of the shell in order to determine a mean value $\Delta\bar{\varepsilon}_{ij}$:

$$\Delta\bar{\varepsilon}_{ij} = \tfrac{1}{2}\int_{-1}^{1} \Delta\varepsilon_{ij} d\zeta \tag{10.37}$$

and then project it to determine the straining in the fiber direction:

$$\bar{\varepsilon}^{\,f} = \hat{Y}^T \Delta\bar{\varepsilon}_{ij} \hat{Y} \tag{10.38}$$

Using the interpolation functions through the integration points the strains in the fiber directions are extrapolated to the nodal points if 2×2 selectively reduced integration is employed. The nodal fiber lengths can now be updated:

$$h_a^{n+1} = h_a^n \left(1 + \bar{\varepsilon}_a^{\,f}\right) \tag{10.39}$$

10.6 Fully Integrated Hughes-Liu Shells

It is well known that one-point integration results in zero energy modes that must be resisted. The four-node under integrated shell with six degrees of freedom per node has nine zero energy modes, six rigid body modes, and four unconstrained drilling degrees of freedom. Deformations in the zero energy modes are always troublesome but usually not a serious problem except in regions where boundary conditions such as point loads are active. In areas where the zero energy modes are a problem, it is highly desirable to provide the option of using the original formulation of Hughes-Liu with selectively reduced integration.

The major disadvantages of full integration are two-fold:

- nearly four times as much data must be stored;
- the operation count increases three- to fourfold. The level 3 loop is added as shown in Figure 10.6.

However, these disadvantages can be more than offset by the increased reliability and accuracy.

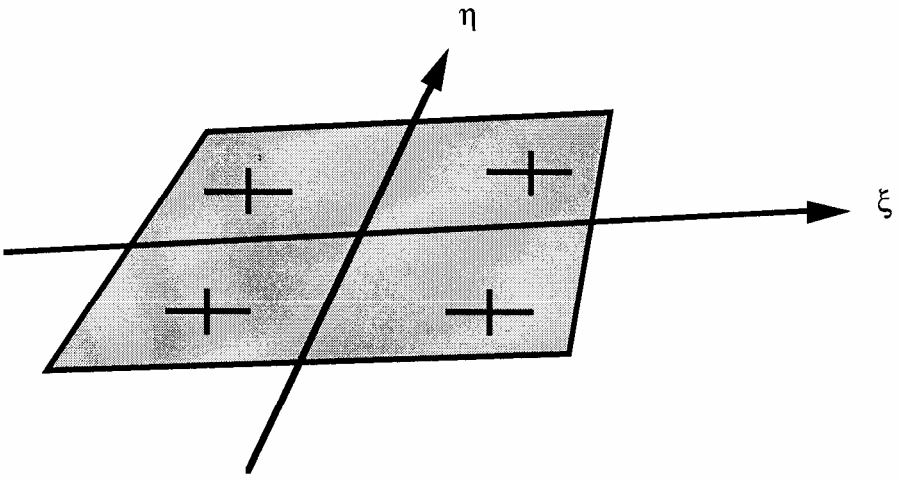


Figure 10.5. Selectively reduced integration rule results in four inplane points being used.

We have implemented two version of the Hughes-Liu shell with selectively reduced integration. The first closely follows the intent of the original paper, and therefore no assumptions are made to reduce costs, which are outlined in operation counts in Table 10.1.

These operation counts can be compared with those in Table 10.2 for the Hughes-Liu shell with uniformly reduced integration. The second formulation, which reduces the number of operation by more than a factor of two, is referred to as the co-rotational Hughes-Liu shell in the LS-DYNA user's manual. This shell is considerably cheaper due to the following simplifications:

- Strains rates are not centered. The strain displacement matrix is only computed at time n+1 and not at time n+½.
- The stresses are stored in the local shell system following the Belytschko-Tsay shell. The transformations of the stresses between the local and global coordinate systems are thus avoided.
- The Jaumann rate rotation is not performed, thereby avoiding even more computations. This does not necessarily preclude the use of the shell in large deformations.

To study the effects of these simplifying assumptions, we can compare results with those obtained with the full Hughes-Liu shell. Thus far, we have been able to get comparable results.

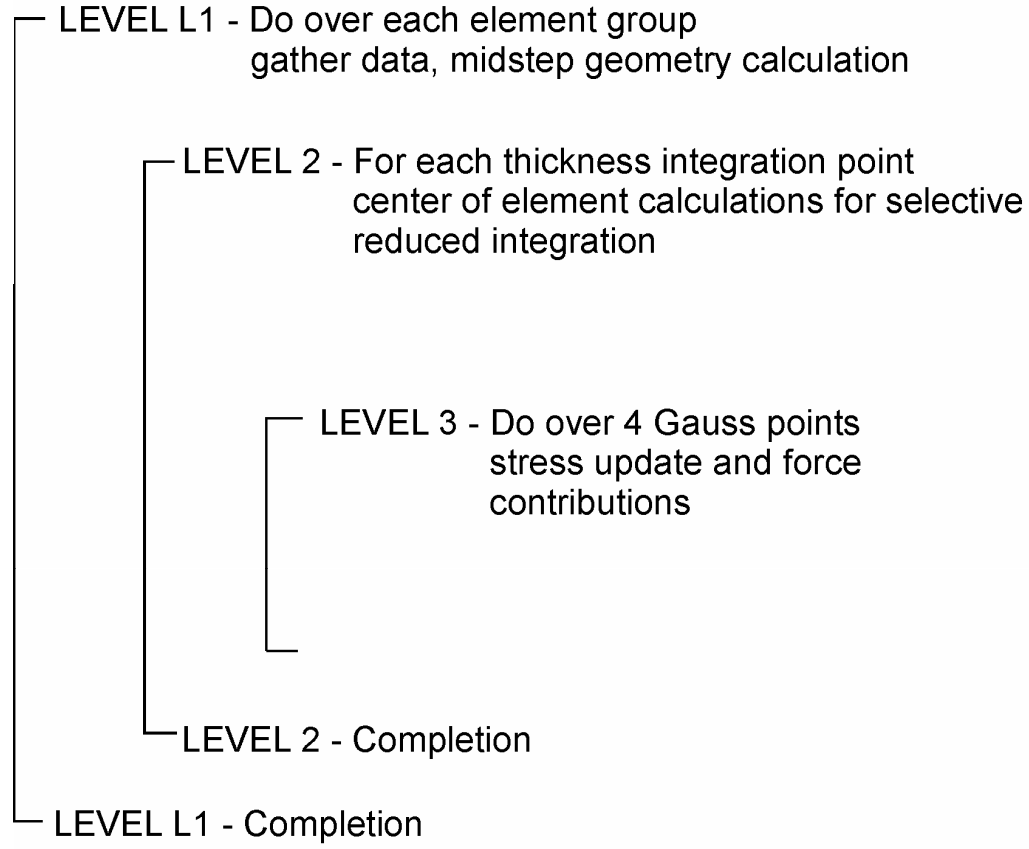


Figure 10.6. An inner loop, LEVEL 3, is added for the Hughes-Liu shell with selectively reduced integration.

LEVEL L1 - Once per element	
Midstep translation geometry, etc.	204
Midstep calculation of $\hat{Y}$	318
LEVEL L2 - For each integration point through thickness (NT points)	
Strain increment at (0, 0, ζ)	316
Hughes-Winget rotation matrix	33
Square root of Hughes-Winget matrix	47
Rotate strain increments into lamina coordinates	66
Calculate rows 3-8 of B matrix	919
LEVEL L3 - For each integration point in lamina	
Rotate stress to n+1/2 configuration	75
Incremental displacement gradient matrix	370
Rotate stress to lamina system	75
Rotate strain increments to lamina system	55
Constitutive model	model dependent
Rotate stress back to global system	69
Rotate stress to n+1 configuration	75
Calculate rows 1 and 2 of B matrix	358
Stresses in n+1 lamina system	75
Stress divergence	245
TOTAL	**522 +NT {1381 +4 * 1397}**

Table 10.1. Operation counts for the Hughes-Liu shell with selectively reduced integration.

LEVEL L1 - Once per element	
Calculate displacement increments	24
Element areas for time step	53
Calculate $\hat{Y}$	238
LEVEL L2 and L3 - Integration point through thickness (NT points)	
Incremental displacement gradient matrix	284
Jaumann rotation for stress	33
Rotate stress into lamina coordinates	75
Rotate stain increments into lamina coordinates	81
Constitutive model	model dependent
Rotate stress to n+1 global coordinates	69
Stress divergence	125
LEVEL L1 - Cleanup	
Finish stress divergence	60
Hourglass control	356
TOTAL	**731 +NT * 667**

Table 10.2. Operation counts for the LS-DYNA implementation of the uniformly reduced Hughes-Liu shell.

11. TRANSVERSE SHEAR TREATMENT FOR LAYERED SHELL

The shell element formulations that include the transverse shear strain components are based on the first order shear deformation theory, which yield constant through thickness transverse shear strains. This violates the condition of zero traction on the top and bottom surfaces of the shell. Normally, this is corrected by the use of a shear correction factor. The shear correction factor is 5/6 for isotropic materials; however, this value is incorrect for sandwich and laminated shells. Not accounting for the correct transverse shear strain and stress could yield a very stiff behavior in sandwich and laminated shells. This problem is addressed here by the use of the equilibrium equations without gradient in the y-direction as described by what follows. Consider the stresses in a layered shell:

$$\begin{aligned}
\sigma_x^{(i)} &= C_{11}^{(i)}(\varepsilon_x^\circ + z\chi_x) + C_{12}^{(i)}(\varepsilon_y^\circ + z\chi_y) = C_{11}^{(i)}\varepsilon_x^\circ + C_{12}^{(i)}\varepsilon_y^\circ + z(C_{11}^{(i)}\chi_x + C_{12}^{(i)}\chi_y) \\
\sigma_y^{(i)} &= C_{12}^{(i)}\varepsilon_x^\circ + C_{22}^{(i)}\varepsilon_y^\circ + z(C_{12}^{(i)}\chi_x + C_{22}^{(i)}\chi_y \\
\tau_{xy}^{(i)} &= C_{44}^{(i)}(\varepsilon_{xy}^\circ + z\chi_{xy})
\end{aligned} \tag{11.1}$$

Assume that the bending center $\overline{z}_x$ is known. Then

$$\sigma_x^{(i)} = (z - \overline{z}_x)\left(C_{11}^{(i)}\chi_\chi + C_{12}^{(i)}\chi_y\right) + C_{11}^{(i)}\varepsilon_x(\overline{z}_x) + C_{12}^{(i)}\varepsilon_y(\overline{z}_x) \tag{11.2}$$

The bending moment is given by the following equation:

$$M_{xx} = \chi_x\left(\sum_{i=1}^{NL} C_{11}^{(i)} \int_{z_{i-1}}^{z_i} z^2 dz\right) + \chi_y\left(\sum_{i=1}^{NL} C_{12}^{(i)} \int_{z_{i-1}}^{z_i} z^2 dz\right)$$

or

$$M_{xx} = \frac{1}{3}\left[\chi_x \sum_{i=1}^{NL} C_{11}^{(i)}(z_i^3 - z_{i-1}^3) + \chi_y \sum_{i=1}^{NL} C_{12}^{(i)}(z_i^3 - z_{i-1}^3)\right] \tag{11.3}$$

where "*NL*" is the number of layers in the material.

Assume $\varepsilon_y = 0$ and $\sigma_x = E_\chi \varepsilon_\chi$, then

$$\varepsilon_x = \frac{z - \overline{z}_x}{\rho} = (z - \overline{z}_x)\chi_x$$

and

$$M_{xx} = \frac{1}{3}(\chi_x \sum\ E_x^{(i)}(z_i^3 - z^3{}_{i-1}))$$

$$\chi_x = \frac{3M_{xx}}{\sum E^{(i)}{}_x (z_i^3 - z_{i-1}^3)}$$

Therefore, the stress becomes

$$\sigma_x^{(i)} = \frac{3M_{xx} E_x^{(i)} (z - \overline{z}_x)}{\sum_{i=1}^{NL} E_x^{(i)} (z_i^3 - z_{i-1}^3)} \tag{11.4}$$

Now considering the first equilibrium equation, one can write the following:

$$\frac{\partial \tau_{xz}}{\partial z} = -\frac{\partial \sigma_x}{\partial x} = -\frac{3Q_{xz} E_x^{(j)} (z - \overline{z}_x)}{\sum_{i=1}^{NL} E_x^{(i)} (z_i^3 - z_{i-1}^3)} \tag{11.5}$$

$$\tau_{xz}^{(j)} = -\frac{3Q_{xz} E_x^{(j)} (\frac{z^2}{2} - z\overline{z}_x)}{\sum_{i=1}^{NL} E_x^{(i)} (z_i^3 - z_{i-1}^3)} + C_j \tag{11.6}$$

where Q_{xz} is the shear force and C_j is the constant of integration. This constant is obtained from the transverse shear stress continuity requirement at the interface of each layer.

let $$(\overline{EI})_x = \sum_{i=1}^{NL} E_x^{(i)} (z_i^3 - z_{i-1}^3)$$

Then

$$C_j = \frac{Q_{xz} E_x^{(i)} \left(\frac{z_{i-1}^2}{2} - z_{i-1}\overline{z}_x \right)}{(\overline{EI})_x} + \tau_{xz}^{i-1}$$

and

$$\tau_{xz}^{(i)} = \tau_{xz}^{(i-1)} + \frac{Q_{xz} E_x^{(i)}}{(\overline{EI})x} \left[\frac{z_{i-1}^2}{2} - z_{i-1}\overline{z}_x - \frac{z^2}{2} + z\overline{z}_x \right] \tag{11.7}$$

For the first layer

$$\tau_{xz} = -\frac{3Q_{xz} C_{11}^{(1)}}{\sum_{i=1}^{NL} C_{11}^{(i)} (z_i^3 + z_{i-1}^3)} \left[\frac{z^2 - z_o^2}{2} - \overline{z}_x (z - z_o) \right] \tag{11.8}$$

for subsequent layers

$$\tau_{xz} = \tau_{xz}^{(i-1)} - \frac{3Q_{xz}C_{11}^{(i)}}{\sum_{j=1}^{NL} C_{11}^{(j)}(z_j^3 - z_{j-1}^3)}\left[\frac{z^2 - z_o^2}{2} - \overline{z}_x(z - z_o)\right], z_{i-1} \le z \le z_i \qquad (11.9)$$

Here $\tau_{xz}^{(i-1)}$ is the stress in previous layer at the interface with the current layer. The shear stress can also be expressed as follows:

$$\tau_{xz} = \frac{3Q_{xz}C_{11}^{(i)}}{\sum_{j=1}^{NL} C_{11}^{(j)}(z_j^3 - z_{j-1}^3)}\left[f_x^{(i)} + \frac{z^2 - z_{i-1}^2}{2} - \overline{z}_x(z - z_{i-1})\right] \qquad (11.10)$$

Where

$$f_x^{(i)} = \frac{1}{C_{11}^{(i)}}\sum_{j=1}^{NL} C_{11}^{(j)} h_j \left[\frac{z_j + z_{j+1}}{2} - \overline{z}_x\right] \qquad (11.11)$$

and

$$h_j = z_j - z_{j-1}$$

To find Q_{xz}, the shear force, assume that the strain energy expressed through average shear modules, $\overline{C}_{66}$, is equal to the strain energy expressed through the derived expressions as follows:

$$U = \frac{1}{2}\frac{Q_{xz}}{\overline{C}_{66}h} = \frac{1}{2}\int \frac{\tau_{xz}^{\ 2}}{C_{66}}dz \qquad (11.12)$$

$$\frac{1}{\overline{C}_{66}} = \frac{9h}{\left[\sum_{j=1}^{NL} C_{11}^{(i)}(z_i^3 - z_{i-1}^3)\right]^2}\int \frac{C_{11}^2}{C_{66}}\left[f_x^{(i)} + \frac{(z^2 - z_{i-1}^2)}{2} - \overline{z}_x(z - z_{i-1})\right]^2 dz$$

$$= \frac{9h}{\left[\sum_{j=1}^{NL} C_{11}^{(i)}(z_i^3 - z_{i-1}^3)\right]^2}\sum_{i=1}^{NL} \frac{(C_{11}^{\ (i)})^2}{C_{66}^{(i)}}\int_{z_{i-1}}^{z_i}\left[f_x^{(i)} + \frac{z^2 - z_{i-1}^{\ 2}}{2} - \overline{z}_x(z - z_{i-1})\right]^2 dz$$

$$= \frac{1}{60}\frac{9h}{\left[\sum_{j=1}^{NL} C_{11}^{(i)}(z_i^3 - z_{i-1}^3)\right]^2}\sum_{i}^{NL} \frac{(C_{11}^{(i)})^2 h}{C_{66}^i}\{f_x^i\left[60f_x^i + 20h_i(z_i + 2z_{i-1} - 3\overline{z}_x)\right] +$$

$$\bar{z}_x h_i \left[20\bar{z}_x h_i + 35z_{i-1}^2 - 10z_{i-1}(z_i + z_{i-1}) - 15z_i^2 \right] + z_i(z_i + z_{i-1})(3z_i^2 - 7z_{i-1}^2) + 8z_{i-1}^4 \} = \frac{1}{\bar{C}_{66}} \tag{11.13}$$

then

$$Q_{xz} = \bar{\tau}_{xz} h = \bar{C}_{66} \bar{\gamma}_{xz} h$$

to calculate $\bar{z}_x$ use τ_{xz} for last layer at surface $z = 0$

$$\sum_{i=1}^{NL} C_{11}^{(i)} \left[\left(\frac{z_i^2 - z_{i-1}^2}{2} \right) - \bar{z}_x (z_i - z_{i-1}) \right] = 0$$

where

$$\bar{z}_x = \frac{\sum_{i=1}^{NL} C_{11}^{(i)} h_i (z_i + z_{i+1})}{2 \sum_{i=1}^{NL} C_{11}^{(i)} h_i} \tag{11.14}$$

Algorithm:
The following algorithm is used in the implementation of the transverse shear treatment.

1. Calculate $\bar{z}_x$ according to equation (14)
2. Calculate f_x^i according to equation (11)
3. Calculate $\frac{1}{3} \sum_{i=1}^{NL} C_{11}^{(i)} (z_i^3 - z_{i-1}^3)$
4. Calculate $h \left(\frac{1}{3} \sum_{i=1}^{NL} C_{11}^{(i)} (z_i^3 - z_{i-1}^3) \right)^2$
5. Calculate $\bar{C}_{66}$ according to equation (13)
6. Calculate $Q_{xz} = \bar{C}_{66} \bar{\gamma}_{xz} h$
7. Calculate τ_{xz}

$$\tau_{xz}^{(i)} = -\frac{1}{3} \sum_{i=1}^{NL} C_{11}^{(i)} (z_i^3 - z_{i-1}^3) Q_{xz} C_{11}^{(i)} \left[f_x^{(i)} + \frac{z^2 - z_{i-1}^2}{2} - \bar{z}_x (z - z_{i-1}) \right] \quad z_{i-1} \le z \le z_i$$

Steps 1-5 are performed at the initialization stage. Step 6 is performed in the shell formulation subroutine, and step 7 is performed in the stress calculation inside the constitutive subroutine.

12. EIGHT-NODE SOLID SHELL ELEMENT

The isoparametric eight-node brick element discussed in Section 3 forms the basis for this shell element with enhancements based on the Hughes-Liu and the Belytschko-Lin-Tsay shells. Like the eight-node brick, the geometry is interpolated from the nodal point coordinates as:

$$x_i\left(X_\alpha,t\right)=x_i\left(X_\alpha\left(\xi,\eta,\zeta\right),t\right)=\sum_{j=1}^{8}\phi_j\left(\xi,\eta,\zeta\right)x_i^j\left(t\right) \tag{12.1}$$

where the shape function ϕ_j are defined as

$$\phi_j=\frac{1}{8}\left(1+\xi\xi_j\right)\left(1+\eta\eta_j\right)\left(1+\zeta\zeta_j\right) \tag{12.2}$$

where ξ_j,η_j,ζ_j take on their nodal values of (±1, ±1, ±1) and x_i^j is the nodal coordinate of the *j*th node in the *i*th direction (See Figure 12.1).

As with solid elements, N is the 3 x 24 rectangular interpolation matrix:

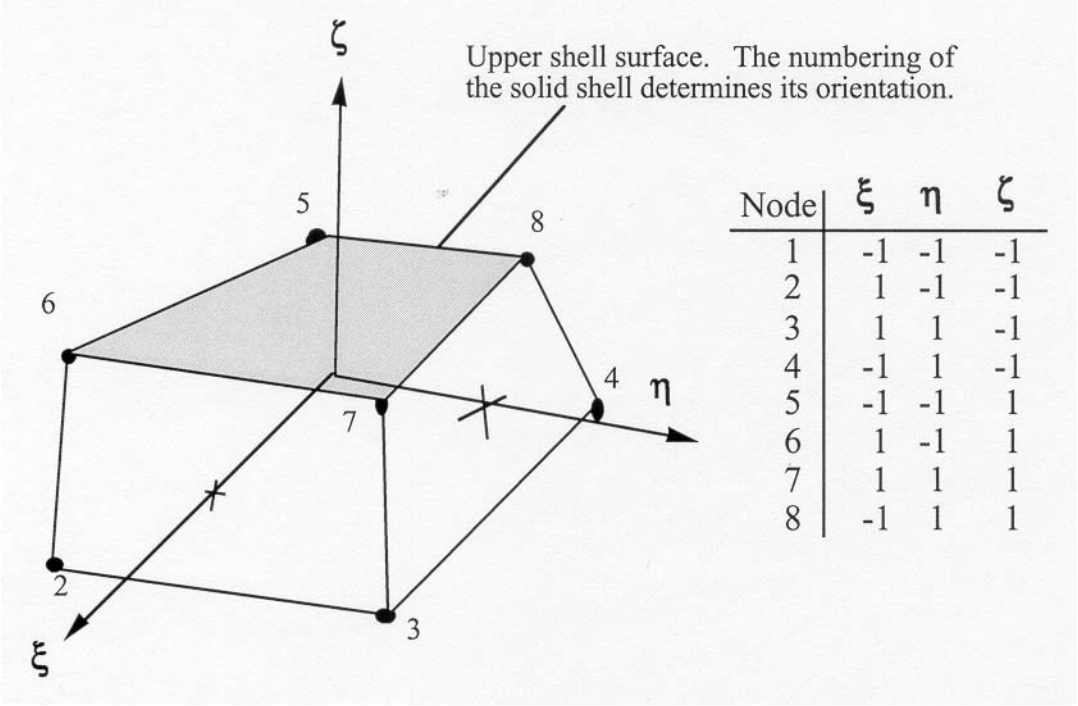
$$\mathrm{N}\left(\xi,\eta,\zeta\right)=\begin{bmatrix}\phi_1 & 0 & 0 & \phi_2 & 0 & \dots & 0 & 0\\ 0 & \phi_1 & 0 & 0 & \phi_2 & \dots & \phi_8 & 0\\ 0 & 0 & \phi_1 & 0 & 0 & \dots & 0 & \phi_8\end{bmatrix} \tag{12.3}$$

σ is the stress vector:

$$o^t=\left(\sigma_{xx},\sigma_{yy},\sigma_{zz},\sigma_{xy},\sigma_{yz},\sigma_{zx}\right) \tag{12.4}$$

and B is the 6 x 24 strain-displacement matrix:

$$\mathrm{B}=\begin{bmatrix}\frac{\partial}{\partial x} & 0 & 0\\ 0 & \frac{\partial}{\partial y} & 0\\ 0 & 0 & \frac{\partial}{\partial z}\\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0\\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y}\\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x}\end{bmatrix}\mathrm{N} \tag{12.5}$$



Node	ξ	η	ζ
1	-1	-1	-1
2	1	-1	-1
3	1	1	-1
4	-1	1	-1
5	-1	-1	1
6	1	-1	1
7	1	1	1
8	-1	1	1

Figure 12.1. Eight node solid shell element

Terms in the strain-displacement matrix are readily calculated. Note that

$$\begin{aligned}
\frac{\partial \phi_i}{\partial \xi} &= \frac{\partial \phi_i}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial \phi_i}{\partial y}\frac{\partial y}{\partial \xi} + \frac{\partial \phi_i}{\partial z}\frac{\partial z}{\partial \xi} \\
\frac{\partial \phi_i}{\partial \eta} &= \frac{\partial \phi_i}{\partial x}\frac{\partial x}{\partial \eta} + \frac{\partial \phi_i}{\partial y}\frac{\partial y}{\partial \eta} + \frac{\partial \phi_i}{\partial z}\frac{\partial z}{\partial \eta} \\
\frac{\partial \phi_i}{\partial \zeta} &= \frac{\partial \phi_i}{\partial x}\frac{\partial x}{\partial \zeta} + \frac{\partial \phi_i}{\partial y}\frac{\partial y}{\partial \zeta} + \frac{\partial \phi_i}{\partial z}\frac{\partial z}{\partial \zeta}
\end{aligned} \tag{12.6}$$

which can be rewritten as

$$\begin{bmatrix} \frac{\partial \phi_i}{\partial \xi} \\ \frac{\partial \phi_i}{\partial \eta} \\ \frac{\partial \phi_i}{\partial \zeta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \begin{bmatrix} \frac{\partial \phi_i}{\partial x} \\ \frac{\partial \phi_i}{\partial y} \\ \frac{\partial \phi_i}{\partial z} \end{bmatrix} = J \begin{bmatrix} \frac{\partial \phi_i}{\partial x} \\ \frac{\partial \phi_i}{\partial y} \\ \frac{\partial \phi_i}{\partial z} \end{bmatrix} \tag{12.7}$$

Inverting the Jacobian matrix, J, we can solve for the desired terms

$$\begin{bmatrix} \frac{\partial \phi_i}{\partial x} \\ \frac{\partial \phi_i}{\partial y} \\ \frac{\partial \phi_i}{\partial z} \end{bmatrix} = J^{-1} \begin{bmatrix} \frac{\partial \phi_i}{\partial \xi} \\ \frac{\partial \phi_i}{\partial \eta} \\ \frac{\partial \phi_i}{\partial \zeta} \end{bmatrix} \tag{12.8}$$

To obtain shell-like behavior from the solid element, it is necessary to use multiple integration points through the shell thickness along the ζ axis while employing a plane stress constitutive subroutine. Consequently, it is necessary to construct a reference surface within the brick shell. We locate the reference surface midway between the upper and lower surfaces and construct a local coordinate system exactly as was done for the Belytschko-Lin-Tsay shell element. Following the procedure outlined in Section 7, Equations (7.1) – (7.3), the local coordinate system can be constructed as depicted in Figure 12.2. Equation (7.5a) gives the transformation matrix in terms of the local basis:

$$\{A\} = \begin{Bmatrix} A_x \\ A_y \\ A_z \end{Bmatrix} = \begin{bmatrix} e_{1x} & e_{2x} & e_{3x} \\ e_{1y} & e_{2y} & e_{3y} \\ e_{1z} & e_{2z} & e_{3z} \end{bmatrix} \begin{Bmatrix} \hat{A}_x \\ \hat{A}_y \\ \hat{A}_z \end{Bmatrix} = [\mu]\{\hat{A}\} = [q]^T \{\hat{A}\} \tag{12.9}$$

As with the Hughes-Liu shell, the next step is to perform the Jaumann rate update:

$$\underline{\sigma}_{ij}^{n+1} = \sigma_{ij}^n + \sigma_{ip}^n \Delta\omega_{pj} + \sigma_{jp}^n \Delta\omega_{pi} \tag{10.22}$$

to account for the material rotation between time steps n and $n+1$. The Jaumann rate update of the stress tensor is applied in the global configuration before the constitutive evaluation is performed. In the solid shell, as in the Hughes-Liu shell, the stresses and history variables are stored in the global coordinate system. To evaluate the constitutive relation, the stresses and the strain increments are rotated from the global to the lamina coordinate system using the transformation defined previously:

$$\underline{\sigma}_{ij}^{l^{n+1}} = q_{ik} \underline{\sigma}_{kn}^{n+1} q_{jn} \tag{10.23}$$

$$\Delta\varepsilon_{ij}^{l^{n+1/2}} = q_{ik} \Delta\varepsilon_{kn}^{n+1/2} q_{jn} \tag{10.24}$$

where the superscript l indicates components in the lamina (local) coordinate system. The stress is updated incrementally:

$$\sigma_{ij}^{l^{n+1}} = \underline{\sigma}_{ij}^{l^{n+1}} + \Delta\sigma_{ij}^{l^{n+1/2}} \tag{10.25}$$

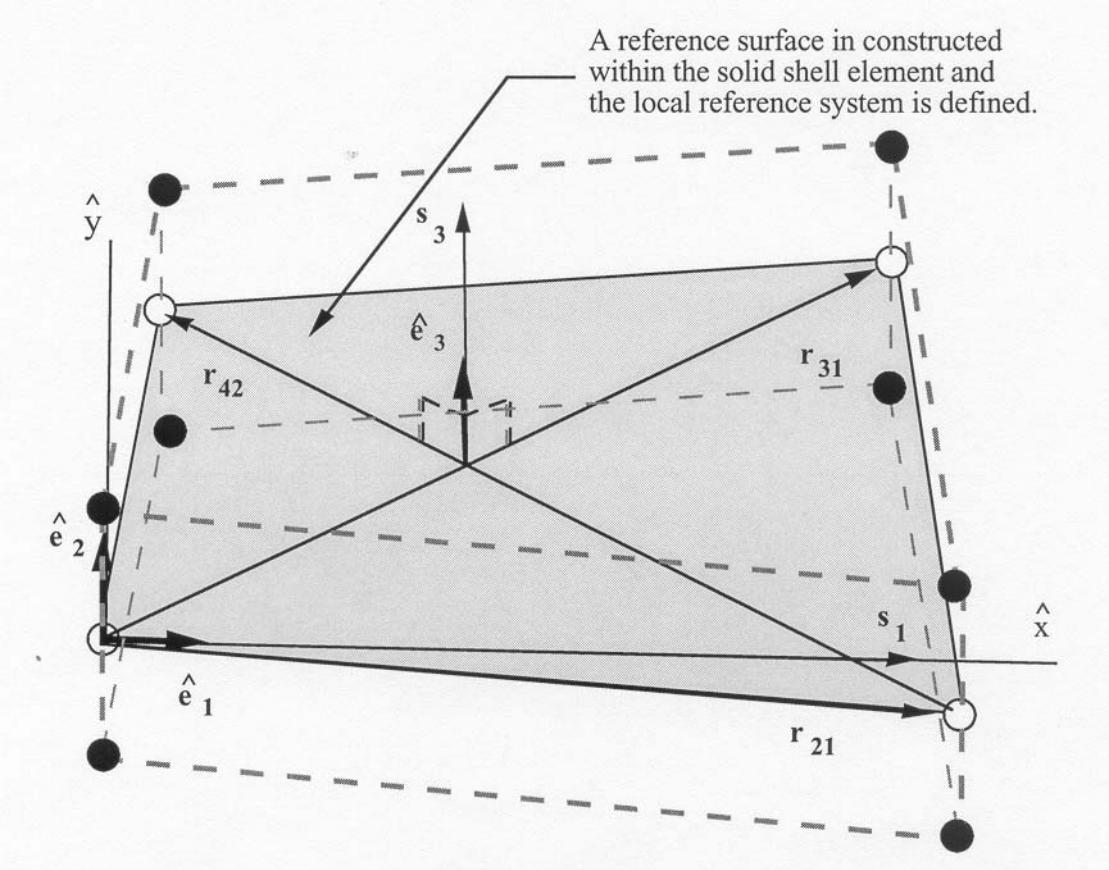


Figure 12.2. Construction of the reference surface in the solid shell element.

Independently from the constitutive evaluation

$$\sigma_{33}^{l} = 0 \tag{12.10}$$

which ensures that the plane stress condition is satisfied, we update the normal stress which is used as a penalty to maintain the thickness of the shell:

$$\sigma_{33}^{penalty^{n+1}} = \sigma_{33}^{penalty^{n}} + E\Delta\varepsilon_{33}^{l} \tag{12.11}$$

where E is the elastic Young's modulus for the material. The stress tensor of Equation (10.25) is rotated back to the global system:

$$\sigma_{ij}^{n+1} = q_{ki}\sigma_{kn}^{l^{n+1}} q_{nj} \tag{10.26}$$

A penalty stress tensor is then formed by transforming the normal penalty stress tensor (a null tensor except for the 33 term) back to the global system:

$$\sigma_{ij}^{penalty^{n+1}} = q_{ki}\sigma_{kn}^{penalty^{l^{n+1}}} q_{nj} \tag{12.12}$$

before computing the internal force vector. The internal force vector can now be computed:

$$f^{\text{int}} = \int B^{t^{n+1}} \left[\sigma^{n+1} + \sigma^{penalty^{n+1}} \right] dv \tag{12.13}$$

The brick shell exhibits no discernible locking problems with this approach.

The treatment of the hourglass modes is identical to that described for the solid elements in Section 3.

13. EIGHT-NODE SOLID ELEMENT FOR THICK SHELL SIMULATIONS

13.1 Abstract

An eight-node hexahedral solid element is incorporated into LS-DYNA to simulate thick shell structure. The element formulations are derived in a co-rotational coordinate system and the strain operator is calculated with a Taylor series expansion about the center of the element. Special treatments are made on the dilatational strain component and shear strain components to eliminate the volumetric and shear locking. The use of consistent tangential stiffness and geometric stiffness greatly improves the convergence rate in implicit analysis.

13.2 Introduction

Large-scale finite element analyses are extensively used in engineering designs and process controls. For example, in automobile crashworthiness, hundreds of thousands of unknowns are involved in the computer simulation models, and in metal forming processing, tests in the design of new dies or new products are done by numerical computations instead of costly experiments. The efficiency of the elements is of crucial importance to speed up the design processes and reduce the computational costs for these problems. Over the past ten years, considerable progress has been achieved in developing fast and reliable elements.

In the simulation of shell structures, Belytschko-Lin-Tsay [Belytschko, 1984a] and Hughes-Liu [Hughes, 1981a and 1981b] shell elements are widely used. However, in some cases thick shell elements are more suitable. For example, in the sheet metal forming with large curvature, traditional thin shell elements cannot give satisfactory results. Also thin shell elements cannot give us detailed strain information though the thickness. In LS-DYNA, the eight-node solid thick shell element is still based on the Hughes-Liu and Belytschko-Lin-Tsay shells [Hallquist, 1998]. A new eight-node solid element based on Liu, 1985, 1994 and 1998 is incorporated into LS-DYNA, intended for thick shell simulation. The strain operator of this element is derived from a Taylor series expansion and special treatments on strain components are utilized to avoid volumetric and shear locking.

The organization of this paper is as follows. The element formulations are described in the next section. Several numerical problems are studied in the third section, followed by the conclusions.

13.3 Element Formulations

13.3.1 Strain Operator

The new element is based on the eight-node hexahedral element proposed and enhanced by Liu, 1985, 1994, 1998. For an eight-node hexahedral element, the spatial coordinates, x_i, and the velocity components, v_i, in the element are approximated in terms of nodal values, x_{ia} and v_{ia}, by

$$x_i = \sum_{a=1}^{8} N_a(\xi,\eta,\zeta)x_{ia} \tag{13.1}$$

$$v_i = \sum_{a=1}^{8} N_a(\xi,\eta,\zeta)v_{ia}, \quad i=1,2,3 \tag{13.2}$$

where the trilinear shape functions are expressed as

$$N_a(\xi,\eta,\zeta)=\frac{1}{8}(1+\xi_a\xi)(1+\eta_a\eta)(1+\zeta_a\zeta) \tag{13.3}$$

and the subscripts i and a denote coordinate components ranging from one to three and the element nodal numbers ranging from one to eight, respectively. The referential coordinates ξ, η, and ζ of node a are denoted by ξ_a, η_a, and ζ_a, respectively.

The strain rate (or rate of deformation), $\dot{\varepsilon}$, is composed of six components,

$$\dot{\varepsilon}^t=\begin{bmatrix} \varepsilon_{xx} & \varepsilon_{yy} & \varepsilon_{zz} & \varepsilon_{xy} & \varepsilon_{yz} & \varepsilon_{zx} \end{bmatrix} \tag{13.4}$$

and is related to the nodal velocities by a strain operator, $\overline{\mathrm{B}}$,

$$\dot{\varepsilon}=\overline{\mathrm{B}}(\xi,\eta,\zeta)\,\mathrm{v}, \tag{13.5}$$

where

$$v^t=\begin{bmatrix} v_{x1} & v_{y1} & v_{z1} & \cdots & v_{x8} & v_{y8} & v_{z8} \end{bmatrix}, \tag{13.6}$$

$$\overline{\mathrm{B}}=\begin{bmatrix} \overline{\mathrm{B}}_{xx} \\ \overline{\mathrm{B}}_{yy} \\ \overline{\mathrm{B}}_{zz} \\ \overline{\mathrm{B}}_{xy} \\ \overline{\mathrm{B}}_{yz} \\ \overline{\mathrm{B}}_{zx} \end{bmatrix} = \begin{bmatrix} B_1(1) & 0 & 0 & \cdots & B_1(8) & 0 & 0 \\ 0 & B_2(1) & 0 & \cdots & 0 & B_2(8) & 0 \\ 0 & 0 & B_3(1) & \cdots & 0 & 0 & B_3(8) \\ B_2(1) & B_1(1) & 0 & \cdots & B_2(8) & B_1(8) & 0 \\ 0 & B_3(1) & B_2(1) & \cdots & 0 & B_3(8) & B_2(8) \\ B_3(1) & 0 & B_1(1) & \cdots & B_3(8) & 0 & B_1(8) \end{bmatrix} \tag{13.7}$$

and B_1, B_2 and B_3 are gradient vectors,

$$\begin{bmatrix} \mathrm{B}_1 \\ \mathrm{B}_2 \\ \mathrm{B}_3 \end{bmatrix} = \begin{bmatrix} N_{,x}(\xi,\eta,\zeta) \\ N_{,y}(\xi,\eta,\zeta) \\ N_{,z}(\xi,\eta,\zeta) \end{bmatrix}. \tag{13.8}$$

Unlike standard solid element where the strain operator is computed by differentiating the shape functions, the strain operator for this new element is expanded in a Taylor series about the element center up to bilinear terms as follows [Liu, 1994, 1998],

$$\begin{aligned} \overline{\mathrm{B}}(\xi,\eta,\zeta)=\overline{\mathrm{B}}(0)+\overline{\mathrm{B}}_{,\xi}(0)\xi+\overline{\mathrm{B}}_{,\eta}(0)\eta+\overline{\mathrm{B}}_{,\zeta}(0)\zeta \\ +2\left[\overline{\mathrm{B}}_{,\xi\eta}(0)\xi\eta+\overline{\mathrm{B}}_{,\eta\zeta}(0)\eta\zeta+\overline{\mathrm{B}}_{,\zeta\xi}(0)\zeta\xi\right]. \end{aligned} \tag{13.9}$$

The first term on the right-hand side of the above Equation (13.9) corresponds to the constant strain rates evaluated at the central point and the remaining terms are linear and bilinear strain rate terms.

Let

$$x_1^t = x^t = \left[x_1 \;\; x_2 \;\; x_3 \;\; x_4 \;\; x_5 \;\; x_6 \;\; x_7 \;\; x_8 \right], \tag{13.10}$$

$$x_2^t = y^t = \left[y_1 \;\; y_2 \;\; y_3 \;\; y_4 \;\; y_5 \;\; y_6 \;\; y_7 \;\; y_8 \right], \tag{13.11}$$

$$x_3^t = z^t = \left[z_1 \;\; z_2 \;\; z_3 \;\; z_4 \;\; z_5 \;\; z_6 \;\; z_7 \;\; z_8 \right], \tag{13.12}$$

$$\xi^t = \left[-1 \;\; 1 \;\; 1 \;\; -1 \;\; -1 \;\; 1 \;\; 1 \;\; -1 \right], \tag{13.13}$$

$$\eta^t = \left[-1 \;\; -1 \;\; 1 \;\; 1 \;\; -1 \;\; -1 \;\; 1 \;\; 1 \right], \tag{13.14}$$

$$\zeta^t = \left[-1 \;\; -1 \;\; -1 \;\; -1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \right], \tag{13.15}$$

the Jacobian matrix at the center of the element can be evaluated as

$$\mathrm{J}(0) = [J_{ij}] = \frac{1}{8} \begin{bmatrix} \xi^t \mathrm{x} & \xi^t y & \xi^t z \\ \eta^t \mathrm{x} & \eta^t y & \eta^t z \\ \zeta^t \mathrm{x} & \zeta^t y & \zeta^t z \end{bmatrix}; \tag{13.16}$$

the determinant of the Jacobian matrix is denoted by j_0 and the inverse matrix of $\mathrm{J}(0)$ is denoted by D

$$\mathrm{D} = \left[D_{ij} \right] = J^{-1}(0) \tag{13.17}$$

The gradient vectors and their derivatives with respect to the natural coordinates at the center of the element are given as follows,

$$\mathrm{b}_1 = \mathrm{N}_{,x}(0) = \frac{1}{8} \left[D_{11}\xi + D_{12}\eta + D_{13}\zeta \right], \tag{13.18}$$

$$\mathrm{b}_2 = \mathrm{N}_{,y}(0) = \frac{1}{8} \left[D_{21}\xi + D_{22}\eta + D_{23}\zeta \right], \tag{13.19}$$

$$\mathrm{b}_3 = \mathrm{N}_{,z}(0) = \frac{1}{8} \left[D_{31}\xi + D_{32}\eta + D_{33}\zeta \right]. \tag{13.20}$$

$$\mathrm{b}_{1,\xi} = \mathrm{N}_{,x\xi}(0) = \frac{1}{8} \left[D_{12}\gamma_1 + D_{13}\gamma_2 \right], \tag{13.21}$$

$$\mathrm{b}_{2,\xi} = \mathrm{N}_{,y\xi}(0) = \frac{1}{8} \left[D_{22}\gamma_1 + D_{23}\gamma_2 \right], \tag{13.22}$$

$$\mathrm{b}_{3,\xi} = \mathrm{N}_{,z\xi}(0) = \frac{1}{8} \left[D_{32}\gamma_1 + D_{33}\gamma_2 \right], \tag{13.23}$$

$$\mathrm{b}_{1,\eta} = \mathrm{N}_{,x\eta}(0) = \frac{1}{8} \left[D_{11}\gamma_1 + D_{13}\gamma_3 \right], \tag{13.24}$$

$$\mathrm{b}_{2,\eta} = \mathrm{N}_{,y\eta}(0) = \frac{1}{8}\left[D_{21}\gamma_1 + D_{23}\gamma_3\right], \tag{13.25}$$

$$\mathrm{b}_{3,\eta} = \mathrm{N}_{,z\eta}(0) = \frac{1}{8}\left[D_{31}\gamma_1 + D_{33}\gamma_3\right], \tag{13.26}$$

$$\mathrm{b}_{1,\zeta} = \mathrm{N}_{,x\zeta}(0) = \frac{1}{8}\left[D_{11}\gamma_2 + D_{12}\gamma_3\right], \tag{13.27}$$

$$\mathrm{b}_{2,\zeta} = \mathrm{N}_{,y\zeta}(0) = \frac{1}{8}\left[D_{21}\gamma_2 + D_{22}\gamma_3\right], \tag{13.28}$$

$$\mathrm{b}_{3,\zeta} = \mathrm{N}_{,z\zeta}(0) = \frac{1}{8}\left[D_{31}\gamma_2 + D_{32}\gamma_3\right], \tag{13.29}$$

$$\mathrm{b}_{1,\xi\eta} = \mathrm{N}_{,x\xi\eta}(0) = \frac{1}{8}\left[D_{13}\gamma_4 - (\mathrm{p}_1^t\,\mathrm{x}_i)\,\mathrm{b}_{i,\xi} - (\mathrm{r}_1^t\,\mathrm{x}_i)\,\mathrm{b}_{i,\eta}\right], \tag{13.30}$$

$$\mathrm{b}_{2,\xi\eta} = \mathrm{N}_{,y\xi\eta}(0) = \frac{1}{8}\left[D_{23}\gamma_4 - (\mathrm{p}_2^t\,\mathrm{x}_i)\,\mathrm{b}_{i,\xi} - (\mathrm{r}_2^t\,\mathrm{x}_i)\,\mathrm{b}_{i,\eta}\right], \tag{13.31}$$

$$\mathrm{b}_{3,\xi\eta} = \mathrm{N}_{,z\xi\eta}(0) = \frac{1}{8}\left[D_{33}\gamma_4 - (\mathrm{p}_3^t\,\mathrm{x}_i)\,\mathrm{b}_{i,\xi} - (\mathrm{r}_3^t\,\mathrm{x}_i)\,\mathrm{b}_{i,\eta}\right], \tag{13.32}$$

$$\mathrm{b}_{1,\eta\zeta} = \mathrm{N}_{,x\eta\zeta}(0) = \frac{1}{8}\left[D_{11}\gamma_4 - (\mathrm{q}_1^t\,\mathrm{x}_i)\,\mathrm{b}_{i,\eta} - (\mathrm{p}_1^t\,\mathrm{x}_i)\,\mathrm{b}_{i,\zeta}\right], \tag{13.33}$$

$$\mathrm{b}_{2,\eta\zeta} = \mathrm{N}_{,y\eta\zeta}(0) = \frac{1}{8}\left[D_{21}\gamma_4 - (\mathrm{q}_2^t\,\mathrm{x}_i)\,\mathrm{b}_{i,\eta} - (\mathrm{p}_2^t\,\mathrm{x}_i)\,\mathrm{b}_{i,\zeta}\right], \tag{13.34}$$

$$\mathrm{b}_{3,\eta\zeta} = \mathrm{N}_{,z\eta\zeta}(0) = \frac{1}{8}\left[D_{31}\gamma_4 - (\mathrm{q}_3^t\,\mathrm{x}_i)\,\mathrm{b}_{i,\eta} - (\mathrm{p}_3^t\,\mathrm{x}_i)\,\mathrm{b}_{i,\zeta}\right], \tag{13.35}$$

$$\mathrm{b}_{1,\zeta\xi} = \mathrm{N}_{,x\zeta\xi}(0) = \frac{1}{8}\left[D_{12}\gamma_4 - (\mathrm{r}_1^t\,\mathrm{x}_i)\,\mathrm{b}_{i,\zeta} - (\mathrm{q}_1^t\,\mathrm{x}_i)\,\mathrm{b}_{i,\xi}\right], \tag{13.36}$$

$$\mathrm{b}_{2,\zeta\xi} = \mathrm{N}_{,y\zeta\xi}(0) = \frac{1}{8}\left[D_{22}\gamma_4 - (\mathrm{r}_2^t\,\mathrm{x}_i)\,\mathrm{b}_{i,\zeta} - (\mathrm{q}_2^t\,\mathrm{x}_i)\,\mathrm{b}_{i,\xi}\right], \tag{13.37}$$

$$\mathrm{b}_{3,\zeta\xi} = \mathrm{N}_{,z\zeta\xi}(0) = \frac{1}{8}\left[D_{32}\gamma_4 - (\mathrm{r}_3^t\,\mathrm{x}_i)\,\mathrm{b}_{i,\zeta} - (\mathrm{q}_3^t\,\mathrm{x}_i)\,\mathrm{b}_{i,\xi}\right], \tag{13.38}$$

where

$$\mathrm{p}_i = D_{i1}\,\mathrm{h}_1 + D_{i3}\,\mathrm{h}_3, \tag{13.39}$$

$$\mathrm{q}_i = D_{i1}\,\mathrm{h}_2 + D_{i2}\,\mathrm{h}_3, \tag{13.40}$$

$$\mathrm{r}_i = D_{i2}\,\mathrm{h}_1 + D_{i3}\,\mathrm{h}_2, \tag{13.41}$$

$$\gamma_\alpha = \mathrm{h}_\alpha - \left(\mathrm{h}_\alpha^t\,\mathrm{x}_i\right)\,\mathrm{b}_i, \tag{13.42}$$

and

$$\mathrm{h}_1^t = \left[1\ \ -1\ \ 1\ \ -1\ \ 1\ \ -1\ \ 1\ \ -1\right], \tag{13.43}$$

$$\mathrm{h}_2^t = \left[1\ \ -1\ \ -1\ \ 1\ \ -1\ \ 1\ \ 1\ \ -1\right], \tag{13.44}$$

$$\mathrm{h}_3^t = \left[1 \;\; 1 \;\; -1 \;\; -1 \;\; -1 \;\; -1 \;\; 1 \;\; 1\right], \tag{13.45}$$

$$\mathrm{h}_4^t = \left[-1 \;\; 1 \;\; -1 \;\; 1 \;\; 1 \;\; -1 \;\; 1 \;\; -1\right], \tag{13.46}$$

In the above equations h_1 is the $\xi\eta$-hourglass vector, h_2 the $\eta\zeta$-hourglass vector, h_3 the $\zeta\xi$-hourglass vector and h_4 the $\xi\eta\zeta$-hourglass vector. They are the zero energy-deformation modes associated with the one-point-quadrature element which result in a non-constant strain field in the element [Flanagan, 1981, Belytschko, 1984 and Liu, 1984]. The γ_α in equations (13.21)–(13.38) are the stabilization vectors. They are orthogonal to the linear displacement field and provide a consistent stabilization for the element.

The strain operators, $\bar{\mathrm{B}}(\xi,\eta,\zeta)$, can be decomposed into two parts, the dilatational part, $\bar{\mathrm{B}}^{dil}(\xi,\eta,\zeta)$, and the deviatoric part, $\bar{\mathrm{B}}^{dev}(\xi,\eta,\zeta)$, both of which can be expanded about the element center as in Equation (13.9)

$$\begin{aligned}\bar{\mathrm{B}}^{dil}(\xi,\eta,\zeta) = {} & \bar{\mathrm{B}}^{dil}(0) + \bar{\mathrm{B}}^{dil}_{,\xi}(0)\,\xi + \bar{\mathrm{B}}^{dil}_{,\eta}(0)\,\eta + \bar{\mathrm{B}}^{dil}_{,\zeta}(0)\,\zeta \\ & + 2\left[\, \bar{\mathrm{B}}^{dil}_{,\xi\eta}(0)\,\xi\eta + \bar{\mathrm{B}}^{dil}_{,\eta\zeta}(0)\,\eta\zeta + \bar{\mathrm{B}}^{dil}_{,\zeta\xi}(0)\,\zeta\xi \right],\end{aligned} \tag{13.47}$$

$$\begin{aligned}\bar{\mathrm{B}}^{dev}(\xi,\eta,\zeta) = {} & \bar{\mathrm{B}}^{dev}(0) + \bar{\mathrm{B}}^{dev}_{,\xi}(0)\,\xi + \bar{\mathrm{B}}^{dev}_{,\eta}(0)\,\eta + \bar{\mathrm{B}}^{dev}_{,\zeta}(0)\,\zeta \\ & + 2\left[\, \bar{\mathrm{B}}^{dev}_{,\xi\eta}(0)\,\xi\eta + \bar{\mathrm{B}}^{dev}_{,\eta\zeta}(0)\,\eta\zeta + \bar{\mathrm{B}}^{dev}_{,\zeta\xi}(0)\,\zeta\xi \right],\end{aligned} \tag{13.48}$$

To avoid volumetric locking, the dilatational part of the strain operators is evaluated only at one quadrature point, the center of the element, i.e., they are constant terms

$$\bar{\mathrm{B}}^{dil}(\xi,\eta,\zeta) = \bar{\mathrm{B}}^{dil}(0)\,. \tag{13.49}$$

To remove shear locking, the deviatoric strain submatrices can be written in an orthogonal co-rotational coordinate system rotating with the element as

$$\begin{aligned}\bar{\mathrm{B}}^{dev}_{xx}(\xi,\eta,\zeta) = {} & \bar{\mathrm{B}}^{dev}_{xx}(0) + \bar{\mathrm{B}}^{dev}_{xx,\xi}(0)\,\xi + \bar{\mathrm{B}}^{dev}_{xx,\eta}(0)\,\eta + \bar{\mathrm{B}}^{dev}_{xx,\zeta}(0)\,\zeta \\ & + 2\left[\, \bar{\mathrm{B}}^{dev}_{xx,\xi\eta}(0)\,\xi\eta + \bar{\mathrm{B}}^{dev}_{xx,\eta\zeta}(0)\,\eta\zeta + \bar{\mathrm{B}}^{dev}_{xx,\zeta\xi}(0)\,\zeta\xi \right],\end{aligned} \tag{13.50}$$

$$\begin{aligned}\bar{\mathrm{B}}^{dev}_{yy}(\xi,\eta,\zeta) = {} & \bar{\mathrm{B}}^{dev}_{yy}(0) + \bar{\mathrm{B}}^{dev}_{yy,\xi}(0)\,\xi + \bar{\mathrm{B}}^{dev}_{yy,\eta}(0)\,\eta + \bar{\mathrm{B}}^{dev}_{yy,\zeta}(0)\,\zeta \\ & + 2\left[\, \bar{\mathrm{B}}^{dev}_{yy,\xi\eta}(0)\,\xi\eta + \bar{\mathrm{B}}^{dev}_{yy,\eta\zeta}(0)\,\eta\zeta + \bar{\mathrm{B}}^{dev}_{yy,\zeta\xi}(0)\,\zeta\xi \right],\end{aligned} \tag{13.51}$$

$$\begin{aligned}\bar{\mathrm{B}}^{dev}_{zz}(\xi,\eta,\zeta) = {} & \bar{\mathrm{B}}^{dev}_{zz}(0) + \bar{\mathrm{B}}^{dev}_{zz,\xi}(0)\,\xi + \bar{\mathrm{B}}^{dev}_{zz,\eta}(0)\,\eta + \bar{\mathrm{B}}^{dev}_{zz,\zeta}(0)\,\zeta \\ & + 2\left[\, \bar{\mathrm{B}}^{dev}_{zz,\xi\eta}(0)\,\xi\eta + \bar{\mathrm{B}}^{dev}_{zz,\eta\zeta}(0)\,\eta\zeta + \bar{\mathrm{B}}^{dev}_{zz,\zeta\xi}(0)\,\zeta\xi \right],\end{aligned} \tag{13.52}$$

$$\bar{\mathrm{B}}^{dev}_{xy}(\xi,\eta,\zeta) = \bar{\mathrm{B}}^{dev}_{xy}(0) + \bar{\mathrm{B}}^{dev}_{xy,\zeta}(0)\,\zeta\,, \tag{13.53}$$

$$\bar{\mathrm{B}}^{dev}_{yz}(\xi,\eta,\zeta) = \bar{\mathrm{B}}^{dev}_{yz}(0) + \bar{\mathrm{B}}^{dev}_{yz,\xi}(0)\,\xi, \tag{13.54}$$

$$\bar{\mathrm{B}}^{dev}_{zx}(\xi,\eta,\zeta) = \bar{\mathrm{B}}^{dev}_{zx}(0) + \bar{\mathrm{B}}^{dev}_{zx,\eta}(0)\,\eta. \tag{13.55}$$

Here, only one linear term is left for shear strain components such that the modes causing shear locking are removed. The normal strain components keep all non-constant terms given in equation (13.48).

Summation of equation (13.49) and equations (13.50)–(13.55) yields the following strain submatrices which can eliminate the shear and volumetric locking:

$$\begin{aligned}\bar{\mathrm{B}}_{xx}(\xi,\eta,\zeta) = {} & \bar{\mathrm{B}}_{xx}(0) + \bar{\mathrm{B}}^{dev}_{xx,\xi}(0)\,\xi + \bar{\mathrm{B}}^{dev}_{xx,\eta}(0)\,\eta + \bar{\mathrm{B}}^{dev}_{xx,\zeta}(0)\,\zeta \\ & + 2\left[\, \bar{\mathrm{B}}^{dev}_{xx,\xi\eta}(0)\,\xi\eta + \bar{\mathrm{B}}^{dev}_{xx,\eta\zeta}(0)\,\eta\zeta + \bar{\mathrm{B}}^{dev}_{xx,\zeta\xi}(0)\,\zeta\xi \right],\end{aligned} \tag{13.56}$$

$$\begin{aligned}\bar{\mathrm{B}}_{yy}(\xi,\eta,\zeta) = {} & \bar{\mathrm{B}}_{yy}(0) + \bar{\mathrm{B}}^{dev}_{yy,\xi}(0)\,\xi + \bar{\mathrm{B}}^{dev}_{yy,\eta}(0)\,\eta + \bar{\mathrm{B}}^{dev}_{yy,\zeta}(0)\,\zeta \\ & + 2\left[\, \bar{\mathrm{B}}^{dev}_{yy,\xi\eta}(0)\,\xi\eta + \bar{\mathrm{B}}^{dev}_{yy,\eta\zeta}(0)\,\eta\zeta + \bar{\mathrm{B}}^{dev}_{yy,\zeta\xi}(0)\,\zeta\xi \right],\end{aligned} \tag{13.57}$$

$$\begin{aligned}\bar{\mathrm{B}}_{zz}(\xi,\eta,\zeta) = {} & \bar{\mathrm{B}}_{zz}(0) + \bar{\mathrm{B}}^{dev}_{zz,\xi}(0)\,\xi + \bar{\mathrm{B}}^{dev}_{zz,\eta}(0)\,\eta + \bar{\mathrm{B}}^{dev}_{zz,\zeta}(0)\,\zeta \\ & + 2\left[\, \bar{\mathrm{B}}^{dev}_{zz,\xi\eta}(0)\,\xi\eta + \bar{\mathrm{B}}^{dev}_{zz,\eta\zeta}(0)\,\eta\zeta + \bar{\mathrm{B}}^{dev}_{zz,\zeta\xi}(0)\,\zeta\xi \right],\end{aligned} \tag{13.58}$$

$$\bar{\mathrm{B}}_{xy}(\xi,\eta,\zeta) = \bar{\mathrm{B}}_{xy}(0) + \bar{\mathrm{B}}^{dev}_{xy,\zeta}(0)\,\zeta, \tag{13.59}$$

$$\bar{\mathrm{B}}_{yz}(\xi,\eta,\zeta) = \bar{\mathrm{B}}_{yz}(0) + \bar{\mathrm{B}}^{dev}_{yz,\xi}(0)\,\xi, \tag{13.60}$$

$$\bar{\mathrm{B}}_{zx}(\xi,\eta,\zeta) = \bar{\mathrm{B}}_{zx}(0) + \bar{\mathrm{B}}^{dev}_{zx,\eta}(0)\,\eta. \tag{13.61}$$

It is noted that the elements developed above cannot pass the patch test if the elements are skewed. To remedy this drawback, the gradient vectors defined in (13.18)–(13.20) are replaced by the uniform gradient matrices, proposed by Flanagan [1981],

$$\begin{bmatrix} \tilde{\mathrm{b}}_1 \\ \tilde{\mathrm{b}}_2 \\ \tilde{\mathrm{b}}_3 \end{bmatrix} = \frac{1}{V_e}\int_{\Omega_e} \begin{bmatrix} \mathrm{B}_1(\xi,\eta,\zeta) \\ \mathrm{B}_2(\xi,\eta,\zeta) \\ \mathrm{B}_3(\xi,\eta,\zeta) \end{bmatrix} dV, \tag{13.62}$$

where V_e is the element volume and the stabilization vectors are redefined as

$$\tilde{\gamma}_\alpha = \mathrm{h}_\alpha - (\mathrm{h}^t_\alpha\,\mathrm{x}_i)\,\tilde{\mathrm{b}}_i. \tag{13.63}$$

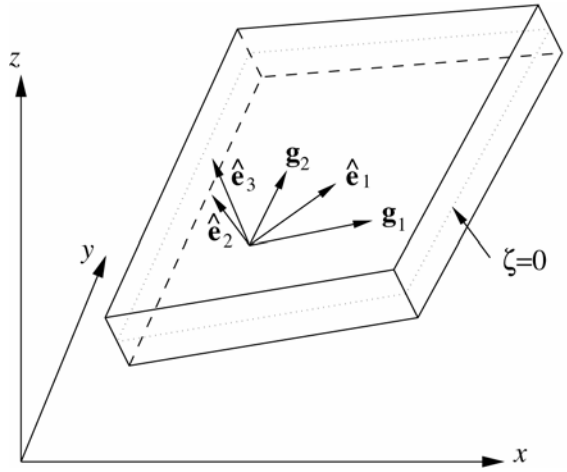


Figure 13.1. Definition of co-rotational coordinate system

The element using the strain submatrices (13.56)-(13.61) and uniform gradient matrices (13.62) with four-point quadrature scheme is called HEXDS element.

13.3.2 Co-rotational Coordinate System

In elements for shell/plate structure simulations, the elimination of the shear locking depends on the proper treatment of the shear strain. It is necessary to attach a local coordinate system to the element so that the strain tensor in this local system is relevant for the treatment. The co-rotational coordinate system determined here is one of the most convenient ways to define such a local system.

A co-rotational coordinate system is defined as a Cartesian coordinate system which rotates with the element. Let $\{x_a, y_a, z_a\}$ denote the current nodal spatial coordinates in the global system. For each quadrature point with natural coordinates (ξ,η,ζ), we can have two tangent directions on the mid-surface $(\zeta=0)$ within the element (see Fig. 13.1)

$$\mathrm{g}_1=\frac{\partial \mathrm{x}}{\partial \xi}=\left[\frac{\partial x}{\partial \xi}\ \frac{\partial y}{\partial \xi}\ \frac{\partial z}{\partial \xi}\right]=\left[N_{a,\xi}x_a\ \ N_{a,\xi}y_a\ \ N_{a,\xi}z_a\right]_{(\xi,\eta,0)}\ , \tag{13.64}$$

$$\mathrm{g}_2=\frac{\partial \mathrm{x}}{\partial \eta}=\left[\frac{\partial x}{\partial \eta}\ \frac{\partial y}{\partial \eta}\ \frac{\partial z}{\partial \eta}\right]=\left[N_{a,\eta}x_a\ \ N_{a,\eta}y_a\ \ N_{a,\eta}z_a\right]_{(\xi,\eta,0)}\ . \tag{13.65}$$

The unit vector $\hat{\mathrm{e}}_1$ of the co-rotational coordinate system is defined as the bisector of the angle intersected by these two tangent vectors g_1 and g_2; the unit vector $\hat{\mathrm{e}}_3$ is perpendicular to the mid-surface and the other unit vector is determined by $\hat{\mathrm{e}}_1$ and $\hat{\mathrm{e}}_3$, i.e.,

$$\hat{\mathrm{e}}_1=\left(\frac{\mathrm{g}_1}{|\mathrm{g}_1|}+\frac{\mathrm{g}_2}{|\mathrm{g}_2|}\right)\Bigg/\left(\left|\frac{\mathrm{g}_1}{|\mathrm{g}_1|}+\frac{\mathrm{g}_2}{|\mathrm{g}_2|}\right|\right), \tag{13.66}$$

$$\hat{e}_3 = \frac{g_1 \times g_2}{|g_1 \times g_2|}, \tag{13.67}$$

$$\hat{e}_2 = \hat{e}_3 \times \hat{e}_1, \tag{13.68}$$

which lead to the transformation matrix

$$R = \begin{bmatrix} \hat{e}_1 \\ \hat{e}_2 \\ \hat{e}_3 \end{bmatrix}. \tag{13.69}$$

13.3.3 Stress and Strain Measures

Since the co-rotational coordinate system rotates with the configuration, the stress defined in this co-rotational system does not change with the rotation or translation of the material body and is thus objective. Therefore, we use the Cauchy stress in the co-rotational coordinate system, called the co-rotational Cauchy stress, as our stress measure.

The rate of deformation (or velocity strain tensor), also defined in the co-rotational coordinate system, is used as the measure of the strain rate,

$$\dot{\varepsilon} = \hat{d} = \frac{1}{2}\left[\frac{\partial \hat{v}^{def}}{\partial \hat{x}} + \left(\frac{\partial \hat{v}^{def}}{\partial \hat{x}} \right)^t \right], \tag{13.70}$$

where $\hat{v}^{def}$ is the deformation part of the velocity in the co-rotational system $\hat{x}$. If the initial strain $\hat{\varepsilon}(X,0)$ is given, the strain tensor can be expressed as,

$$\hat{\varepsilon}(X,t) = \hat{\varepsilon}(X,0) + \int_0^t \hat{d}(X,\tau)\, d\tau. \tag{13.71}$$

The strain increment is then given by the mid-point integration of the velocity strain tensor,

$$\Delta\hat{\varepsilon} = \int_{t_n}^{t_{n+1}} \hat{d}\, d\tau \doteq \frac{1}{2}\left[\frac{\partial \Delta \hat{u}^{def}}{\partial \hat{x}_{n+\frac{1}{2}}} + \left(\frac{\partial \Delta \hat{u}^{def}}{\partial \hat{x}_{n+\frac{1}{2}}} \right)^t \right], \tag{13.72}$$

where $\Delta \hat{u}^{def}$ is the deformation part of the displacement increment in the co-rotational system $\hat{x}_{n+\frac{1}{2}}$ referred to the mid-point configuration.

13.3.4 Co-rotational Stress and Strain Updates

For stress and strain updates, we assume that all variables at the previous time step t_n are known. Since the stress and strain measures defined in the earlier section are objective in the co-rotational system, we only need to calculate the strain increment from the displacement field within the time increment $\left[t_n, t_{n+1}\right]$. The stress is then updated by using the radial return algorithm.

All the kinematical quantities must be computed from the last time step configuration, Ω_n, at $t=t_n$ and the current configuration, Ω_{n+1} at $t=t_{n+1}$ since these are the only available data. Denoting the spatial coordinates of these two configurations as x_n and x_{n+1} in the fixed global Cartesian coordinate system $O\mathrm{x}$, as shown in Figure 13.2, the coordinates in the corresponding co-rotational Cartesian coordinate systems, $O\hat{\mathrm{x}}_n$ and $O\hat{\mathrm{x}}_{n+1}$, can be obtained by the following transformation rules:

$$\hat{\mathrm{x}}_n = \mathrm{R}_n \mathrm{x}_n \ , \tag{13.73}$$

$$\hat{\mathrm{x}}_{n+1} = \mathrm{R}_{n+1} \mathrm{x}_{n+1} \ , \tag{13.74}$$

where R_n and R_{n+1} are the orthogonal transformation matrices which rotate the global coordinate system to the corresponding co-rotational coordinate systems, respectively.

Since the strain increment is referred to the configuration at $t=t_{n+\frac{1}{2}}$, by assuming the velocities within the time increment $\left[t_n, t_{n+1}\right]$ are constant, we have

$$x_{n+\frac{1}{2}} = \frac{1}{2}\left(x_n + x_{n+1}\right) , \tag{13.75}$$

and the transformation to the co-rotational system associated with this mid-point configuration, $\Omega_{n+\frac{1}{2}}$, is given by

$$\hat{\mathrm{x}}_{n+\frac{1}{2}} = \mathrm{R}_{n+\frac{1}{2}} \mathrm{x}_{n+\frac{1}{2}} \ . \tag{13.76}$$

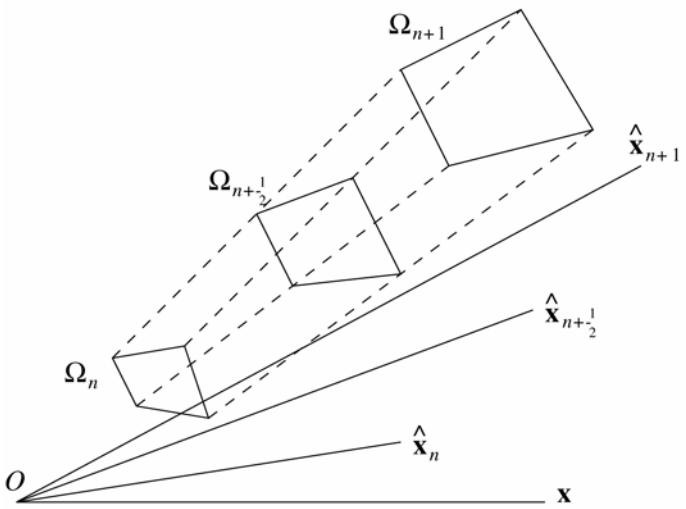


Figure 13.2. Configurations at times $t_n, t_{n+\frac{1}{2}}$ and t_{n+1},

Similar to polar decomposition, an incremental deformation can be separated into the summation of a pure deformation and a pure rotation [Belytschko, 1973]. Letting $\Delta \mathrm{u}$ indicate the displacement increment within the time increment $\left[t_n, t_{n+\frac{1}{2}}\right]$, we write

$$\Delta \mathrm{u} = \Delta \mathrm{u}^{def} + \Delta \mathrm{u}^{rot} \ , \tag{13.77}$$

where $\Delta \mathrm{u}^{def}$ and $\Delta \mathrm{u}^{rot}$ are, respectively, the deformation part and the pure rotation part of the displacement increment in the global coordinate system. The deformation part also includes the translation displacements which cause no strains.

In order to obtain the deformation part of the displacement increment referred to the configuration at $t = t_{n+\frac{1}{2}}$, we need to find the rigid rotation from Ω_n to Ω_{n+1} provided that the mid-point configuration, $\Omega_{n+\frac{1}{2}}$, is held still. Defining two virtual configurations, Ω'_n and Ω'_{n+1}, by rotating the element bodies Ω_n and Ω_{n+1} into the co-rotational system $O\hat{\mathrm{x}}_{n+\frac{1}{2}}$ (Fig. 13.3) and denoting and $\hat{\mathrm{x}}'_{n+1}$ as the coordinates of Ω'_n and Ω'_{n+1} in the co-rotational system $O\hat{\mathrm{x}}_{n+\frac{1}{2}}$, we have

$$\hat{\mathrm{x}}'_n = \hat{\mathrm{x}}_n, \quad \hat{\mathrm{x}}'_{n+1} = \hat{\mathrm{x}}_{n+1} \ . \tag{13.78}$$

We can see that from Ω_n to Ω'_n and from Ω'_{n+1} to Ω_{n+1}, the body experiences two rigid rotations and the rotation displacements are given by

$$\Delta \mathrm{u}_1^{rot} = \mathrm{x}'_n - \mathrm{x}_n = \mathrm{R}^t_{n+\frac{1}{2}} \hat{\mathrm{x}}'_n - \mathrm{x}_n = \mathrm{R}^t_{n+\frac{1}{2}} \hat{\mathrm{x}}_n - \mathrm{x}_n \ , \tag{13.79}$$

$$\Delta \mathrm{u}_2^{rot} = \mathrm{x}_{n+1} - \mathrm{x}'_{n+1} = \mathrm{x}_{n+1} - \mathrm{R}^t_{n+\frac{1}{2}} \hat{\mathrm{x}}'_{n+1} = \mathrm{x}_{n+1} - \mathrm{R}^t_{n+\frac{1}{2}} \hat{\mathrm{x}}_{n+1} \ . \tag{13.80}$$

Thus the total rotation displacement increment can be expressed as

$$\begin{aligned}\Delta \mathrm{u}^{rot} &= \Delta \mathrm{u}_1^{rot} + \Delta \mathrm{u}_2^{rot} = \mathrm{x}_{n+1} - \mathrm{x}_n - \mathrm{R}_{n+\frac{1}{2}}^{t}\left(\hat{\mathrm{x}}_{n+1} - \hat{\mathrm{x}}_n\right) \\ &= \Delta \mathrm{u} - \mathrm{R}_{n+\frac{1}{2}}^{t}\left(\hat{\mathrm{x}}_{n+1} - \hat{\mathrm{x}}_n\right).\end{aligned} \tag{13.81}$$

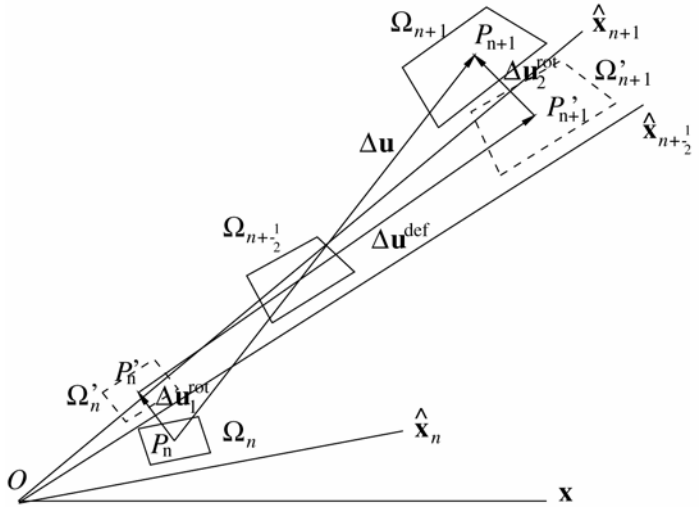


Figure 13.3. Separation of the displacement increment

Then the deformation part of the displacement increment referred to the configuration $\Omega_{n+\frac{1}{2}}$ is

$$\Delta \mathrm{u}^{def} = \Delta \mathrm{u} - \Delta \mathrm{u}^{rot} = \mathrm{R}_{n+\frac{1}{2}}^{t}\left(\hat{\mathrm{x}}_{n+1} - \hat{\mathrm{x}}_n\right). \tag{13.82}$$

Therefore, the deformation displacement increment in the co-rotational coordinate system $O\hat{\mathrm{x}}_{n+\frac{1}{2}}$ is obtained as

$$\Delta \hat{\mathrm{u}}^{def} = \mathrm{R}_{n+\frac{1}{2}} \Delta \mathrm{u}^{def} = \hat{\mathrm{x}}_{n+1} - \hat{\mathrm{x}}_n \,. \tag{13.83}$$

Once the strain increment is obtained by equation (13.72), the stress increment, also referred to the mid-point Configuration, can be calculated with the radial return algorithm. The total strain and stress can then be updated as

$$\hat{\varepsilon}_{n+1} = \hat{\varepsilon}_n + \Delta\hat{\varepsilon}\,, \tag{13.84}$$

$$\hat{\sigma}_{n+1} = \hat{\sigma}_n + \Delta\hat{\sigma}\,. \tag{13.85}$$

Note that the resultant stress and strain tensors are both referred to the current configuration and defined in the current co-rotational coordinate system. By using the tensor transformation rule we can have the strain and stress components in the global coordinate system.

13.3.5 Tangent Stiffness Matrix and Nodal Force Vectors

From the Hu-Washizu variational principle, at both vth and $(v+1)$th iteration, we have

$$\int_{\hat{\Omega}^v} \delta\hat{\varepsilon}_{ij}^v \hat{\sigma}_{ij}^v dV = \delta\hat{\pi}_{\text{ext}}^v \ , \tag{13.86}$$

$$\int_{\hat{\Omega}^{v+1}} \delta\hat{\varepsilon}_{ij}^{v+1} \hat{\sigma}_{ij}^{v+1} dV = \delta\hat{\pi}_{\text{ext}}^{v+1} \ , \tag{13.87}$$

where $\delta\hat{\pi}_{\text{ext}}$ is the virtual work done by the external forces. Note that both equations are written in the co-rotational coordinate system defined in the vth iterative configuration given by x_{n+1}^v. The variables in this section are within the time step $\left[t_n, t_{n+\frac{1}{2}}\right]$ and superscripts indicate the number of iterations.

Assuming that all external forces are deformation-independent, linearization of Equation (13.87) gives [Liu, 1992]

$$\int_{\hat{\Omega}^v} \delta\hat{u}_{i,j}^v \hat{C}_{ijkl}^v \Delta\hat{u}_{k,l} dV + \int_{\hat{\Omega}^v} \delta\hat{u}_{i,j}^v \hat{T}_{ijkl}^v \Delta\hat{u}_{k,l} dV = \delta\hat{\pi}_{\text{ext}}^{v+1} - \delta\hat{\pi}_{\text{ext}}^v \ , \tag{13.88}$$

where the Green-Naghdi rate of Cauchy stress tensor is used, i.e.,

$$\hat{T}_{ijkl}^v = \delta_{ik} \hat{\sigma}_{jl}^v \ . \tag{13.89}$$

The first term on the left hand side of (13.88) denotes the material response since it is due to pure deformation or stretching; the second term is an initial stress part resulting from finite deformation effect.

Taking account of the residual of the previous iteration, Equation (13.87) can be approximated as

$$\int_{\hat{\Omega}^v} \delta\hat{u}_{i,j}^v \left(\hat{C}_{ijkl}^v + \hat{T}_{ijkl}^v\right) \Delta\hat{u}_{k,l} \, dV = \delta\hat{\pi}_{\text{ext}}^{v+1} - \int_{\hat{\Omega}^v} \delta\hat{\varepsilon}_{ij}^v \, \hat{\sigma}_{ij}^v \, dV \, . \tag{13.90}$$

If the strain and stress vectors are defined as

$$\varepsilon^t = \left[\varepsilon_x \ \varepsilon_y \ \varepsilon_z \ 2\varepsilon_{xy} \ 2\varepsilon_{yz} \ 2\varepsilon_{zx} \ 2\omega_{xy} \ 2\omega_{yz} \ 2\omega_{zx} \ \right] , \tag{13.91}$$

$$\sigma^t = \left[\sigma_x \ \sigma_y \ \sigma_z \ \sigma_{xy} \ \sigma_{yz} \ \sigma_{zx} \right] , \tag{13.92}$$

We can rewrite equation (13.90) as

$$\int_{\hat{\Omega}^v} \delta\hat{\varepsilon}_i^v \left(\hat{C}_{ij}^v + \hat{T}_{ij}^v\right) \delta\hat{\varepsilon}_j \, dV = \delta\hat{\pi}_{\text{ext}}^{v+1} - \int_{\hat{\Omega}^v} \delta\hat{\varepsilon}_i^v \, \hat{\sigma}_j^v \, dV \, , \tag{13.93}$$

where $\hat{C}_{ij}^{v}$ is the consistent tangent modulus tensor corresponding to pure deformation (see Section 3.2.3) but expanded to a 9 by 9 matrix; $\hat{T}_{ij}^{v}$ is the geometric stiffness matrix which is given as follows [Liu 1992]:

$$\mathrm{T}=\begin{bmatrix}
\sigma_1 & 0 & 0 & \frac{\sigma_4}{2} & 0 & \frac{\sigma_6}{2} & \frac{\sigma_4}{2} & 0 & -\frac{\sigma_6}{2} \\
 & \sigma_2 & 0 & \frac{\sigma_4}{2} & \frac{\sigma_5}{2} & 0 & -\frac{\sigma_4}{2} & \frac{\sigma_5}{2} & 0 \\
 & & \sigma_3 & 0 & \frac{\sigma_5}{2} & \frac{\sigma_6}{2} & 0 & -\frac{\sigma_5}{2} & \frac{\sigma_6}{2} \\
 & & & \frac{\sigma_1+\sigma_2}{4} & \frac{\sigma_6}{4} & \frac{\sigma_5}{4} & \frac{\sigma_2-\sigma_1}{4} & \frac{\sigma_6}{4} & -\frac{\sigma_5}{4} \\
 & & & & \frac{\sigma_2+\sigma_3}{4} & \frac{\sigma_4}{4} & -\frac{\sigma_6}{4} & \frac{\sigma_3-\sigma_2}{4} & \frac{\sigma_4}{4} \\
 & & \text{symm.} & & & \frac{\sigma_1+\sigma_3}{4} & \frac{\sigma_5}{4} & -\frac{\sigma_4}{4} & \frac{\sigma_1-\sigma_3}{4} \\
 & & & & & & \frac{\sigma_1+\sigma_2}{4} & -\frac{\sigma_6}{4} & -\frac{\sigma_5}{4} \\
 & & & & & & & \frac{\sigma_2+\sigma_3}{4} & -\frac{\sigma_4}{4} \\
 & & & & & & & & \frac{\sigma_3+\sigma_1}{4}
\end{bmatrix}. \qquad (13.94)$$

By interpolation

$$\Delta \mathrm{u}=\mathrm{N}\Delta \mathrm{d}, \qquad \delta \mathrm{u}=\mathrm{N}\delta \mathrm{d}; \qquad (13.95)$$

$$\Delta\varepsilon=\overline{\mathrm{B}}\Delta \mathrm{d}, \qquad \delta\varepsilon=\overline{\mathrm{B}}\delta \mathrm{d}, \qquad (13.96)$$

where N and $\overline{\mathrm{B}}$ are, respectively, the shape functions and strain operators defined in Section 2. This leads to a set of equations

$$\hat{\mathrm{K}}^{v}\Delta\hat{\mathrm{d}}=\hat{\mathrm{r}}^{v+1}=\hat{\mathrm{f}}_{\mathrm{ext}}^{v+1}-\hat{\mathrm{f}}_{\mathrm{int}}^{v}\ , \qquad (13.97)$$

where the tangent stiffness matrix, $\hat{\mathrm{K}}^{v}$, and the internal nodal force vector, $\hat{\mathrm{f}}_{\mathrm{int}}^{v}$, are

$$\hat{\mathrm{K}}^{v}=\int_{\hat{\Omega}^{v}}\hat{\overline{\mathrm{B}}}^{t}\left(\hat{\mathrm{C}}^{v}+\hat{\mathrm{T}}^{v}\right)\hat{\overline{\mathrm{B}}}\,dV\ , \qquad (13.98)$$

$$\hat{\mathrm{f}}_{\mathrm{int}}^{v}=\int_{\hat{\Omega}^{v}}\hat{\overline{\mathrm{B}}}^{t}\ \hat{\sigma}^{v}\,dV\ . \qquad (13.99)$$

The tangent stiffness and nodal force are transformed into the global coordinate system tensorially as

$$\mathrm{K}^{v} = \mathrm{R}^{vt}\,\hat{\mathrm{K}}^{v}\,\mathrm{R}^{v}\ , \tag{13.100}$$

$$\mathrm{r}^{v+1} = \mathrm{R}^{vt}\,\hat{\mathrm{r}}^{v}_{\mathrm{int}}\ , \tag{13.101}$$

where R^{v} is the transformation matrix of the co-rotational system defined by x^{v}_{n+1}. Finally, we get a set of linear algebraic equations

$$\mathrm{K}^{v}\,\Delta\mathrm{d}^{v+1} = \mathrm{r}^{v+1}\ . \tag{13.102}$$

13.4 Numerical Examples

To investigate the performance of the element introduced in this paper, a variety of problems including linear elastic and nonlinear elastic-plastic/large deformation problems are studied. Since the element is developed to avoid locking, the applicability to problems of thin structures is studied by solving the standard test problems including pinched cylinder and Scordelis-Lo roof, which are proposed by MacNeal, 1985 and Belytschko, 1984b. Also a sheet metal forming problem is solved to test and demonstrate the effectiveness and efficiency of this element.

13.4.1 Timoshenko Cantilever Beam

The first problem is a linear, elastic cantilever beam with a load at its end as shown in Fig. 13.4, where M and P at the left end of the cantilever are reactions at the support. The analytical solution from Timoshenko, 1970 is

$$u_x(x,y) = \frac{-Py}{6\bar{E}I}\left[(6L-3x)x + (2+\bar{\nu})\left(y^2 - \tfrac{1}{4}D^2\right)\right], \tag{13.103}$$

$$u_y(x,y) = \frac{P}{6\bar{E}I}\left[3\bar{\nu}y^2(L-x) + \frac{1}{4}(4+5\bar{\nu})D^2x + (3L-x)x^2\right], \tag{13.104}$$

where

$$I = \frac{1}{12}D^3\ ,$$

$$\bar{E} = \begin{cases} E, & \\ E/(1-\nu^2), & \end{cases} \qquad \bar{\nu} = \begin{cases} \nu, & \text{for plane stress;} \\ \nu/(1-\nu), & \text{for plane strain.} \end{cases}$$

The displacements at the support end, $x=0$, $-\frac{1}{2}D \le y \le \frac{1}{2}D$ are nonzero except at the top, bottom and midline (as shown in Fig. 13.5). Reaction forces are applied at the support based on the stresses corresponding to the displacement field at $x=0$, which are

$$\sigma_{xx} = -\frac{Py}{I}(L-x), \qquad \sigma_{yy} = 0, \qquad \sigma_{xy} = \frac{P}{2I}\left(\frac{1}{4}D^2 - y^2\right). \qquad (13.105)$$

The distribution of the applied load to the nodes at $x = L$ is also obtained from the closed-form stress fields.

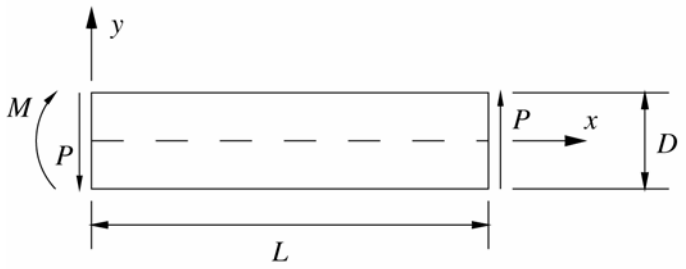
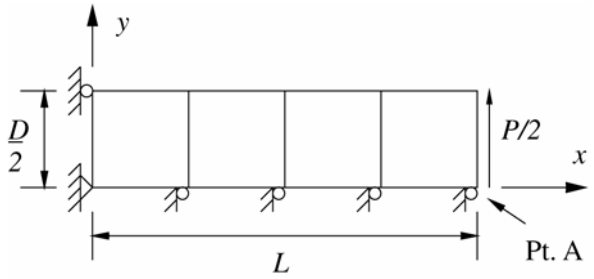
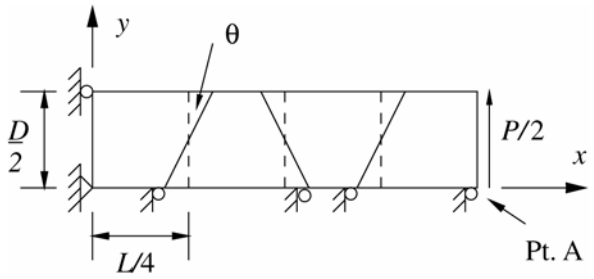


Figure 13.4. Timoshenko cantilever beam.



(a) Regular mesh



(b) Skewed mesh

Figure 13.5. Top half of anti-symmetric beam mesh

The parameters for the cantilever beam are: $L = 1.0, D = 0.02, P = 2.0, E = 1\_\times 10^7$; and two values of Poisson's ratio: *(1)* $\nu = 0.25$, *(2)* $\nu = 0.4999$.

Since the problem is anti-symmetric, only the top half of the beam is modeled. Plane strain conditions are assumed in the z-direction and only one layer of elements is used in this direction. Both regular mesh and skewed mesh are tested for this problem.

Normalized vertical displacements at point A for each case are given in Table 13.1. Tables 13.1a and 13.1b show the normalized displacement at point A for the regular mesh. There is no shear or volumetric locking for this element. For the skewed mesh, with the skewed angle increased, we need more elements to get more accurate solution (Table 13.1c).

(a) $\nu = 0.25$, regular mesh

Analytical solution $w_A = 9.3777 \times 10^{-2}$

Mesh	$4 \times 1 \times 1$	$8 \times 1 \times 1$	$8 \times 2 \times 1$
HEXDS	1.132	1.142	1.029

(b) $\nu = 0.4999$, regular mesh

Analytical solution $w_A = 7.5044 \times 10^{-2}$

Mesh	$4 \times 1 \times 1$	$8 \times 1 \times 1$	$8 \times 2 \times 1$
HEXDS	1.182	1.197	1.039

(c) $\nu = 0.25$, skewed mesh

θ	1°	5°	10°
$4 \times 1 \times 1$	1.078	0.580	0.317
$8 \times 1 \times 1$	1.136	0.996	0.737
$16 \times 1 \times 1$	1.142	1.090	0.955

Table 13.1. Normalized displacement at point A of cantilever beam.

13.4.2 Pinched Cylinder

Figure 13.6 shows a pinched cylinder subjected to a pair of concentrated loads. Two cases are studied in this example. In the first case, both ends of the cylinder are assumed to be free. In the second case, both ends of the cylinder are covered with rigid diaphragms so that only the displacement in the axial direction is allowed at the ends. The parameters for the first case

(without diaphragms) are

$$E = 1.05 \times 10^6, \nu = 0.3125, L = 10.35, R = 1.0, t = 0.094, P = 100.0;$$

while for the second case (with diaphragms), the parameters are set to be

$$E = 3 \times 10^6, \nu = 0.3, L = 600.0, R = 300.0, t = 3.0, P = 1.0.$$

Due to symmetry only one octant of the cylinder is modeled. The computed displacements at the loading point are compared to the analytic solutions in Table 13.2. HEXDS element works well in both cases, indicating that this element can avoid not only shear locking but also membrane locking; this is not unexpected since membrane locking occurs primarily in curved elements [Stolarski, 1983].

13.4.3 Scordelis-Lo Roof

Scordelis-Lo roof subjected to its own weight is shown in Figure 13.7. Both ends of the roof are assumed to be covered with rigid diaphragms. The parameters are selected to be: $E=4.32\times10^8$, $v=0.0$, $L=50.0$, $R=25.0$, $t=0.25$, $\theta=40^\circ$, and the gravity is 360.0 per volume.

(a) First case without diaphragms

Analytical solution $w_{max}=0.1137$

Mesh	$10\times10\times2$	$16\times16\times4$	$20\times20\times4$
HEXDS	1.106	1.054	1.067

(b) Second case with diaphragms

Analytical solution $w_{max}=1.8248\times10^{-5}$

Mesh	$10\times10\times2$	$16\times16\times4$	$20\times20\times4$
HEXDS	0.801	0.945	0.978

Table 13.2. Normalized displacement at loading point of pinched cylinder

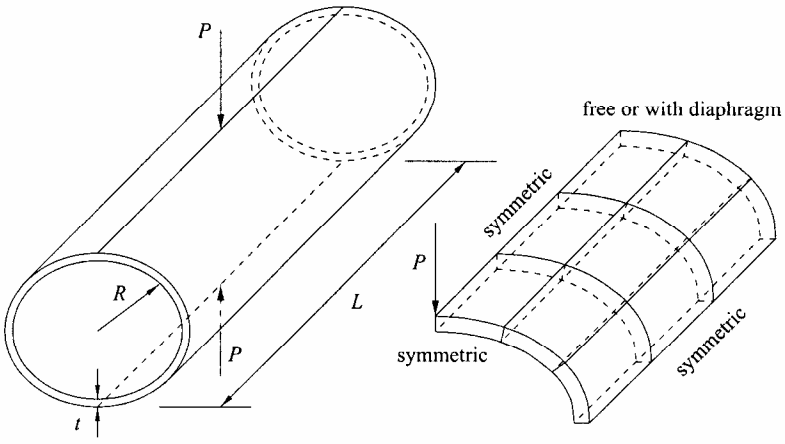


Figure 13.6. Pinched cylinder and the element model

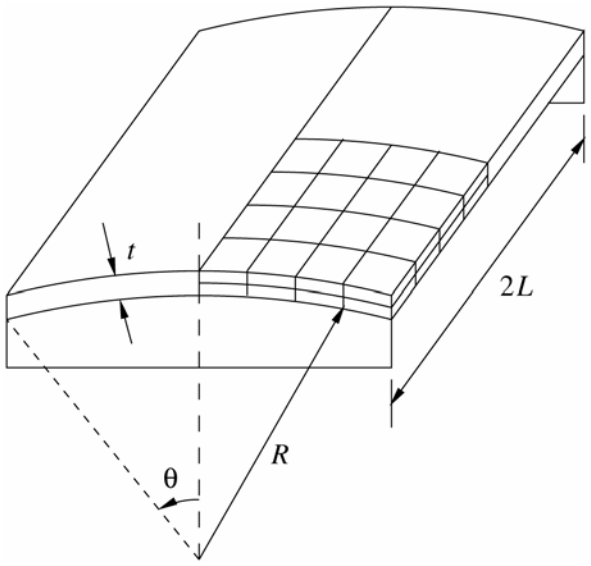


Figure 13.7. Scordelis-Lo roof under self-weight

Due to symmetry only one quarter of the roof is modeled. The computed displacement at the midpoint of the edge is compared to the analytic solution in Table 13.3. In this example the HEXDS element can get good result with 100×2 elements.

Analytical solution w_{max} = 0.3024

Mesh	$8 \times 8 \times 1$	$16 \times 16 \times 1$	$32 \times 32 \times 1$	$10 \times 10 \times 2$
HEXDS	1.157	1.137	1.132	1.045

Table 13.3. Normalized displacement at mid-edge of Scordelis-Lo roof

13.4.4 Circular Sheet Stretched with a Tight Die

A circular sheet is stretched under a hemisphere punch and a tight die with a small corner radius (Fig. 13.8). The material is elastoplastic with nonlinear hardening rule. The elastic material constants are: $E = 206\,\text{GPa}$ and $\nu = 0.3$. In the plastic range, the uniaxial stress-strain curve is given by

$$\sigma = K\varepsilon^n,$$

where $K = 509.8$ MPa, $n = 0.21$, σ is Cauchy stress and ε is natural strain (logarithmic strain). The initial yield stress is obtained to be $\sigma_0 = 103.405$ Mpa and the tangent modulus at the initial yield point is $E_t = 0.4326 \times 10^5$ MPa.

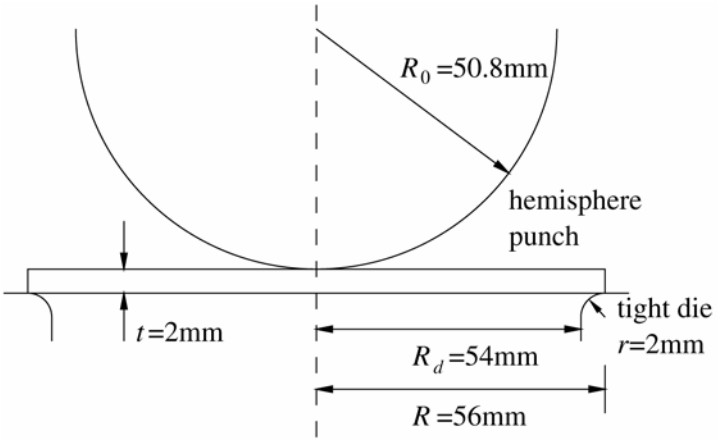


Figure 13.8. Circular sheet stretched with a tight die

Because of the small corner radius of the die, the same difficulties as in the problem of sheet stretch under the rigid cylinders lead the shell elements to failure in this problem. Three-dimensional solid elements are needed and fine meshes should be put in the areas near the center and the edge of the sheet.

One quarter of the sheet is modeled with 1400×2 HEXDS elements due to the double symmetries. The mesh is shown in Fig. 13.9. Two layers of elements are used in the thickness. Around the center and near the circular edge of the sheet, fine mesh is used. The nodes on the edge are fixed in x- and y-directions and the bottom nodes on the edge are prescribed in three directions. No friction is considered in this simulation. For comparison, the axisymmetric four-node element with reduced integration (CAX4R) is also used and the mesh for this element is the same as shown in the top of Figure 13.9.

The results presented here are after the punch has traveled down 50 mm. The profile of the circular sheet is shown in Figure 13.10 where we can see that the sheet under the punch experiences most of the stretching and the thickness of the sheet above the die changes a lot. The deformation between the punch and the die is small. However, the sheet thickness obtained by the CAX4R element is less than that by the HEXDS element and there is slight difference above the die. These observations can be verified by the strain distributions in the sheet along the radial direction (Figure 13.12). The direction of the radial strain is the tangent of the mid-surface of the element in the *rz* plane and the thickness strain is in the direction perpendicular to the mid-surface of the element. The unit vector of the circumferential strain is defined as the cross-product of the directional cosine vectors of the radial strain and the thickness strain. We can see that the CAX4R element yields larger strain components in the area under the punch than the HEXDS element. The main difference of the strain distributions in the region above the die is that the CAX4R element gives zero circumferential strain in this area but the HEXDS element yields non-zero strain. The value of the reaction force shown in the Figure 13.11 is only one quarter of the total punch reaction force since only one quarter of the sheet is modeled. From this figure we can see that the sheet begins softening after the punch travels down about 45 mm, indicating that the sheet may have necking though this cannot be seen clearly from Figure 13.10.

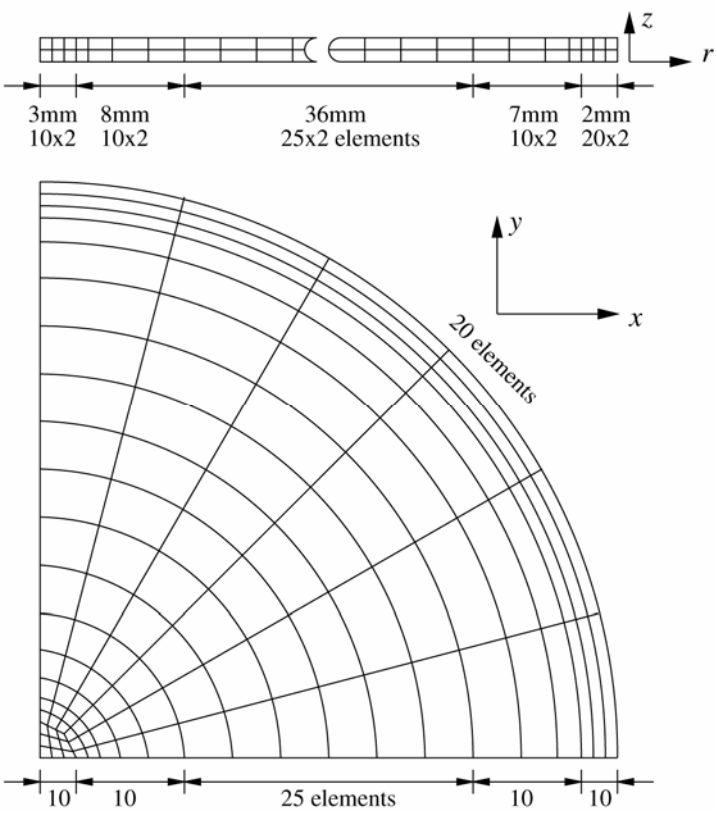


Figure 13.9. Mesh for circular sheet stretching

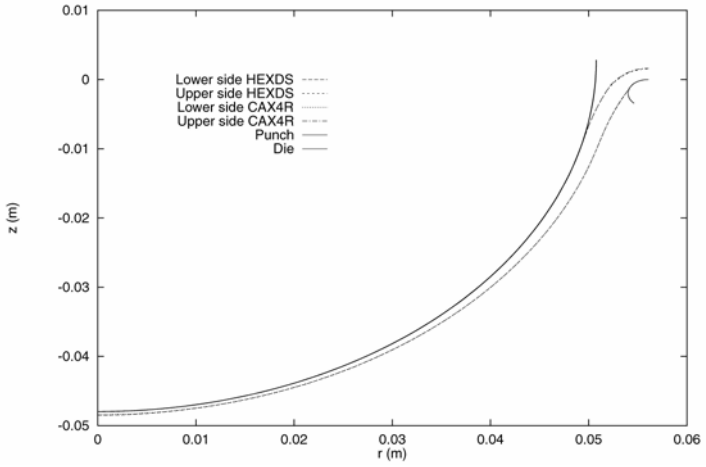


Figure 13.10. Deformed shape of a circular sheet with punch travel 50 mm

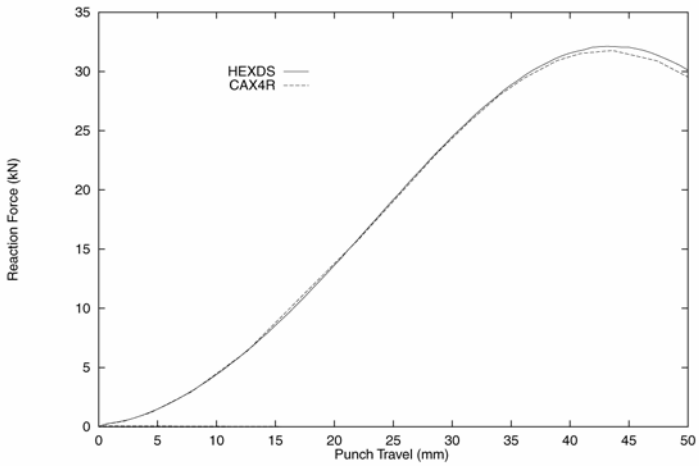
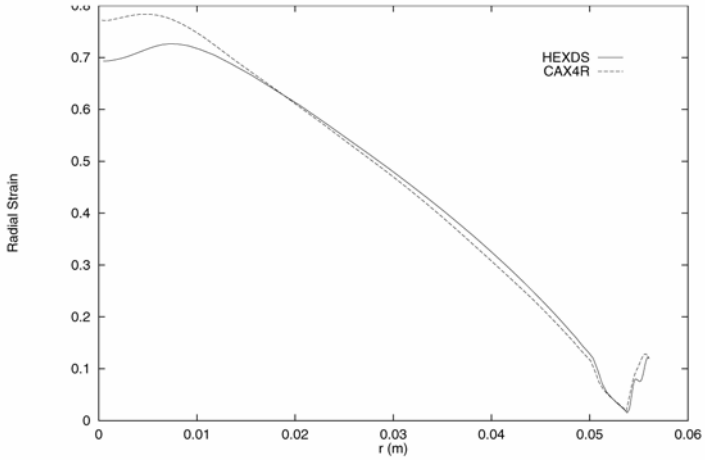
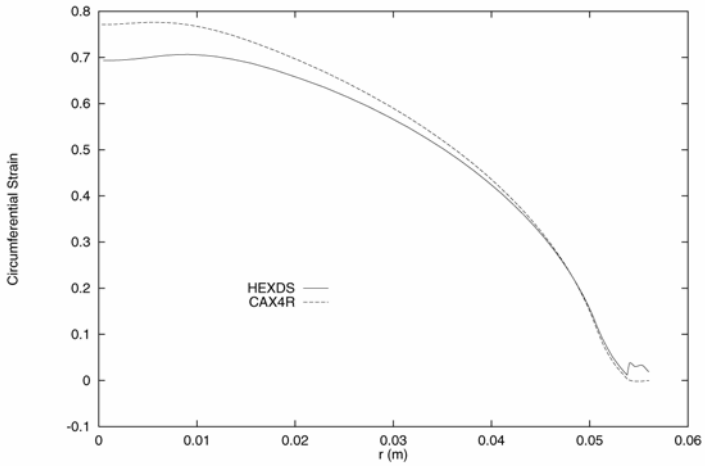


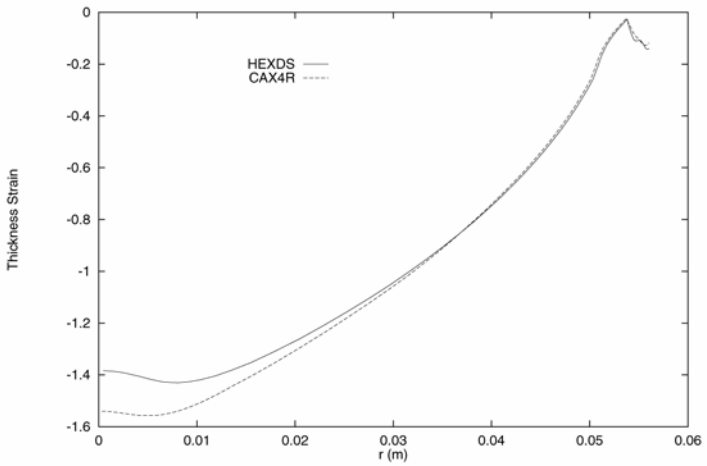
Figure 13.11. Reaction force vs. punch travel for the circular sheet



(a) Radial strain distribution



(b) Circumferential strain distribution



(c) Thickness strain distribution

Figure 13.12. Strain distributions in circular sheet with punch travel 50 mm

13.5 Conclusions

A new eight-node hexahedral element is implemented for the large deformation elastic-plastic analysis. Formulated in the co-rotational coordinate system, this element is shown to be effective and efficient and can achieve fast convergence in solving a wide variety of nonlinear problems.

By using a co-rotational system which rotates with the element, the locking phenomena can be suppressed by omitting certain terms in the generalized strain operators. In addition, the integration of the constitutive equation in the co-rotational system takes the same simple form as small deformation theory since the stress and strain tensors defined in this co-rotational system are objective.

Radial return algorithm is used to integrate the rate-independent elastoplastic constitutive equation. The tangent stiffness matrix consistently derived from this integration scheme is crucial to preserve the second order convergence rate of the Newton's iteration method for the nonlinear static analyses.

Test problems studied in this paper demonstrate that the element is suitable to continuum and structural numerical simulations. In metal sheet forming analysis, this element has advantages over shell elements for certain problems where through the thickness deformation and strains are significant.

14. TRUSS ELEMENT

One of the simplest elements is the pin-jointed truss element shown in Figure 14.1. This element has three degrees of freedom at each node and carries an axial force. The displacements and velocities measured in the local system are interpolated along the axis according to

$$u = u_1 + \frac{x}{L}(u_2 - u_1) \tag{14.1}$$

$$\dot{u} = \dot{u}_1 + \frac{x}{L}(\dot{u}_2 - \dot{u}_1) \tag{14.2}$$

where at $x=0$, $u = u_1$ and at $x = L$, $u = u_2$. Incremental strains are found from

$$\Delta\varepsilon = \frac{(\dot{u}_2 - \dot{u}_1)}{L}\Delta t \tag{14.3}$$

and are computed in LS-DYNA using

$$\Delta\varepsilon^{n+1/2} = \frac{2\left(\dot{u}_2^{n+1/2} - \dot{u}_1^{n+1/2}\right)}{L^n + L^{n+1}}\Delta t^{n+1/2} \tag{14.4}$$

The normal force N is then incrementally updated using a tangent modulus E^t according to

$$N^{n+1} = N^n + AE^t \Delta\varepsilon^{n+1/2} \tag{14.5}$$

Two constitutive models are implemented for the truss element: elastic and elastic-plastic with kinematic hardening.

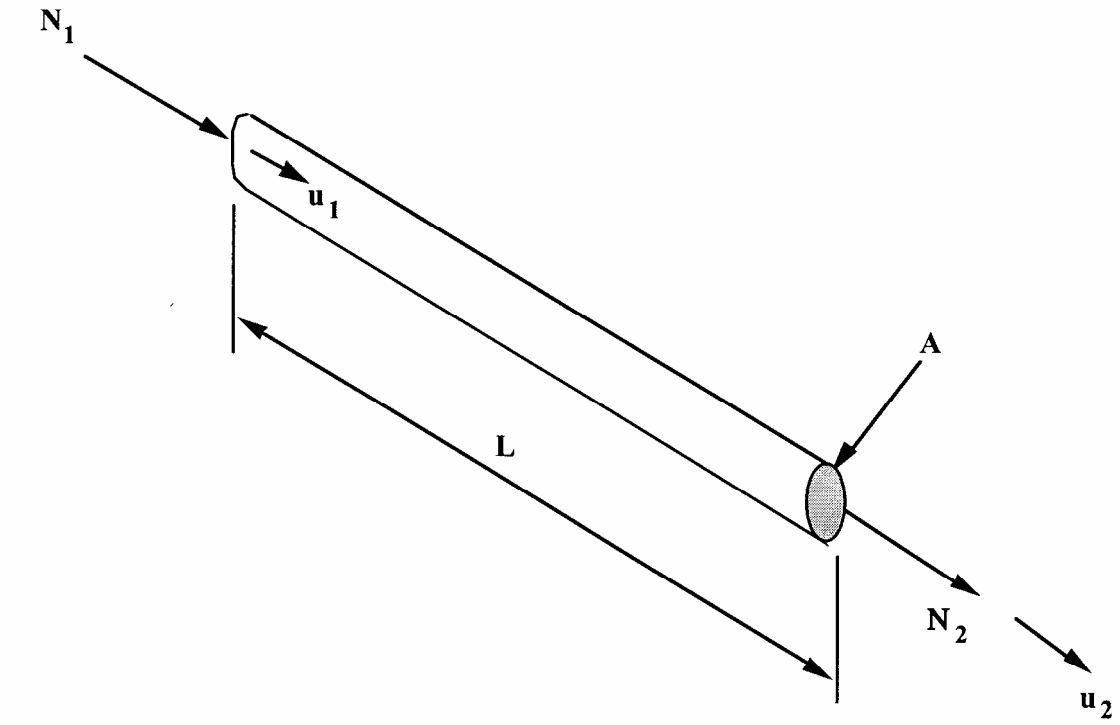


Figure 14.1. Truss element.

15. MEMBRANE ELEMENT

The Belytschko-Lin-Tsay shell element {Belytschko and Tsay [1981], Belytschko et al., [1984a]} is the basis for this very efficient membrane element. In this section we briefly outline the theory employed which, like the shell on which it is based, uses a combined co-rotational and velocity-strain formulation. The efficiency of the element is obtained from the mathematical simplifications that result from these two kinematical assumptions. The co-rotational portion of the formulation avoids the complexities of nonlinear mechanics by embedding a coordinate system in the element. The choice of velocity strain or rate of deformation in the formulation facilitates the constitutive evaluation, since the conjugate stress is the more familiar Cauchy stress.

In membrane elements the rotational degrees of freedom at the nodal points may be constrained, so that only the translational degrees-of-freedom contribute to the straining of the membrane. A triangular membrane element may be obtained by collapsing adjacent nodes of the quadrilateral.

15.1 Co-rotational Coordinates

The mid-surface of the quadrilateral membrane element is defined by the location of the element's four corner nodes. An embedded element coordinate system (Figure 7.1) that deforms with the element is defined in terms of these nodal coordinates. The co-rotational coordinate system follows the development in Section 7, Equations (7.1)—(7.3).

15.2 Velocity-Strain Displacement Relations

The co-rotational components of the velocity strain (rate of deformation) are given by:

$$\hat{d}_{ij} = \frac{1}{2}\left(\frac{\partial \hat{v}_i}{\partial \hat{x}_j} + \frac{\partial \hat{v}_j}{\partial \hat{x}_i} \right) \tag{7.7}$$

The above velocity-strain relations are evaluated only at the center of the shell. Standard bilinear nodal interpolation is used to define the mid-surface velocity, angular velocity, and the element's coordinates (isoparametric representation). These interpolation relations are given by

$$v^m = N_I(\xi,\eta) v_I \tag{7.9a}$$

$$x^m = N_I(\xi,\eta) x_I \tag{7.9c}$$

where the subscript I is summed over all the element's nodes and the nodal velocities are obtained by differentiating the nodal coordinates with respect to time, i.e., $v_I = \dot{x}_I$. The bilinear shape functions are defined in Equations (7.10).

The velocity strains at the center of the element, i.e., at $\xi = 0$, and $\eta = 0$, are obtained as in Section 7 giving:

$$\hat{d}_x = B_{1I} \hat{v}_{xI} \tag{7.11a}$$

$$\hat{d}_y = B_{2I}\hat{v}_{yI} \tag{7.11b}$$

$$2\hat{d}_{xy} = B_{2I}\hat{v}_{xI} + B_{1I}\hat{v}_{yI} \tag{7.11c}$$

where

$$B_{1I} = \frac{\partial N_I}{\partial \hat{x}} \tag{7.12a}$$

$$B_{2I} = \frac{\partial N_I}{\partial \hat{y}} \tag{7.12b}$$

15.3 Stress Resultants and Nodal Forces

After suitable constitutive evaluations using the above velocity strains, the resulting stresses are multiplied by the thickness of the membrane, h, to obtain local resultant forces. Therefore,

$$\hat{f}^R_{\alpha\beta} = h\hat{\sigma}_{\alpha\beta} \tag{15.1}$$

where the superscript *R* indicates a resultant force and the Greek subscripts emphasize the limited range of the indices for plane stress plasticity.

The above element centered force resultants are related to the local nodal forces by

$$\hat{f}_{xI} = A\left(B_{1I}\,\hat{f}^R_{xx} + B_{2I}\,\hat{f}^R_{xy}\right) \tag{7.14a}$$

$$\hat{f}_{yI} = A\left(B_{2I}\,\hat{f}^R_{yy} + B_{1I}\,\hat{f}^R_{xy}\right) \tag{7.14b}$$

where A is the area of the element.

The above local nodal forces are then transformed to the global coordinate system using the transformation relations given in Equation (7.5a).

15.4 Membrane Hourglass Control

Hourglass deformations need to be resisted for the membrane element. The hourglass control for this element is discussed in Section 7.4.

16. DISCRETE ELEMENTS AND MASSES

The discrete elements and masses in LS-DYNA provide a capability for modeling simple spring-mass systems as well as the response of more complicated mechanisms. Occasionally, the response of complicated mechanisms or materials needs to be included in LS-DYNA models, e.g., energy absorbers used in passenger vehicle bumpers. These mechanisms are often experimentally characterized in terms of force-displacement curves. LS-DYNA provides a selection of discrete elements that can be used individually or in combination to model complex force-displacement relations.

The discrete elements are assumed to be massless. However, to solve the equations of motion at unconstrained discrete element nodes or nodes joining multiple discrete elements, nodal masses must be specified at these nodes. LS-DYNA provides a direct method for specifying these nodal masses in the model input.

All of the discrete elements are two-node elements, i.e., three-dimensional springs or trusses. A discrete element may be attached to any of the other LS-DYNA continuum, structural, or rigid body element. The force update for the discrete elements may be written as

$$\hat{f}^{\,i+1} = \hat{f}^{\,i} + \Delta \hat{f} \tag{16.1}$$

where the superscript $i + 1$ indicates the time increment and the superposed caret $(\hat{})$ indicates the force in the local element coordinates, i.e., along the axis of the element. In the default case, i.e., no orientation vector is used; the global components of the discrete element force are obtained by using the element's direction cosines:

$$\begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix} = \frac{\hat{f}}{l} \begin{Bmatrix} \Delta l_x \\ \Delta l_y \\ \Delta l_z \end{Bmatrix} = \hat{f} \begin{Bmatrix} n_x \\ n_y \\ n_z \end{Bmatrix} = \hat{f}\, \underset{\sim}{n} \tag{16.2}$$

where

$$\underset{\sim}{\Delta l} = \begin{Bmatrix} \Delta l_x \\ \Delta l_y \\ \Delta l_z \end{Bmatrix} = \begin{Bmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{Bmatrix} \tag{16.3}$$

l is the length

$$l = \sqrt{\Delta l_x^2 + \Delta l_y^2 + \Delta l_z^2} \tag{16.4}$$

and (x_i, y_i, z_i) are the global coordinates of the nodes of the spring element. The forces in Equation (16.2) are added to the first node and subtracted from the second node.

For a node tied to ground we use the same approach but for the (x_2, y_2, z_2) coordinates in Equation (16.2) the initial coordinates of node 1, i.e., (x_0, y_0, z_0) are used instead; therefore,

$$\begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix} = \frac{\hat{f}}{l} \begin{Bmatrix} x_0 - x_1 \\ y_0 - y_1 \\ z_0 - z_1 \end{Bmatrix} = \hat{f} \begin{Bmatrix} n_x \\ n_y \\ n_z \end{Bmatrix} \tag{16.5}$$

The increment in the element force is determined from the user specified force-displacement relation. Currently, nine types of force-displacement/velocity relationships may be specified:

1. linear elastic;
2. linear viscous;
3. nonlinear elastic;
4. nonlinear viscous;
5. elasto-plastic with isotropic hardening;
6. general nonlinear;
7. linear viscoelastic.
8. inelastic tension and compression only.
9. muscle model.

The force-displacement relations for these models are discussed in the following later.

16.1 Orientation Vectors

An orientation vector,

$$\underset{\sim}{m} = \begin{Bmatrix} m_1 \\ m_2 \\ m_3 \end{Bmatrix} \tag{16.6}$$

can be defined to control the direction the spring acts. If orientation vectors are used, it is strongly recommended that the nodes of the discrete element be coincident and remain approximately so throughout the calculation. If the spring or damper is of finite length, rotational constraints will appear in the model that can substantially affect the results. If finite length springs are needed with directional vectors, then the discrete beam elements, the type 6 beam, should be used with the coordinate system flagged for the finite length case.

We will first consider the portion of the displacement that lies in the direction of the vector. The displacement of the spring is updated based on the change of length given by

$$\Delta I = I - I_0 \tag{16.7}$$

where I_0 is the initial length in the direction of the vector and l is the current length given for a node to node spring by

$$I = m_1 (x_2 - x_1) + m_2 (y_2 - y_1) + m_3 (z_2 - z_1) \tag{16.8}$$

and for a node to ground spring by

$$I = m_1 (x_0 - x_1) + m_2 (y_0 - y_1) + m_3 (z_0 - z_1) \tag{16.9}$$

The latter case is not intuitively obvious and can affect the sign of the force in unexpected ways if the user is not familiar with the relevant equations. The nodal forces are then given by

$$\begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix} = \hat{f} \begin{Bmatrix} m_1 \\ m_2 \\ m_3 \end{Bmatrix} \tag{16.10}$$

The orientation vector can be either permanently fixed in space as defined in the input or acting in a direction determined by two moving nodes which must not be coincident but may be independent of the nodes of the spring. In the latter case, we recompute the direction every cycle according to:

$$\begin{Bmatrix} m_1 \\ m_2 \\ m_3 \end{Bmatrix} = \frac{1}{l^n} \begin{Bmatrix} x_2^n - x_1^n \\ y_2^n - y_1^n \\ z_2^n - z_1^n \end{Bmatrix} \tag{16.11}$$

In Equation (16.9) the superscript, n, refers to the orientation nodes.

For the case where we consider motion in the plane perpendicular to the orientation vector we consider only the displacements in the plane, $\underset{\sim}{\Delta l}^p$, given by,

$$\underset{\sim}{\Delta l}^p = \underset{\sim}{\Delta l} - \underset{\sim}{m}\left(\underset{\sim}{m} \cdot \underset{\sim}{\Delta l}\right) \tag{16.12}$$

We update the displacement of the spring based on the change of length in the plane given by

$$\Delta l^p = l^p - l_0^p \tag{16.13}$$

where l_0^p is the initial length in the direction of the vector and l is the current length given for a node to node spring by

$$l^p = m_1^p (x_2 - x_1) + m_2^p (y_2 - y_1) + m_3^p (z_2 - z_1) \tag{16.14}$$

and for a node to ground spring by

$$l^p = m_1^p (x_0 - x_1) + m_2^p (y_0 - y_1) + m_3^p (z_0 - z_1) \tag{16.15}$$

where

$$\begin{Bmatrix} m_1^p \\ m_2^p \\ m_3^p \end{Bmatrix} = \frac{1}{\sqrt{\Delta l_x^{p\,2} + \Delta l_y^{p\,2} + \Delta l_z^{p\,2}}} \begin{Bmatrix} \Delta l_x^p \\ \Delta l_y^p \\ \Delta l_z^p \end{Bmatrix} \tag{16.16}$$

After computing the displacements, the nodal forces are then given by

$$\begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix} = \hat{f} \begin{Bmatrix} m_1^p \\ m_2^p \\ m_3^p \end{Bmatrix} \tag{16.17}$$

16.2 Dynamic Magnification "Strain Rate" Effects

To account for "strain rate" effects, we have a simple method of scaling the forces based to the relative velocities that applies to all springs. The forces computed from the spring elements are assumed to be the static values and are scaled by an amplification factor to obtain the dynamic value:

$$F_{dynamic} = \left(1. + k_d \frac{V}{V_0}\right) F_{static} \tag{16.18}$$

where

k_d	=	is a user defined input value
V	=	absolute relative velocity
V_0	=	dynamic test velocity

For example, if it is known that a component shows a dynamic crush force at 15m/s equal to 2.5 times the static crush force, use $k_d = 1.5$ and $V_0 = 15$.

16.3 Deflection Limits in Tension and Compression

The deflection limit in compression and tension is restricted in its application to no more than one spring per node subject to this limit, and to deformable bodies only. For example in the former case, if three spring are in series either the center spring or the two end springs may be subject to a limit but not all three. When the limiting deflection is reached momentum conservation calculations are performed and a common acceleration is computed:

$$\hat{a}_{common} = \frac{\hat{f}_1 + \hat{f}_2}{m_1 + m_2} \tag{16.19}$$

An error termination will occur if a rigid body node is used in a spring definition where compression is limited.

16.4 Linear Elastic or Linear Viscous

These discrete elements have the simplest force-displacement relations. The linear elastic discrete element has a force-displacement relation of the form

$$\hat{f} = K\Delta l \tag{16.20}$$

where K is the element's stiffness and Δl is the change in length of the element. The linear viscous element has a similar force-velocity (rate of displacement) relation:

$$\hat{f} = C\frac{\Delta l}{\Delta t} \tag{16.21}$$

where C is a viscous damping parameter and Δt is the time step increment.

16.5 Nonlinear Elastic or Nonlinear Viscous

These discrete elements use piecewise force-displacement or force-relative velocity relations. The nonlinear elastic discrete element has a tabulated force-displacement relation of the form

$$\hat{f} = K\Delta l \tag{16.22}$$

where $K(\Delta l)$ is the tabulated force that depends on the total change in the length of the element (Figure 16.1) The nonlinear viscous element has a similar tabulated force-relative velocity relation:

$$\hat{f} = C\frac{\Delta l}{\Delta t} \tag{16.23}$$

where $C\left(\frac{\Delta l}{\Delta t}\right)$ is the viscous damping force that depends on the rate of change of the element's length. Nonlinear discrete elements unload along the loading paths.

If the spring element is initially of zero length and if no orientation vectors are used then only the tensile part of the stress strain curve needs to be defined. However, if the spring element is initially of finite length then the curve must be defined in both the positive and negative quadrants.

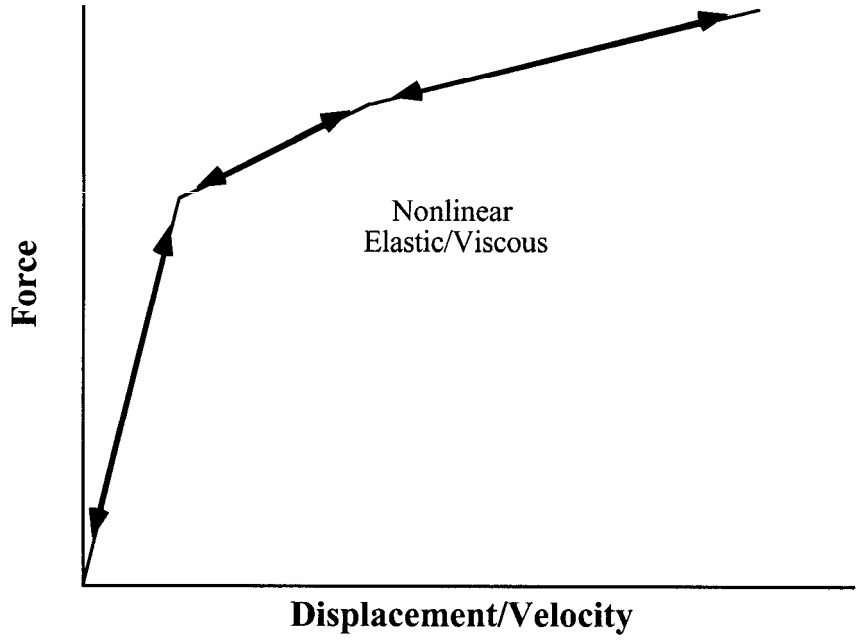


Figure 16.1. Piecewise linear force-displacement curve for nonlinear elastic discrete element.

16.6 Elasto-Plastic with Isotropic Hardening

The elasto-plastic discrete element has a bilinear force-displacement relationship that is specified by the elastic stiffness, a tangent stiffness and a yield force (Figure 16.2). This discrete element uses the elastic stiffness model for unloading until the yield force is exceeded during unloading. The yield force is updated to track its maximum value which is equivalent to an isotropic hardening model. The force-displacement relation during loading may be written as

$$\hat{f} = F_y\left(1-\frac{K_t}{K}\right)+K_t\Delta l \tag{16.24}$$

where F_y is the yield force and K_t is the tangent stiffness.

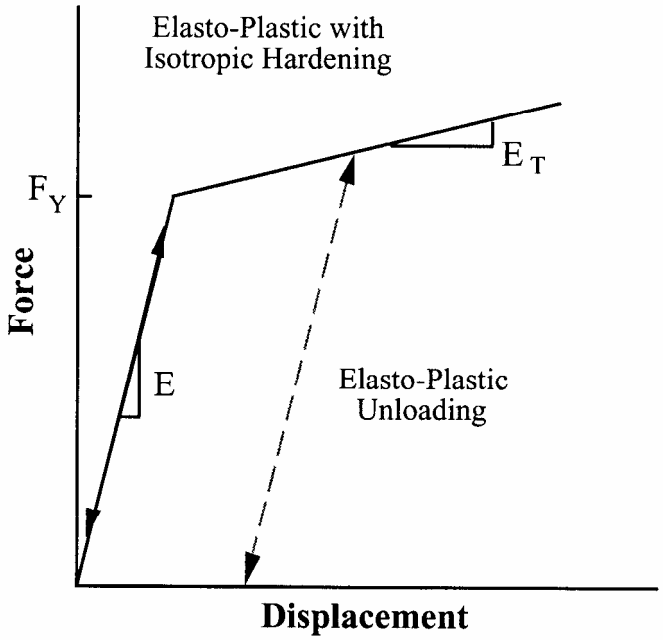


Figure 16.2. Loading and unloading force-displacement curves for elasto-plastic discrete element.

16.7 General Nonlinear

The general nonlinear discrete element allows the user to specify independent and nonsymmetrical piecewise linear loading and unloading paths (Figure 16.3(a)).

This element combines the features of the above-described nonlinear elastic and elasto-plastic discrete elements by allowing the user to specify independent initial yield forces in tension (F_{YT}) and in compression (F_{YC}). If the discrete element force remains between these initial yield values, the element unloads along the loading path (Figure 16.3(b)). This corresponds to the nonlinear elastic discrete element.

However, if the discrete element force exceeds either of these initial yield values, the specified unloading curve is used for subsequent unloading. Additionally, the initial loading and unloading curves are allowed to move in the force-displacement space by specifying a mixed hardening parameter β, where $\beta = 0$ corresponds to kinematic hardening (Figure 16.3(c)) and $\beta = 0$ $\beta = 1$ corresponds to isotropic hardening (Figure 16.3(d)).

16.8 Linear Visco-Elastic

The linear viscoelastic discrete element [Schwer, Cheva, and Hallquist 1991] allows the user to model discrete components that are characterized by force relaxation or displacement creep behavior. The element's variable stiffness is defined by three parameters and has the form

$$K(t) = K_\infty + (K_0 - K_\infty)e^{-\beta t} \qquad (16.25)$$

where K_∞ is the long duration stiffness, K_0 is the short time stiffness, and β is a decay parameter that controls the rate at which the stiffness transitions between the short and long duration stiffness (Figure 16.4).

This model corresponds to a three-parameter Maxwell model (see insert in Figure 16.4) which consists of a spring and damper in series connected to another spring in parallel. Although this discrete element behavior could be built up using the above- described linear elastic and linear viscous discrete elements, such a model would also require the user to specify the nodal mass at the connection of the series spring and damper. This mass introduces a fourth parameter which would further complicate fitting the model to experimental data.

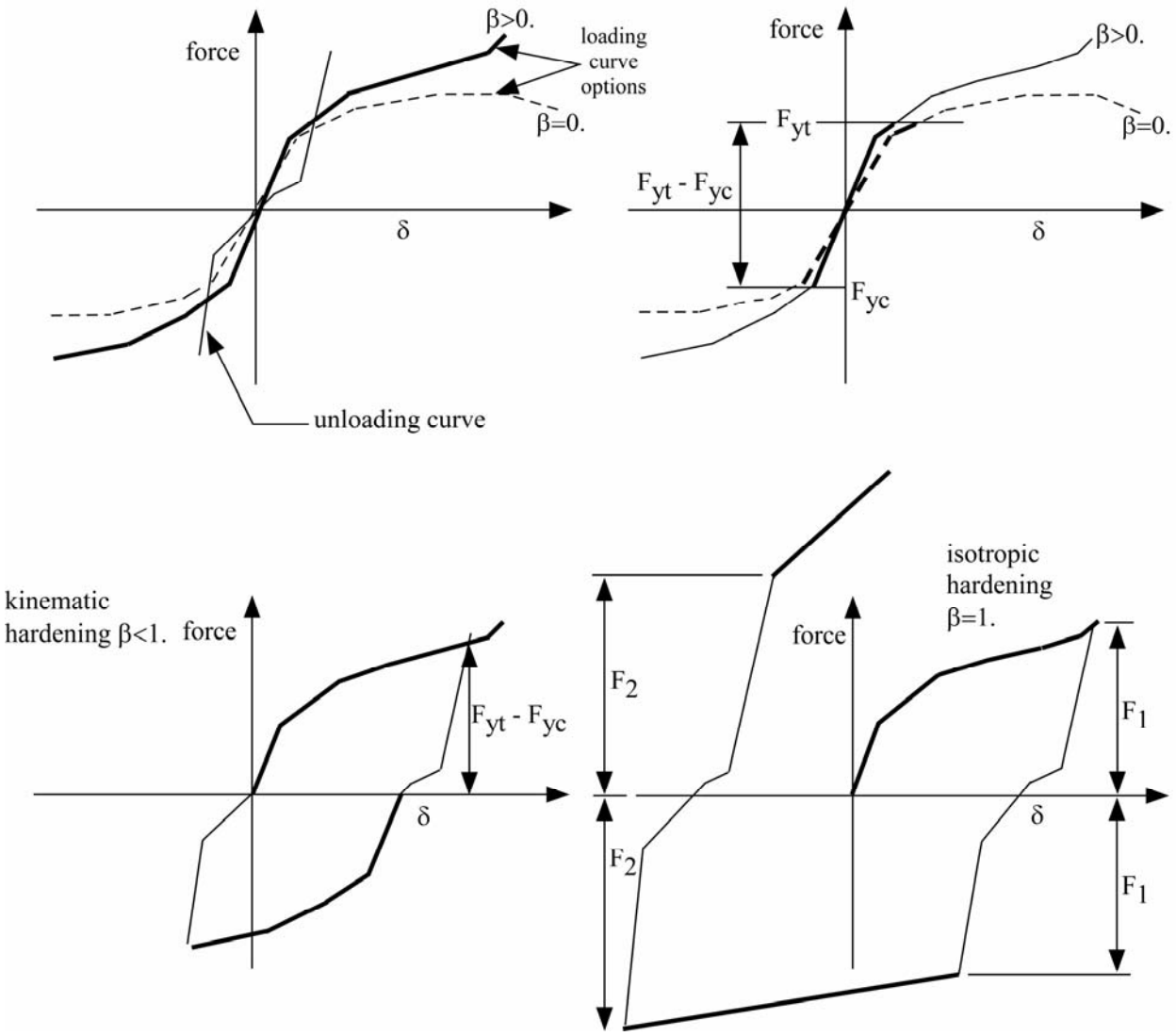


Figure 16.3. Loading and unloading force displacement curves for general nonlinear discrete element.

16.9 Muscle Model

This is Material Type 15 for discrete springs and dampers. This material is a Hill-type muscle model with activation. It is for use with discrete elements. The LS-DYNA implementation is due to Dr. J.A. Weiss.

L0	Initial muscle length, *Lo*.
VMAX	Maximum CE shortening velocity, *Vmax*.
SV	Scale factor, *Sv*, for *Vmax* vs. active state. LT.0: absolute value gives load curve ID GE.0: constant value of 1.0 is used
A	Activation level vs. time function. LT.0: absolute value gives load curve ID GE.0: constant value of *A* is used
FMAX	Peak isometric force, *Fmax*.
TL	Active tension vs. length function. LT.0: absolute value gives load curve ID GE.0: constant value of 1.0 is used
TV	Active tension vs. velocity function. LT.0: absolute value gives load curve ID GE.0: constant value of 1.0 is used
FPE	Force vs. length function, *Fpe*, for parallel elastic element. LT.0: absolute value gives load curve ID EQ.0: exponential function is used (see below) GT.0: constant value of 0.0 is used
LMAX	Relative length when *Fpe* reaches *Fmax*. Required if *Fpe*=0 above.
KSH	Constant, *Ksh*, governing the exponential rise of *Fpe*. Required if *Fpe*=0 above.

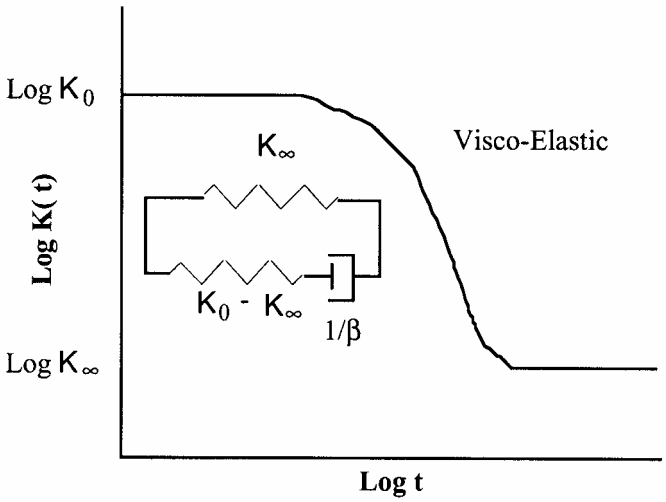


Figure 16.4. Typical stiffness relaxation curve used for the viscoelastic discrete element.

The material behavior of the muscle model is adapted from the original model proposed by Hill (1938). Reviews of this model and extensions can be found in Winters (1990) and Zajac (1989). The most basic Hill-type muscle model consists of a contractile element (CE) and a parallel elastic element (PE) (Figure 16.5). An additional series elastic element (SEE) can be added to represent tendon compliance. The main assumptions of the Hill model are that the contractile element is entirely stress free and freely distensible in the resting state, and is described exactly by Hill's equation (or some variation). When the muscle is activated, the series and parallel elements are elastic, and the whole muscle is a simple combination of identical sarcomeres in series and parallel. The main criticism of Hill's model is that the division of forces between the parallel elements and the division of extensions between the series elements is arbitrary, and cannot be made without introducing auxiliary hypotheses. However, these criticisms apply to *any* discrete element model. Despite these limitations, the Hill model has become extremely useful for modeling musculoskeletal dynamics, as illustrated by its widespread use today.

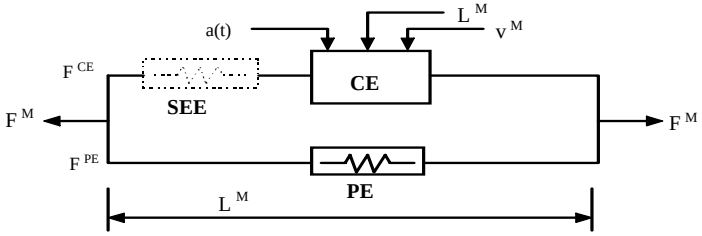


Figure 16.5. Discrete model for muscle contraction dynamics, based on a Hill-type representation. The total force is the sum of passive force F^{PE} and active force F^{CE}. The passive element (PE) represents energy storage from muscle elasticity, while the contractile element (CE) represents force generation by the muscle. The series elastic element (SEE), shown in dashed lines, is often neglected when a series tendon compliance is included. Here, a(t) is the activation level, L^M is the length of the muscle, and v^M is the shortening velocity of the muscle.

When the contractile element (CE) of the Hill model is inactive, the entire resistance to elongation is provided by the PE element and the tendon load-elongation behavior. As activation is increased, force then passes through the CE side of the parallel Hill model, providing the contractile dynamics. The original Hill model accommodated only full activation - this limitation is circumvented in the present implementation by using the modification suggested by Winters (1990). The main features of his approach were to realize that the CE force-velocity input force equals the CE tension-length output force. This yields a three-dimensional curve to describe the force-velocity-length relationship of the CE. If the force-velocity y-intercept scales with activation, then given the activation, length and velocity, the CE force can be determined.

Without the SEE, the total force in the muscle FM is the sum of the force in the CE and the PE because they are in parallel:

$$F^M = F^{PE} + F^{CE}$$

The relationships defining the force generated by the CE and PE as a function of L^M, V^M and a(t) are often scaled by F_{max}, the peak isometric force (p. 80, Winters 1990), L_0, the initial length of the muscle (p. 81, Winters 1990), and V_{max}, the maximum unloaded CE shortening velocity (p. 80, Winters 1990). From these, dimensionless length and velocity can be defined:

$$L = \frac{L^M}{L_o},$$

$$V = \frac{V^M}{V_{max} * S_V(a(t))}$$

Here, S_V scales the maximum CE shortening velocity V_{max} and changes with activation level a(t). This has been suggested by several researchers, i.e. Winters and Stark [1985]. The activation level specifies the level of muscle stimulation as a function of time. Both have values between 0 and 1. The functions $S_V(a(t))$ and a(t) are specified via load curves in LS-DYNA, or default values of S_V=1 and a(t)=0 are used. Note that L is always positive and that V is positive for lengthening and negative for shortening.

The relationship between F^{CE}, V and L was proposed by Bahler et al. [1967]. A three-dimensional relationship between these quantities is now considered standard for computer implementations of Hill-type muscle models [i.e., eqn 5.16, p. 81, Winters 1990]. It can be written in dimensionless form as:

$$F^{CE} = a(t) * F_{max} * f_{TL}(L) * f_{TV}(V)$$

Here, f_{TL} and f_{TV} are the tension-length and tension-velocity functions for active skeletal muscle. Thus, if current values of L^M, V^M, and a(t) are known, then F^{CE} can be determined (Figure 16.5).

The force in the parallel elastic element F^{PE} is determined directly from the current length of the muscle using an exponential relationship [eqn 5.5, p. 73, Winters 1990]:

$$f_{PE} = \frac{F^{PE}}{F_{MAX}} = 0, \quad L \le 1$$

$$f_{PE} = \frac{F^{PE}}{F_{MAX}} = \frac{1}{\exp(K_{sh}) - 1}\left[\exp\left(\frac{K_{sh}}{L_{max}}(L-1)\right) - 1\right], \quad L > 1$$

Here, L_{max} is the relative length at which the force F_{max} occurs, and K_{sh} is a dimensionless shape parameter controlling the rate of rise of the exponential. Alternatively, the user can define a custom f_{PE} curve giving tabular values of normalized force versus dimensionless length as a load curve.

For computation of the total force developed in the muscle F^M, the functions for the tension-length f_{TL} and force-velocity f_{TV} relationships used in the Hill element must be defined. These relationships have been available for over 50 years, but have been refined to allow for behavior such as active lengthening. The active tension-length curve f_{TL} describes the fact that isometric muscle force development is a function of length, with the maximum force occurring at an optimal length. According to Winters, this optimal length is typically around L=1.05, and the force drops off for shorter or longer lengths, approaching zero force for L=0.4 and L=1.5. Thus the curve has a bell-shape. Because of the variability in this curve between muscles, the user must specify the function f_{TL} via a load curve, specifying pairs of points representing the normalized force (with values between 0 and 1) and normalized length L (Figure 16.6).

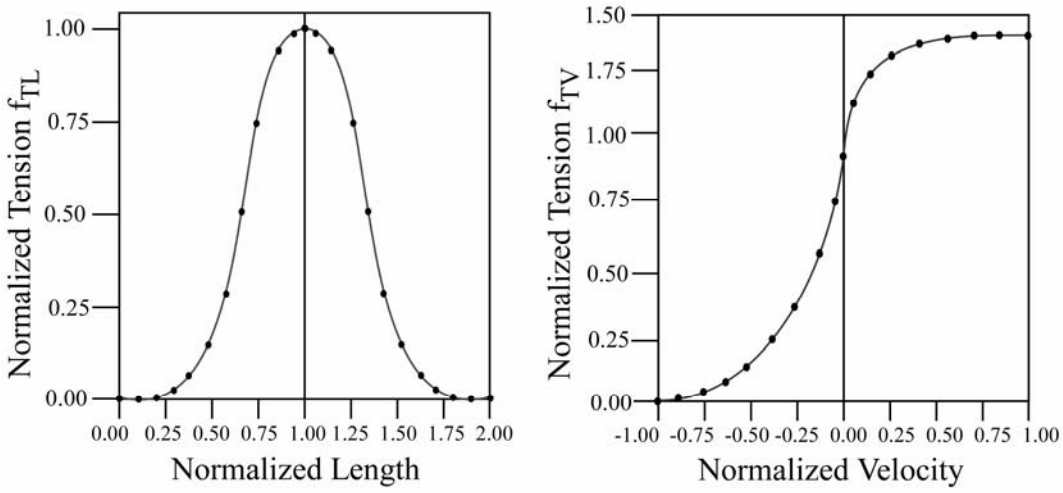


Figure 16.6. Typical normalized tension-length (TL) and tension-velocity (TV) curves for skeletal muscle.

The active tension-velocity relationship f_{TV} used in the muscle model is mainly due to the original work of Hill. Note that the dimensionless velocity V is used. When V=0, the normalized tension is typically chosen to have a value of 1.0. When V is greater than or equal to 0, muscle lengthening occurs. As V increases, the function is typically designed so that the force increases from a value of 1.0 and asymptotes towards a value near 1.4. When V is less than zero, muscle shortening occurs and the classic Hill equation hyperbola is used to drop the normalized tension to 0 (Figure 16.6). The user must specify the function f_{TV} via a load curve, specifying

pairs of points representing the normalized tension (with values between 0 and 1) and normalized velocity V.

16.10 Seat Belt Material

The seat belt capability reported here was developed by Walker and co-workers [Walker and Dallard 1991, Strut, Walker, et al., 1991] and this section excerpted from their documentation. Each belt material defines stretch characteristics and mass properties for a set of belt elements. The user enters a load curve for loading, the points of which are *(Strain, Force)*. Strain is defined as engineering strain, i.e.

$$Strain = \frac{current\ \ length}{initial\ length} - 1.$$

Another similar curve is entered to describe the unloading behavior. Both loadcurves should start at the origin (0,0) and contain positive force and strain values only. The belt material is tension only with zero forces being generated whenever the strain becomes negative. The first non-zero point on the loading curve defines the initial yield point of the material. On unloading, the unloading curve is shifted along the strain axis until it crosses the loading curve at the 'yield' point from which unloading commences. If the initial yield has not yet been exceeded or if the origin of the (shifted) unloading curve is at negative strain, the original loading curves will be used for both loading and unloading. If the strain is less than the strain at the origin of the unloading curve, the belt is slack and no force is generated. Otherwise, forces will then be determined by the unloading curve for unloading and reloading until the strain again exceeds yield after which the loading curves will again be used.

A small amount of damping is automatically included. This reduces high frequency oscillation, but, with realistic force-strain input characteristics and loading rates, does not significantly alter the overall forces-strain performance. The damping forced opposes the relative motion of the nodes and is limited by stability:

$$D = \frac{.1 \times mass \times relative\ velocity}{timestep\ size}$$

In addition, the magnitude of the damping forces is limited to one tenth of the force calculated from the forces-strain relationship and is zero when the belt is slack. Damping forces are not applied to elements attached to sliprings and retractors.

The user inputs a mass per unit length that is used to calculate nodal masses on initialization.

A 'minimum length' is also input. This controls the shortest length allowed in any element and determines when an element passes through sliprings or are absorbed into the retractors. One tenth of a typical initial element length is usually a good choice.

16.11 Seat Belt Elements

Belt elements are single degree of freedom elements connecting two nodes and are treated in a manner similar to the spring elements. When the strain in an element is positive (i.e., the current length is greater then the unstretched length), a tension force is calculated from the material characteristics and is applied along the current axis of the element to oppose further stretching. The unstretched length of the belt is taken as the initial distance between the two nodes defining the position of the element plus the initial slack length. At the beginning of the calculation the seatbelt elements can be obtained within a retractor.

16.12 Sliprings

Sliprings are defined in the LS-DYNA input by giving a slipring ID and element ID's for two elements who share a node which is coincident with the slipring node. The slipring node may not be attached to any belt elements.

Sliprings allow continuous sliding of a belt through a sharp change of angle. Two elements (1 and 2 in Figure 16.5) meet at the slipring. Node B in the belt material remains attached to the slipring node, but belt material (in the form of unstretched length) is passed from element 1 to element 2 to achieve slip. The amount of slip at each timestep is calculated from the ratio of forces in elements 1 and 2. The ratio of forces is determined by the relative angle between elements 1 and 2 and the coefficient of friction, μ. The tension in the belts is taken as T_1 and T_2, where T_2 is on the high-tension side and T_1 is the force on the low-tension side. Thus if T_2 is sufficiently close to T_1 no slip occurs; otherwise, slip is just sufficient to reduce the ratio T_2/T_1 to $e^{\mu\Theta}$. No slip occurs if both elements are slack. The out-of-balance force at node B is reacted on the slipring node; the motion of node B follows that of slipring node.

If, due to slip through the slipring, the unstretched length of an element becomes less than the minimum length (as entered on the belt material card), the belt is remeshed locally: the short element passes through the slipring and reappears on the other side (see Figure 16.5). The new unstretched length of e1 is 1.1 $\times$ minimum length. Force and strain in e2 and e3 are unchanged; force and strain in e1 are now equal to those in e2. Subsequent slip will pass material from e3 to e1. This process can continue with several elements passing in turn through the slipring.

To define a slipring, the user identifies the two belt elements which meet at the slipring, the friction coefficient, and the slipring node. The two elements must have a common node coincident with the slipring node. No attempt should be made to restrain or constrain the common node for its motion will automatically be constrained to follow the slipring node. Typically, the slipring node is part of the vehicle body structure and, therefore, belt elements should not be connected to this node directly, but any other feature can be attached, including rigid bodies.

16.13 Retractors

Retractors are defined by giving a node, the "retractor node" and an element ID of an element outside the retractor but with one node that is coincident with the retractor node. Also sensor ID's must be defined for up to four sensors which can activate the seatbelt.

Retractors allow belt material to be paid out into a belt element, and they operate in one of two regimes: unlocked when the belt material is paid out or reeled in under constant tension and locked when a user defined force-pullout relationship applies.

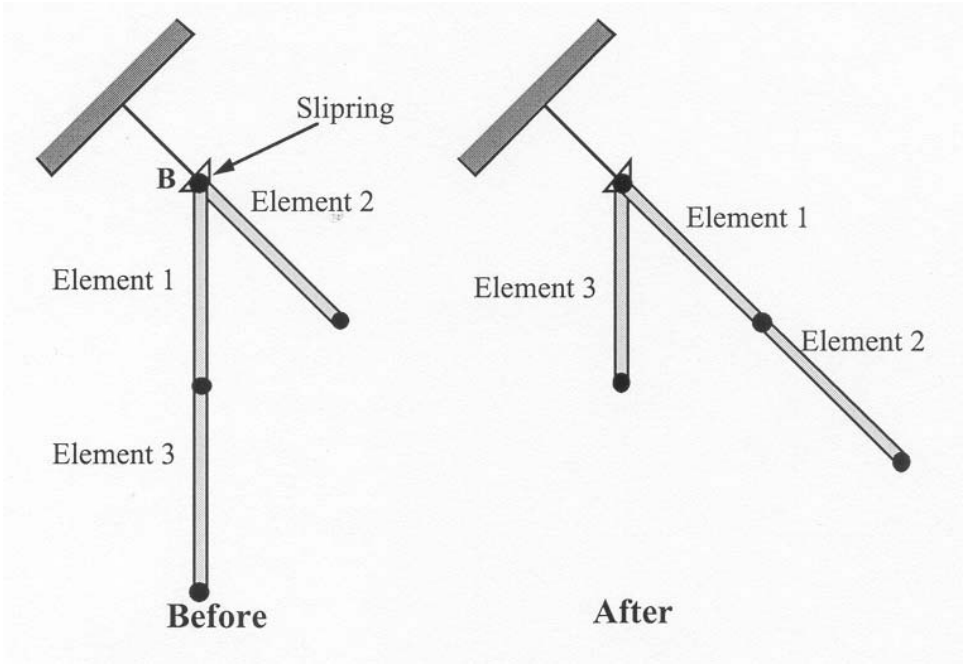


Figure 16.5. Elements passing through slipring.

The retractor is initially unlocked, and the following sequence of events must occur for it to become locked:

1. Any one of up to four sensors must be triggered. (The sensors are described below).
2. Then a user-defined time delay occurs.
3. Then a user-defined length of belt must be payed out (optional).
4. Then the retractor locks.

and once locked, it remains locked.

In the unlocked regime, the retractor attempts to apply a constant tension to the belt. This feature allows an initial tightening of the belt, and takes up any slack whenever it occurs. The tension value is taken from the first point on the force-pullout load curve. The maximum rate of pull out or pull in is given by $0.01 \times$ fed length per time step. Because of this, the constant tension value is not always achieved.

In the locked regime, a user-defined curve describes the relationship between the force in the attached element and the amount of belt material paid out. If the tension in the belt subsequently relaxes, a different user-defined curve applies for unloading. The unloading curve is followed until the minimum tension is reached.

The curves are defined in terms of initial length of belt. For example, if a belt is marked at 10mm intervals and then wound onto a retractor, and the force required to make each mark emerge from the (locked) retractor is recorded, the curves used for input would be as follows:

0	Minimum tension (should be > zero)
10mm	Force to emergence of first mark
20mm	Force to emergence of second mark
.	.
.	.
.	.

Pyrotechnic pretensions may be defined which cause the retractor to pull in the belt at a predetermined rate. This overrides the retractor force-pullout relationship from the moment when the pretensioner activates.

If desired, belt elements may be defined which are initially inside the retractor. These will emerge as belt material is paid out, and may return into the retractor if sufficient material is reeled in during unloading.

Elements e2, e3 and e4 are initially inside the retractor, which is paying out material into element e1. When the retractor has fed L_{crit} into e1, where:

L_{crit} = fed length - 1.1 × minimum length
(minimum length defined on belt material input)
(fed length defined on retractor input)

element e2 emerges with an unstretched length of 1.1 × minimum length; the unstretched length of element e1 is reduced by the same amount. The force and strain in e1 are unchanged; in e2, they are set equal to those in e1. The retractor now pays out material into e2.

If no elements are inside the retractor, e2 can continue to extend as more material is fed into it.

As the retractor pulls in the belt (for example, during initial tightening), if the unstretched length of the mouth element becomes less than the minimum length, the element is taken into the retractor.

To define a retractor, the user enters the retractor node, the ‘mouth’ element (into which belt material will be fed, e1 in Figure 16.6, up to 4 sensors which can trigger unlocking, a time delay, a payout delay (optional), load and unload curve numbers, and the fed length. The retractor node is typically part of the vehicle stricture; belt elements should not be connected to this node directly, but any other feature can be attached including rigid bodies. The mouth element should have a node coincident with the retractor but should not be inside the retractor. The fed length would typically be set either to a typical element initial length, for the distance between painted marks on a real belt for comparisons with high-speed film. The fed length should be at least three times the minimum length.

If there are elements initially inside the retractor (e2, e3 and e4 in the Figure) they should not be referred to on the retractor input, but the retractor should be identified on the element input for these elements. Their nodes should all be coincident with the retractor node and should not be restrained or constrained. Initial slack will automatically be set to 1.1 × minimum length for these elements; this overrides any user-defined value.

Weblockers can be included within the retractor representation simply by entering a ‘locking up’ characteristic in the force pullout curve, see Figure 16.7. The final section can be very steep (but must have a finite slope).

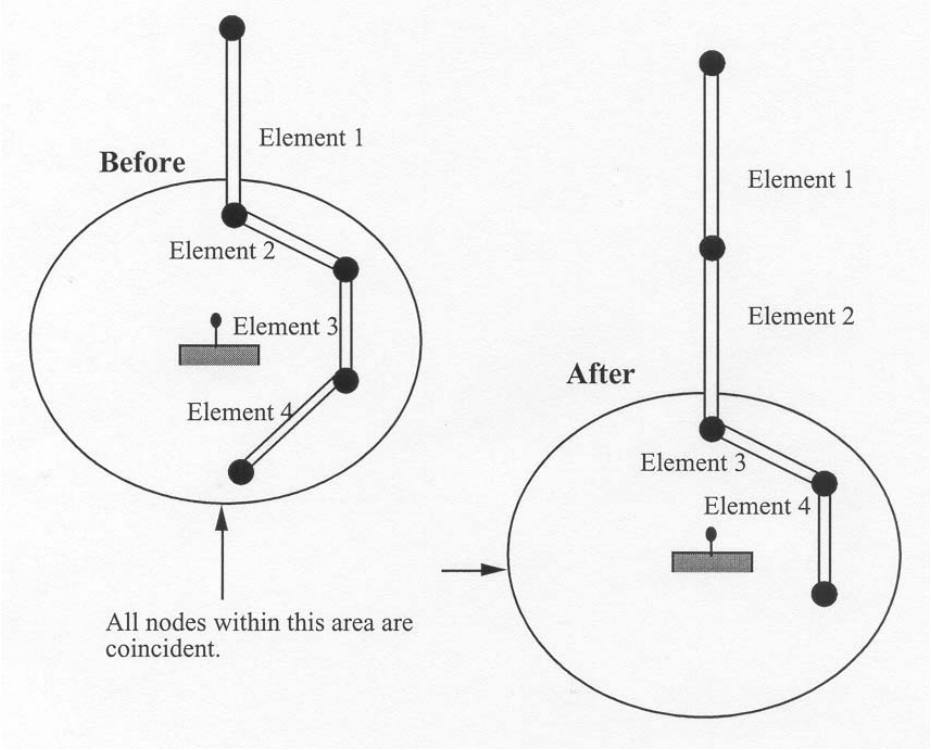


Figure 16.6. Elements in a retractor.

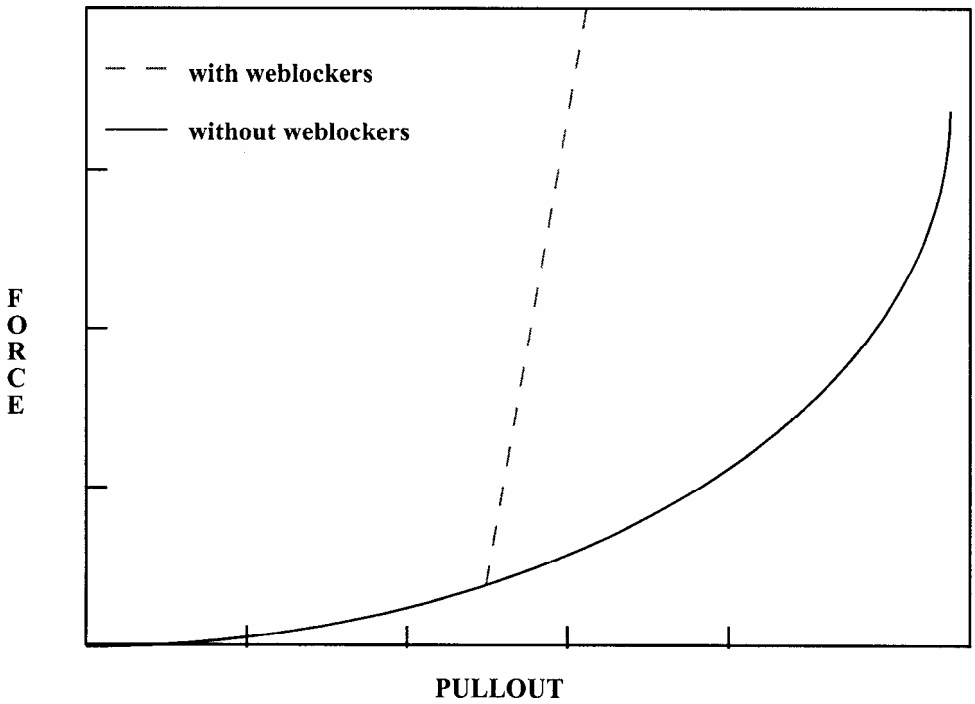


Figure 16.7. Retractor force pull characteristics.

16.14 Sensors

Sensors are used to trigger locking of retractors and activate pretensioners. Four types of sensor are available which trigger according to the following criteria:

Type 1 – When the magnitude of x-, y-, or z- acceleration of a given node has remained above a given level continuously for a given time, the sensor triggers. This does not work with nodes on rigid bodies.

Type 2 – When the rate of belt payout from a given retractor has remained above a given level continuously for a given time, the sensor triggers.

Type 3 – The sensor triggers at a given time.

Type 4 – The sensor triggers when the distance between two nodes exceeds a given maximum or becomes less than a given minimum. This type of sensor is intended for use with an explicit mas/spring representation of the sensor mechanism.

By default, the sensors are inactive during dynamic relaxation. This allows initial tightening of the belt and positioning of the occupant on the seat without locking the retractor or firing any pretensioners. However, a flag can be set in the sensor input to make the sensors active during the dynamic relaxation phase.

16.15 Pretensioners

Pretensioners allow modeling of three types of active devices which tighten the belt during the initial stages of a crash. The first type represents a pyrotechnic device which spins the spool of a retractor, causing the belt to be reeled in. The user defines a pull-in versus time curve which applies once the pretensioner activates. The remaining types represents preloaded springs or torsion bars which move the buckle when released. The pretensioner is associated with any type of spring element including rotational. Note that the preloaded spring, locking spring and any restraints on the motion of the associated nodes are defined in the normal way; the action of the pretensioner is merely to cancel the force in one spring until (or after) it fires. With the second type, the force in the spring element is cancelled out until the pretensioner is activated. In this case the spring in question is normally a stiff, linear spring which acts as a locking mechanism, preventing motion of the seat belt buckle relative to the vehicle. A preloaded spring is defined in parallel with the locking spring. This type avoids the problem of the buckle being free to ‘drift’ before the pretensioner is activated.

To activate the pretensioner the following sequence of events must occur:

1. Any one of up to four sensors must be triggered.
2. Then a user-defined time delay occurs.
3. Then the pretensioner acts.

16.16 Accelerometers

The accelerometer is defined by three nodes in a rigid body which defines a triad to measure the accelerations in a local system. The presence of the accelerometer means that the accelerations and velocities of node 1 will be output to **all** output files in local instead of global coordinates.

The local coordinate system is defined by the three nodes as follows:

- local **x** from node 1 to node 2
- local **z** perpendicular to the plane containing nodes, 1, 2, and 3 ($z = \mathbf{x} \times \mathbf{a}$), where **a** is from node 1 to node 3).
- local $\mathbf{y} = \mathbf{x} \times \mathbf{z}$

The three nodes should all be part of the same rigid body. The local axis then rotates with the body.

17. SIMPLIFIED ARBITRARY LAGRANGIAN-EULERIAN

Arbitrary Lagrangian-Eulerian (ALE) formulations may be thought of as algorithms that perform automatic rezoning. Users perform manual rezoning by

1. Stopping the calculation when the mesh is distorted,
2. Smoothing the mesh,
3. Remapping the solution from the distorted mesh to the smooth mesh.

An ALE formulation consists of a Lagrangian time step followed by a "remap" or "advection" step. The advection step performs an incremental rezone, where "incremental" refers to the fact that the positions of the nodes are moved only a small fraction of the characteristic lengths of the surrounding elements. Unlike a manual rezone, the topology of the mesh is fixed in an ALE calculation. An ALE calculation can be interrupted like an ordinary Lagrangian calculation and a manual rezone can be performed if an entirely new mesh is necessary to continue the calculation.

The accuracy of an ALE calculation is often superior to the accuracy of a manually rezoned calculation because the algorithm used to remap the solution from the distorted to the undistorted mesh is second order accurate for the ALE formulation while the algorithm for the manual rezone is only first order accurate.

In theory, an ALE formulation contains the Eulerian formulation as a subset. Eulerian codes can have more than one material in each element, but most ALE implementations are simplified ALE formulations which permit only a single material in each element. The primary advantage of a simplified formulation is its reduced cost per time step. When elements with more than one material are permitted, the number and types of materials present in an element can change dynamically. Additional data is necessary to specify the materials in each element and the data must be updated by the remap algorithms.

The range of problems that can be solved with an ALE formulation is a direct function of the sophistication of the algorithms for smoothing the mesh. Early ALE codes were not very successful largely because of their primitive algorithms for smoothing the mesh. In simplified ALE formulations, most of the difficulties with the mesh are associated with the nodes on the material boundaries. If the material boundaries are purely Lagrangian, i.e., the boundary nodes move with the material at all times, no smooth mesh maybe possible and the calculation will terminate. The algorithms for maintaining a smooth boundary mesh are therefore as important to the robustness of the calculations as the algorithms for the mesh interior.

The cost of the advection step per element is usually much larger than the cost of the Lagrangian step. Most of the time in the advection step is spent in calculating the material transported between the adjacent elements, and only a small part of it is spent on calculating how and where the mesh should be adjusted. Second order accurate monotonic advection algorithms are used in LS-DYNA despite their high cost per element because their superior coarse mesh accuracy which allows the calculation to be performed with far fewer elements than would be possible with a cheaper first order accurate algorithm.

The second order transport accuracy is important since errors in the transport calculations generally smooth out the solution and reduce the peak values in the history variables. Monotonic advection algorithms are constructed to prevent the transport calculations from creating new minimum or maximum values for the solution variables. They were first developed for the solution of the Navier Stokes equations to eliminate the spurious oscillations that appeared

around the shock fronts. Although monotonic algorithms are more diffusive than algorithms that are not monotonic, they must be used for stability in general purpose codes. Many constitutive models have history variables that have limited ranges, and if their values are allowed to fall outside of their allowable ranges, the constitutive models are undefined. Examples include explosive models, which require the burn fraction to be between zero and one, and many elastoplasticity models, such as those with power law hardening, which require a non-negative plastic strain.

The overall flow of an ALE time step is:

1. Perform a Lagrangian time step.
2. Perform an advection step.
   a. Decide which nodes to move.
   b. Move the boundary nodes.
   c. Move the interior nodes.
   d. Calculate the transport of the element-centered variables.
   e. Calculate the momentum transport and update the velocity.

Each element solution variable must be transported. The total number of solution variables, including the velocity, is at least six and depends on the material models. For elements that are modeled with an equation of state, only the density, the internal energy, and the shock viscosity are transported. When the elements have strength, the six components of the stress tensor and the plastic strain must also be advected, for a total of ten solution variables. Kinematic hardening, if it is used, introduces another five solution variables, for a total of fifteen.

The nodal velocities add an extra three solution variables that must be transported, and they must be advected separately from the other solution variables because they are centered at the nodes and not in the elements. In addition, the momentum must be conserved, and it is a product of the node-centered velocity and the element-centered density. This imposes a constraint on how the momentum transport is performed that is unique to the velocity field. A detailed consideration of the difficulties associated with the transport of momentum is deferred until later.

Perhaps the simplest strategy for minimizing the cost of the ALE calculations is to perform them only every few time steps. The cost of an advection step is typically two to five times the cost of the Lagrangian time step. By performing the advection step only every ten steps, the cost of an ALE calculation can often be reduced by a factor of three without adversely affecting the time step size. In general, it is not worthwhile to advect an element unless at least twenty percent of its volume will be transported because the gain in the time step size will not offset the cost of the advection calculations.

17.1 Mesh Smoothing Algorithms

The algorithms for moving the mesh relative to the material control the range of the problems that can be solved by an ALE formulation. The equipotential method which is used in LS-DYNA was developed by Winslow [1990] and is also used in the DYNA2D ALE code [Winslow 1963]. It, and its extensions, have proven to be very successful in a wide variety of problems. The following is extracted from reports prepared by Alan Winslow for LSTC.

17.1.1 Equipotential Smoothing of Interior Nodes

"Equipotential" zoning [Winslow, 1963] is a method of making a structured mesh for finite difference or finite element calculations by using the solutions of Laplace equations (later extended to Poisson equations) as the mesh lines. The same method can be used to smooth selected points in an unstructured three-dimensional mesh provided that it is at least locally structured. This chapter presents a derivation of the three-dimensional equipotential zoning equations, taken from the references, and gives their finite difference equivalents in a form ready to be used for smoothing interior points. We begin by reviewing the well-known two-dimensional zoning equations, and then discuss their extension to three dimensions.

In two dimensions we define curvilinear coordinates ξ,η which satisfy Laplace's equation:

$$\nabla 2\xi = 0 \tag{17.1.1a}$$

$$\nabla 2\eta = 0 \tag{17.1.1b}$$

We solve Equations (16.1.1) *for the coordinates* $x(\xi,\eta)$ *and* $y(\xi,\eta)$ of the mesh lines: that is, we invert them so that the geometric coordinates x, y become the dependent variables and the curvilinear coordinates ξ,η the independent variables. By the usual methods of changing variables we obtain

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = 0 \tag{17.1.2a}$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = 0 \tag{17.1.2b}$$

where

$$\alpha \equiv x_\eta^2 + y_\eta^2, \beta \equiv x_\xi x_\eta + y_\xi y_\eta, \gamma \equiv x_\xi^2 + y_\xi^2 \tag{17.1.3}$$

Equations (17.1.2) can be written in vector form:

$$\alpha\vec{r}_{\xi\xi} - 2\beta\vec{r}_{\xi\eta} + \gamma\vec{r}_{\eta\eta} = 0 \tag{17.1.4}$$

where

$$\vec{r} \equiv x\hat{i} + y\hat{j} \ .$$

We differentiate Equations (17.1.4) and solve them numerically by an iterative method, since they are nonlinear. In (ξ,η) space, we use a mesh whose curvilinear coordinates are straight lines which take on integer values corresponding to the usual numbering in a two-dimensional mesh. The numerical solution then gives us the location of the "equipotential" mesh lines.

In three dimensions $x, y, z,$ we add a third curvilinear coordinate ζ and a third Laplace equation

$$\nabla 2\zeta = 0 \tag{17.1.1c}$$

Inversion of the system of three equations (17.1.1) by change of variable is rather complicated. It is easier, as well as more illuminating, to use the methods of tensor analysis pioneered by Warsi [1982]. Let the curvilinear coordinates be represented by ξ^i *(i=1,2,3)*. For a scalar function *A(x,y,z),* Warsi shows that the transformation of its Laplacian from rectangular Cartesian to curvilinear coordinates is given by

$$\nabla^2 A = \sum_{i,j=1}^{3} g^{ij} A_{\xi^i \xi^j} + \sum_{k=1}^{3} \left(\nabla^2 \xi^k\right) A_{\xi^h} \tag{17.1.5}$$

where a variable subscript indicates differentiation with respect to that variable. Since the curvilinear coordinates are each assumed to satisfy Laplace's equation, the second summation in Equation (17.1.5) vanishes and we have

$$\nabla^2 A = \sum_{i,j=1}^{3} g^{ij} A_{\xi^i \xi^j} \quad . \tag{17.1.6}$$

If now we let $A = x, y,$ and z successively, the left-hand side of (17.1.6) vanishes in each case and we get three equations which we can write in vector form

$$\sum_{i,j=1}^{3} g^{ij} \vec{r}_{\xi^i \xi^j} = 0 \quad . \tag{17.1.7}$$

Equation (17.1.7) is the three-dimensional generalization of Equations (17.1.4), and it only remains to determine the components of the contravariant metric tensor g_{ij} in three dimensions. These are defined to be

$$g^{ij} \equiv \vec{a}^{\,i} \cdot \vec{a}^{\,j} \tag{17.1.8}$$

where the contravariant base vectors of the transformation from *(x,y,z)* to (ξ^1, ξ^2, ξ^3) are given by

$$\vec{a}^{\,i} \equiv \nabla \xi^i = \frac{\vec{a}_j \times \vec{a}_k}{\sqrt{g}} \tag{17.1.9}$$

(i,j,k cyclic). Here the covariant base vectors, the coordinate derivatives, are given by

$$\vec{a}_i \equiv \vec{r}_{\xi^i} \tag{17.1.10}$$

where

$$\vec{r} \equiv x\hat{i} + y\hat{j} + z\hat{k} \ .$$

Also,

$$g \equiv \det g_{ij} = \left[\vec{a}_1 \cdot (\vec{a}_2 \times \vec{a}_3)\right]^2 = J^2 \qquad (17.1.11a)$$

where g_{ij} is the covariant metric tensor given by

$$g_{ij} \equiv \vec{a}_i \cdot \vec{a}_j \qquad (17.1.11b)$$

and J is the Jacobian of the transformation.

Substituting (17.1.9) into (17.1.8), and using the vector identity

$$\left(\vec{a}\times\vec{b}\right)\cdot\left(\vec{c}\times\vec{d}\right) \equiv \left(\vec{a}\cdot\vec{c}\right)\cdot\left(\vec{b}\cdot\vec{d}\right) - \left(\vec{a}\cdot\vec{d}\right)\left(\vec{b}\cdot\vec{c}\right) \qquad (17.1.12)$$

we get

$$gg^{ii} = \vec{a}_j^2\vec{a}_k^2 - \left(\vec{a}_j \cdot \vec{a}_k\right)^2 = g_{jj}g_{kk} - \left(g_{jk}\right)^2 \qquad (17.1.13a)$$

$$gg^{ij} = \left(\vec{a}_i \cdot \vec{a}_k\right)\left(\vec{a}_j \cdot \vec{a}_k\right) - \left(\vec{a}_i \cdot \vec{a}_j\right)\vec{a}_k^2 = g_{ik}g_{jk} - g_{ij}g_{kk} \ . \qquad (17.1.13b)$$

Before substituting (17.1.10) into (17.1.13a, b), we return to our original notation:

$$\xi + \xi^1 \ , \quad \eta + \xi^2 \ , \quad \zeta + \xi^3 \qquad (17.1.14)$$

Then, using (17.1.10), we get

$$gg^{11} = \vec{r}_\eta^{\,2}\vec{r}_\zeta^{\,2} - \left(\vec{r}_\eta \cdot \vec{r}_\zeta\right)^2 \qquad (17.1.15a)$$

$$gg^{22} = \vec{r}_\zeta^{\,2}\vec{r}_\xi^{\,2} - \left(\vec{r}_\zeta \cdot \vec{r}_\xi\right)^2 \qquad (17.1.15b)$$

$$gg^{33} = \vec{r}_\xi^{\,2}\vec{r}_\eta^{\,2} - \left(\vec{r}_\xi \cdot \vec{r}_\eta\right)^2 \qquad (17.1.15c)$$

for the three diagonal components, and

$$gg^{12} = \left(\vec{r}_\xi \cdot \vec{r}_\zeta\right)\left(\vec{r}_\eta \cdot \vec{r}_\zeta\right) - \left(\vec{r}_\xi \cdot \vec{r}_\eta\right)\vec{r}_\zeta^{\,2} \qquad (17.1.16a)$$

$$gg^{23} = \left(\vec{r}_\eta \cdot \vec{r}_\xi\right)\left(\vec{r}_\zeta \cdot \vec{r}_\xi\right) - \left(\vec{r}_\eta \cdot \vec{r}_\zeta\right)\vec{r}_\xi^{\,2} \qquad (17.1.16b)$$

$$gg^{31} = \left(\vec{r}_\zeta \cdot \vec{r}_\eta\right)\left(\vec{r}_\xi \cdot \vec{r}_\eta\right) - \left(\vec{r}_\zeta \cdot \vec{r}_\xi\right)\vec{r}_\eta^{\,2} \qquad (17.1.16c)$$

for the three off-diagonal components of this symmetric tensor.

When we express Equations (17.1.15) in terms of the Cartesian coordinates, some cancellation takes place and we can write them in the form

$$gg^{11} = (x_\eta y_\zeta - x_\zeta y_\eta)^2 + (x_\eta z_\zeta - x_\zeta z_\eta)^2 + (y_\eta z_\zeta - y_\zeta z_\eta)^2 \qquad (17.1.17a)$$

$$gg^{22} = (x_\zeta y_\xi - x_\xi y_\zeta)^2 + (x_\zeta z_\xi - x_\xi z_\zeta)^2 + (y_\zeta z_\xi - y_\xi z_\zeta)^2 \qquad (17.1.17b)$$

$$gg^{33} = (x_\xi y_\eta - x_\eta y\xi)^2 + (x_\xi z_\eta - x_\eta z_\xi)^2 + (y_\xi z_\eta - y_\eta z_\xi)^2 \qquad (17.1.17c)$$

guaranteeing positivity as required by Equations (17.1.8). Writing out Equations (17.1.16) we get

$$\begin{aligned} gg^{12} &= (x_\xi x_\zeta + y_\xi y_\zeta + z_\xi z_\zeta)(x_\eta x_\zeta + y_\eta y_\zeta + z_\eta z_\zeta) \\ &\quad - (x_\xi x_\eta + y_\xi y_\eta + z_\xi z_\eta)\left(x_\zeta^2 + y_\zeta^2 + z_\zeta^2\right) \end{aligned} \qquad (17.1.18a)$$

$$\begin{aligned} gg^{23} &= (x_\eta x_\xi + y_\eta y_\xi + z_\eta z_\xi)(x_\zeta x_\xi + y_\zeta y_\xi + z_\zeta z_\xi) \\ &\quad - (x_\eta x_\zeta + y_\eta y_\zeta + z_\eta z_\zeta)\left(x_\xi^2 + y_\xi^2 + z_\xi^2\right) \end{aligned} \qquad (17.1.18b)$$

$$\begin{aligned} gg^{31} &= (x_\zeta x_\eta + y_\zeta y_\eta + z_\zeta z_\eta)(x_\xi x_\eta + y_\xi y_\eta + z_\xi z_\eta) \\ &\quad - (x_\zeta x_\xi + y_\zeta y_\xi + z_\zeta z_\xi)\left(x_\eta^2 + y_\eta^2 + z_\eta^2\right) \end{aligned} \qquad (17.1.18c)$$

Hence, we finally write Equations (17.1.7) in the form

$$g\left(g^{11}\vec{r}_{\xi\xi} + g^{22}\vec{r}_{\eta\eta} + g^{33}\vec{r}_{\zeta\zeta} + 2g^{12}\vec{r}_{\xi\eta} + 2g^{23}\vec{r}_{\eta\zeta} + 2g^{31}\vec{r}_{\zeta\xi}\right) = 0 \qquad (17.19)$$

where the gg^{ij} are given by Equations (17.1.17) and (17.1.18). Because Equations (17.1.7) are homogeneous, we can use gg^{ij} in place of g^{ij} as long as g is positive, as it must be for a nonsingular transformation. We can test for positivity at each mesh point by using Equation (17.1.20):

$$\sqrt{g} = J = \begin{vmatrix} x_\xi & y_\xi & z_\xi \\ x_\eta & y_\eta & z_\eta \\ x_\zeta & y_\zeta & z_\zeta \end{vmatrix} \qquad (17.1.20)$$

and requiring that $J > 0$.

To check that these equations reproduce the two-dimensional equations when there is no variation in one-dimension, we take ζ as the invariant direction, thus reducing (17.1.19) to

$$gg^{11}\vec{r}_{\xi\xi} + 2gg^{12}\vec{r}_{\xi\eta} + gg^{22}\vec{r}_{\eta\eta} = 0 \tag{17.1.21}$$

If we let $\zeta = z$, then the covariant base vectors become

$$\vec{a}_1 = x_\xi \vec{i} + y_\xi \vec{j} \tag{17.1.22a}$$

$$\vec{a}_2 = x_\eta \vec{i} + y_\eta \vec{j} \tag{17.1.22b}$$

$$\vec{a}_3 = \vec{k} \tag{17.1.22c}$$

From (17.1.22), using (17.1.13), we get

$$gg^{11} = x_\eta^2 + y_\eta^2 \tag{17.1.23a}$$

$$gg^{22} = x_\xi^2 + y_\xi^2 \tag{17.1.23b}$$

$$gg^{12} = -(x_\xi x_\eta + y_\xi y_\eta) \ . \tag{17.1.23c}$$

Substituting (17.1.23) into (17.1.21) yields the two-dimensional equipotential zoning Equations (17.1.2).

Before differencing Equations (17.1.19) we simplify the notation and write them in the form

$$\alpha_1\vec{r}_{\xi\xi} + \alpha_2\vec{r}_{\eta\eta} + \alpha_3\vec{r}_{\zeta\zeta} + 2\beta_1\vec{r}_{\xi\eta} + 2\beta_2\vec{r}_{\eta\zeta} + 2\beta_3\vec{r}_{\zeta\xi} = 0 \tag{17.1.24}$$

where

$$\alpha_1 = (x_\eta y_\zeta - x_\zeta y_\eta)^2 + (x_\eta z_\zeta - x_\zeta z_\eta)^2 + (y_\eta z_\zeta - y_\zeta z_\eta)^2 \tag{17.1.25a}$$

$$\alpha_2 = (x_\zeta y_\xi - x_\xi y_\zeta)^2 + (x_\zeta z_\xi - x_\xi z_\zeta)^2 + (y_\zeta z_\xi - y_\xi z_\zeta)^2 \tag{17.1.25b}$$

$$\alpha_3 = (x_\xi y_\eta - x_\eta y_\xi)^2 + (x_\xi z_\eta - x_\eta z_\xi)^2 + (y_\xi z_\eta - y_\eta z_\xi)^2 \tag{17.1.25c}$$

$$\begin{aligned}\beta_1 = {} & (x_\xi x_\zeta + y_\xi y_\zeta + z_\xi z_\zeta)(x_\eta x_\zeta + y_\eta y_\zeta + z_\eta z_\zeta) \\ & -(x_\xi x_\eta + y_\xi y_\eta + z_\xi z_\eta)(x_\zeta^2 + y_\zeta^2 + z_\zeta^2)\end{aligned} \tag{17.1.25d}$$

$$\begin{aligned}\beta_2 = {} & (x_\eta x_\xi + y_\eta y_\xi + z_\eta z_\xi)(x_\xi x_\zeta + y_\xi y_\zeta + z_\xi z_\zeta) \\ & -(x_\zeta x_\eta + y_\zeta y_\eta + z_\zeta z_\eta)(x_\xi^2 + y_\xi^2 + z_\xi^2)\end{aligned} \tag{17.1.25e}$$

$$\beta_3 = (x_\eta x_\zeta + y_\eta y_\zeta + z_\eta z_\zeta)(x_\xi x_\eta + y_\xi y_\eta + z_\xi z_\eta) - (x_\xi x_\eta + y_\xi y_\eta + z_\xi z_\eta)(x_\eta^2 + y_\eta^2 + z_\eta^2) \quad (17.1.25f)$$

We difference Equations (17.1.24) in a cube in the rectangular $\xi\eta\zeta$ space with unit spacing between the coordinate surfaces, using subscript i to represent the ξ direction, j the η direction, and k the ζ direction, as shown in Figure 17.1.

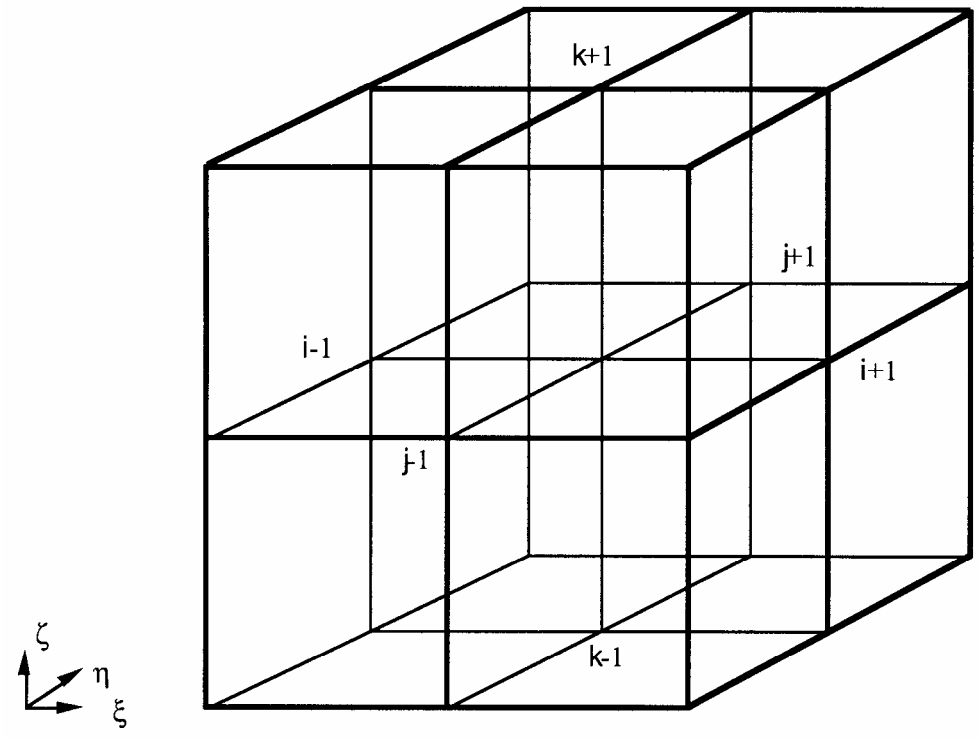


Figure 17.l.

Using central differencing, we obtain the following finite difference approximations for the coordinate derivatives:

$$\vec{r}_\xi = (\vec{r}_{i+1} - \vec{r}_{i-1})/2 \quad (17.1.26a)$$

$$\vec{r}_\eta = (\vec{r}_{j+1} - \vec{r}_{j-1})/2 \quad (17.1.26b)$$

$$\vec{r}_\zeta = (\vec{r}_{k+1} - \vec{r}_{k-1})/2 \quad (17.1.26c)$$

$$\vec{r}_{\xi\xi} = (\vec{r}_{i+1} - 2\vec{r} + \vec{r}_{i-1}) \quad (17.1.26d)$$

$$\vec{r}_{\eta\eta} = (\vec{r}_{j+1} - 2\vec{r} + \vec{r}_{j-1}) \quad (17.1.26e)$$

$$\vec{r}_{\zeta\zeta} = \left(\vec{r}_{k+1} - 2\vec{r} + \vec{r}_{k-1}\right) \tag{17.1.26f}$$

$$\vec{r}_{\xi\eta} = \frac{1}{4}\left[\left(\vec{r}_{i+1,j+1} + \vec{r}_{i-1,j-1}\right) - \left(\vec{r}_{i+1,j-1} + \vec{r}_{i-1,j+1}\right)\right] \tag{17.1.26g}$$

$$\vec{r}_{\eta\zeta} = \frac{1}{4}\left[\left(\vec{r}_{j+1,k+1} + \vec{r}_{j-1,k-1}\right) - \left(\vec{r}_{j+1,k-1} + \vec{r}_{j-1,k+1}\right)\right] \tag{17.1.26h}$$

$$\vec{r}_{\zeta\xi} = \frac{1}{4}\left[\left(\vec{r}_{I+1,k+1} + \vec{r}_{I-1,k-1}\right) - \left(\vec{r}_{I+1,k-1} + \vec{r}_{I-1,k+1}\right)\right] \tag{17.1.26i}$$

where for brevity we have omitted subscripts $i, j,$ or k (e.g., $k+1$ stands for $i, j, k+1$). Note that these difference expressions use only coordinate planes that pass through the central point, and therefore do not include the eight corners of the cube.

Substituting Equations (17.1.26) into (17.1.24,17.1.25) and collecting terms, we get

$$\sum_{m=1}^{18} \omega_m(\vec{r}_m - \vec{r}) = 0 \tag{17.1.27}$$

where the sum is over the 18 nearest (in the transform space) neighbors of the given point. The coefficients ω_m are given in Table 17.1.

Equations (17.1.27) can be written

$$\vec{r}_m = \frac{\sum_m \omega_m \vec{r}_m}{\sum_m \omega_m} \tag{17.1.28}$$

expressing the position of the central point as a weighted mean of its 18 nearest neighbors. The denominator of (17.1.28) is equal to $2(\alpha_1+\alpha_2+\alpha_3)$ which is guaranteed to be positive by (17.1.25). This vector equation is equivalent to the three scalar equations

$$x = \frac{\sum_m \omega_m x_m}{\sum_m \omega_m} \tag{17.1.29a}$$

$$y = \frac{\sum_m \omega_m y_m}{\sum_m \omega_m} \tag{17.1.29b}$$

m	Index	ω_m
1	i+1	α_1
2	i–1	α_1
3	j+1	α_2
4	j–1	α_2
5	k+1	α_3
6	k–1	α_3
7	i+1, j+1	$\beta_1/2$
8	i–1, j–1	$\beta_1/2$
9	i+1, j–1	$-\beta_1/2$
10	i–1, j+1	$-\beta_1/2$
11	j+1, k+1	$\beta_2/2$
12	j–1, k–1	$\beta_2/2$
13	j+1, k–1	$-\beta_2/2$
14	j–1, k+1	$-\beta_2/2$
15	i+1, k+1	$\beta_3/2$
16	i–1, k–1	$\beta_3/2$
17	i+1, k–1	$-\beta_3/2$
18	i–1, k+1	$-\beta_3/2$

Table 17.1. 3D Zoning Weight Coefficients

$$z = \frac{\sum_m \omega_m z_m}{\sum_m \omega_m} \tag{17.1.29c}$$

the same weights ω_m appearing in each equation.

These equations are nonlinear, since the coefficients are functions of the coordinates. Therefore we solve them by an iterative scheme, such as SOR. When applied to Equation (17.1.29a), for example, this gives for the $(n+1)^{st}$ iteration

$$x^{n+1} = (1-f)x^n + f\left(\frac{\sum_m \omega_m x_m}{\sum_m \omega_m}\right) \tag{17.1.30}$$

where the over relaxation factor f must satisfy $0<f<2$. In (17.1.30) the values of x_m at the neighboring points are the latest available values. The coefficients ω_m are recalculated before each iteration using Table 17.1 and Equations (17.1.25).

To smooth one interior point in a three-dimensional mesh, let the point to be smoothed be the interior point of Figure 17.1, assuming that its neighborhood has the logical structure shown. Even though Equations (17.1.29) are nonlinear, the ω_m do not involve the coordinates of the central point, since the α's and β's do not. Hence we simply solve Equations (17.1.29) for the new coordinates (x, y, z), holding the 18 neighboring points fixed, without needing to iterate.

If we wish to smooth a group of interior points, we solve iteratively for the coordinates using equations of the form (17.1.30).

17.1.2 Simple Averaging

The coordinates of a node is the simple average of the coordinates of its surrounding nodes.

$$\vec{x}_{SA}^{n+1} = \frac{1}{m^{tot}} \sum_{m=1}^{m^{tot}} \vec{x}_m^n \tag{17.1.31}$$

17.1.3 Kikuchi's Algorithm

Kikuchi proposed an algorithm that uses a volume-weighted average of the coordinates of the centroids of the elements surrounding a node. Variables that are subscripted with Greek letters refer to element variables, and subscripts with capital letters refer to the local node numbering within an element.

$$\vec{x}_\alpha^n = \frac{1}{8} \sum_A \vec{x}_A^n \tag{17.1.32a}$$

$$\vec{x}_K^{n+1} = \frac{\sum_{\alpha=1}^{\alpha^{tot}} V_a \vec{x}_\alpha^n}{\sum_{\alpha=1}^{\alpha^{tot}} V_a} \tag{17.1.32b}$$

17.1.4 Surface Smoothing

The surfaces are smoothed by extending the two-dimensional equipotential stencils to three dimensions. Notice that the form of Equation (17.1.2a) and (17.1.2b) for the x and y directions are identical. The third dimension, z, takes the same form. When Equation (16.1.2) is applied to all three dimensions, it tends to flatten out the surface and alter the total volume. To conserve the volume and retain the curvature of the surface, the point given by the relaxation stencil is projected on to the tangent plane defined by the normal at the node.

17.1.5 Combining Smoothing Algorithms

The user has the option of using a weighted average of all three algorithms to generate a composite algorithm, where the subscripts E, SA, and K refer to the equipotential, simple averaging, and Kikuchi's smoothing algorithm respectively, and w w is the weighting factor.

$$\vec{x}^{n+1} = w_E \vec{x}_E^{n+1} + w_{SA} \vec{x}_{SA}^{n+1} + w_K \vec{x}_K^{n+1} \tag{17.1.33}$$

17.2 Advection Algorithms

LS-DYNA follows the SALE3D strategy for calculating the transport of the element-centered variables (i.e., density, internal energy, the stress tensor and the history variables). The van Leer MUSCL scheme [van Leer 1977] is used instead of the donor cell algorithm to calculate the values of the solution variables in the transport fluxes to achieve second order accurate monotonic results. To calculate the momentum transport, two algorithms have been implemented. The less expensive of the two is the one that is implemented in SALE3D, but it has known dispersion problems and may violate monotonocity (i.e., introduce spurious oscillations) [Benson 1992]. As an alternative, a significantly more expensive method [Benson 1992], which can be shown analytically to not have either problem on a regular mesh, has also been implemented.

In this section the donor cell and van Leer MUSCL scheme are discussed. Both methods are one-dimensional and their extensions to multidimensional problems are discussed later.

17.2.1 Advection Methods in One Dimension

In this section the donor cell and van Leer MUSCL scheme are discussed. Both methods are one-dimensional and their extensions to multidimensional problems are discussed later.

The remap step maps the solution from a distorted Lagrangian mesh on to the new mesh. The underlying assumptions of the remap step are 1) the topology of the mesh is fixed (a complete rezone does not have this limitation), and 2) the mesh motion during a step is less than the characteristic lengths of the surrounding elements. Within the fluids community, the second condition is simply stated as saying the Courant number, C, is less than one.

$$C = \frac{u\Delta t}{\Delta x} = \frac{f}{V} \le 1 \tag{17.2.1}$$

Since the mesh motion does not occur over any physical time scale, Δt is arbitrary, and $u\Delta t$ is the transport volume, f, between adjacent elements. The transport volume calculation is purely geometrical for ALE formulations and it is not associated with any of the physics of the problem.

The algorithms for performing the remap step are taken from the computational fluids dynamics community, and they are referred to as “advection” algorithms after the first order, scalar conservation equation that is frequently used as a model hyperbolic problem.

$$\frac{\partial \phi}{\partial t} + a(x)\frac{\partial \phi}{\partial x} = 0 \tag{17.2.2}$$

A good advection algorithm for the remap step is accurate, stable, conservative and monotonic. Although many of the solution variables, such as the stress and plastic strain, are not governed by conservation equations like momentum and energy, it is still highly desirable that the volume integral of all the solution variables remain unchanged by the remap step. Monotonicity requires that the range of the solution variables does not increase during the remap. This is particularly important with mass and energy, where negative values would lead to physically unrealistic solutions.

Much of the research on advection algorithms has focused on developing monotonic algorithms with an accuracy that is at least second order. Not all recent algorithms are monotonic. For example, within the finite element community, the streamline upwind Petrov-Galerkin (SUPG) method developed by Hughes and coworkers [Brooks and Hughes 1982] is not monotonic. Johnson et al., [1984] have demonstrated that the oscillations in the SUPG solution are localized, and its generalization to systems of conservation equations works very well for the Euler equations. Mizukami and Hughes [1985] later developed a monotonic SUPG formulation. The essentially non-oscillatory (ENO) [Harten 1989] finite difference algorithms are also not strictly monotonic, and work well for the Euler equations, but their application to hydrodynamics problems has resulted in negative densities [McGlaun 1990]. Virtually all the higher order methods that are commonly used were originally developed for solving the Euler equations, usually as higher order extensions to Godunov's method. Since the operator split approach is the dominant one in Eulerian hydrocodes, these methods are implemented only to solve the scalar advection equation.

The Donor Cell Algorithm. Aside from its first order accuracy, it is everything a good advection algorithm should be: stable, monotonic, and simple. The value of f_j^{ϕ} is dependent on the sign of a at node j, which defines the upstream direction.

$$\phi_{j+1/2}^{n+1} = \phi_{j+1/2}^{n} + \frac{\Delta t}{\Delta x}\left(f_j^{\phi} - f_{j+1}^{\phi}\right) \tag{17.2.3a}$$

$$f_j^{\phi} = \frac{a_j}{2}\left(\phi_{j-1/2}^{n} + \phi_{j+1/2}^{n}\right) + \frac{|a_j|}{2}\left(\phi_{j-1/2}^{n} - \phi_{j+1/2}^{n}\right) \tag{17.2.3b}$$

The donor cell algorithm is a first order Godunov method applied to the advection equation. The initial values of ϕ to the left and the right of node j are $\phi_{j-1/2}^{n}$ and $\phi_{j+1/2}^{n}$, and the velocity of the contact discontinuity at node j is a_j.

The Van Leer MUSCL Algorithm. Van Leer [1977] introduced a family of higher order Godunov methods by improving the estimates of the initial values of left and right states for the Riemann problem at the nodes. The particular advection algorithm that is presented in this section is referred to as the MUSCL (monotone upwind schemes for conservation laws) algorithm for brevity, although MUSCL really refers to the family of algorithms that can be applied to systems of equations.

The donor cell algorithm assumes that the distribution of ϕ is constant over an element. Van Leer replaces the piecewise constant distribution with a higher order interpolation function, $\phi_{j+1/2}^{n}(x)$ that is subject to an element level conservation constraint. The value of ϕ at the element centroid is regarded in this context as the average value of ϕ over the element instead of the spatial value at $x_{j+1/2}$.

$$\phi^n_{j+1/2} = \int_{x_j}^{x_{j+1}} \phi^n_{j+1/2}(x)dx \tag{17.2.4}$$

To determine the range of ϕ, $\left[\phi^{\min}_{j+1/2}, \phi^{\max}_{j+1/2}\right]$, for imposing the monotonicity constraint, the maximum and minimum values of $\phi^n_{j-1/2}$, $\phi^n_{j+1/2}$, and $\phi^n_{j+3/2}$ are used. Monotonicity can be imposed in either of two ways. The first is to require that the maximum and minimum values of $\phi^n_{j+1/2}(x)$ fall within the range determined by the three elements. The second is to restrict the average value of ϕ in the transport volumes associated with element $j + 1/2$. While the difference may appear subtle, the actual difference between the two definitions is quite significant even at relatively low Courant numbers. The second definition allows the magnitude of the ϕ transported to adjacent elements to be larger than the first definition. As a consequence, the second definition is better able to transport solutions with large discontinuities. The magnitude of ϕ an algorithm is able to transport before its monotonicity algorithm restricts ϕ is a measure of the algorithm's "compressiveness."

The first step up from a piecewise constant function is a piecewise linear function, where x is now the volume coordinate. The volume coordinate of a point is simply the volume swept along the path between the element centroid and the point. Conservation is guaranteed by expanding the linear function about the element centroid.

$$\phi^n_{j+1/2}(x) = S^n_{j+1/2}(x - x^n_{j+1/2}) + \phi^n_{j+1/2} \tag{17.2.5}$$

Letting $s^n_{j+1/2}$ be a second order approximation of the slope, the monotonicity limited value of the slope, $s^n_{j+1/2}$, according to the first limiting approach, is determined by assuming the maximum permissible values at the element boundaries.

$$S^n_{j+1/2} = \frac{1}{2}\left(\text{sgn}(s^L) + \text{sgn}(s^R)\right) \times \min\left(\left|s^L\right|, \left|s^n_{j+1/2}\right|, \left|s^R\right|\right) \tag{17.2.6a}$$

$$s^L = \frac{\phi^n_{j+1/2} - \phi^n_{j-1/2}}{\frac{1}{2}\Delta x_{j+1/2}} \tag{17.2.6b}$$

$$s^R = \frac{\phi^n_{j+3/2} - \phi^n_{j+1/2}}{\frac{1}{2}\Delta x_{j+1/2}} \tag{17.2.6c}$$

The second limiter is similar to the first, but it assumes that the maximum permissible values occur at the centroid of the transport volumes. Note that as stated in Equation (17.2.6),

this limiter still limits the slope at the element boundary even if the element is the downstream element at that boundary. A more compressive limiter would not limit the slope based on the values of ϕ at the downstream boundaries. For example, if a_j is negative, only s^R would limit the value of s^n in Equation (17.2.6). If the element is the downstream element at both boundaries, then the slope in the element has no effect on the solution.

$$s^L = \frac{\phi^n_{j+1/2} - \phi^n_{j-1/2}}{\frac{1}{2}\Delta x_{j+1/2} - \frac{1}{2}\max(0, a_j \Delta t)} \tag{17.2.7a}$$

$$s^R = \frac{\phi^n_{j+3/2} - \phi^n_{j+1/2}}{\frac{1}{2}x_{j+1/2} + \frac{1}{2}\min(0, a_{j+1} \Delta t)} \tag{17.2.7b}$$

The flux at node j is evaluated using the upstream approximation of ϕ.

$$f^{\phi}_j = \frac{a_j}{2}\left(\phi^-_j + \phi^+_j\right) + \frac{|a_j|}{2}\left(\phi^-_j - \phi^+_j\right) \tag{17.2.8a}$$

$$\phi^+_j = S^n_{j+1/2}(x^C - x^n_{j+1/2}) + \phi^n_{j+1/2} \tag{17.2.8b}$$

$$\phi^-_j = S^n_{j-1/2}(x^C - x^n_{j-1/2}) + \phi^n_{j-1/2} \tag{17.2.8c}$$

$$x^C = x^n_j + \frac{1}{2}\Delta t a_j \tag{17.2.8d}$$

The method for obtaining the higher order approximation of the slope is not unique. Perhaps the simplest approach is to fit a parabola through the centroids of the three adjacent elements and evaluate its slope at $x^n_{j+1/2}$. When the value of ϕ at the element centroids is assumed to be equal to the element average this algorithm defines a projection.

$$s^n_{j+1/2} = \frac{\left(\phi^n_{j+3/2} - \phi^n_{j+1/2}\right)\Delta x^2_j + \left(\phi^n_{j+1/2} - \phi^n_{j-1/2}\right)\Delta x^2_{j+1}}{\Delta x_j \Delta x_{j+1}\left(\Delta x_j + \Delta x_{j+1}\right)} \tag{17.2.9a}$$

$$\Delta x_j = x^n_{j+1/2} - x^n_{j-1/2} \tag{17.2.9b}$$

17.2.2 Advection Methods in Three Dimensions

For programs that use a logically regular mesh, one-dimensional advection methods are extended to two and three dimensions by using a sequence of one-dimensional sweeps along the logically orthogonal mesh lines. This strategy is not possible for unstructured meshes because they don't have uniquely defined sweep directions through the mesh. CAVEAT [Addessio, et al., 1986] uses one-dimensional sweeps in the spatial coordinate system, but their approach is expensive relative to the other algorithms and it does not always maintain spherical symmetry, which is an important consideration in underwater explosion calculations.

The advection in LS-DYNA is performed isotropically. The fluxes through each face of element A are calculated simultaneously, but the values of ϕ in the transport volumes are calculated using the one-dimensional expressions developed in the previous sections.

$$\phi_A^{n+1} = \frac{1}{V_A^{n+1}}\left(V_A^n \phi_A^n + \sum_{j=1}^{6} f_j^{\phi} \right) \tag{17.2.10}$$

The disadvantage of isotropic advection is that there is no coupling between an element and the elements that are joined to it only at its corners and edges (i.e., elements that don't share faces). The lack of coupling introduces a second order error that is significant only when the transport is along the mesh diagonals.

The one-dimensional MUSCL scheme, which requires elements on either side of the element whose transport is being calculated, cannot be used on the boundary elements in the direction normal to the boundary. Therefore, in the boundary elements, the donor cell algorithm is used to calculate the transport in the direction that is normal to the boundary, while the MUSCL scheme is used in the two tangential directions.

It is implicitly assumed by the transport calculations that the solution variables are defined per unit current volume. In LS-DYNA, some variables, such as the internal energy, are stored in terms of the initial volume of the element. These variables must be rescaled before transport, then the initial volume of the element is advected between the elements, and then the variables are rescaled using the new "initial" volumes. Hyperelastic materials are not currently advected in LS-DYNA because they require the deformation gradient, which is calculated from the initial geometry of the mesh. If the deformation gradient is integrated by using the midpoint rule, and it is advected with the other solution variables, then hyperelastic materials can be advected without any difficulties.

$$F^{n+1} = (I - \frac{\Delta t}{2L^{n+1/2}})^{-1}(I + \frac{\Delta t}{2L^{n+1/2}})F^n \tag{17.2.11}$$

Advection of the Nodal Velocities. Except for the Godunov schemes, the velocity is centered at the nodes or the edges while the remaining variables are centered in the elements. Momentum is advected instead of the velocity in most codes to guarantee that momentum is conserved. The element-centered advection algorithms must be modified to advect the node-centered momentum. Similar difficulties are encountered when node-centered algorithms, such as the SUPG method [Brooks and Hughes 1982], are applied to element-centered quantities [Liu, Chang, and Belytschko, *to be published*]. There are two approaches: 1) construct a new mesh such that the nodes become the element centroids of the new mesh and apply the element-

centered advection algorithms, and 2) construct an auxiliary set of element-centered variables from the momentum, advect them, and then reconstruct the new velocities from the auxiliary variables. Both approaches can be made to work well, but their efficiency is heavily dependent on the architecture of the codes. The algorithms are presented in detail for one dimension first for clarity. Their extensions to three dimensions, which are presented later, are straightforward even if the equations do become lengthy. A detailed discussion of the algorithms in two dimensions is presented in Reference [Benson 1992].

Notation. Finite difference notation is used in this section so that the relative locations of the nodes and fluxes are clear. The algorithms are readily applied, however, to unstructured meshes. To avoid limiting the discussion to a particular element-centered advection algorithm, the transport volume through node i is f, the transported mass is $\tilde{f}_i$, and the flux of ϕ is $\phi_i f_i$. Most of the element-centered flux-limited advection algorithms calculate the flux of ϕ directly, but the mean value of ϕ in the transport volumes is calculated by dividing the $\phi_i f_i$, by the transport volume. A superscript "-" or "+" denotes the value of a variable before or after the advection. Using this notation, the advection of ϕ in one dimension is represented by Equation (17.2.12), where the volume is V.

$$\phi^+_{j+\frac{1}{2}} = \frac{(\phi^-_{j+\frac{1}{2}} V^-_{j+\frac{1}{2}} + \phi_i f_i - \phi_{i+1} f_{i+1})}{V^+} \qquad (17.2.12a)$$

$$V^+_{j+\frac{1}{2}} = V^-_{j+\frac{1}{2}} + f_i - f_{i+1} \qquad (17.2.12b)$$

The Staggered Mesh Algorithm. YAQUI [Amsden and Hirt 1973] was the first code to construct a new mesh that is staggered with respect to original mesh for momentum advection. The new mesh is defined so that the original nodes become the centroids of the new elements. The element-centered advection algorithms are applied to the new mesh to advect the momentum. In theory, the momentum can be advected with the transport volumes or the velocity can be advected with the mass.

$$v^+_j = \frac{(M^-_j v^-_j + v_{j-\frac{1}{2}} \tilde{f}_{j-\frac{1}{2}} - v_{j+\frac{1}{2}} \tilde{f}_{j+\frac{1}{2}})}{M^+_j} \qquad (17.2.13a)$$

$$v^+_j = \frac{(M^-_j v^-_j + \{\rho v\}_{j-\frac{1}{2}} f_{j-\frac{1}{2}} - \{\rho v\}_{j+\frac{1}{2}} f_{j-\frac{1}{2}})}{M^+_j} \qquad (17.2.13b)$$

$$M^+_j = M^-_j + \tilde{f}_{j-\frac{1}{2}} - \tilde{f}_{j+\frac{1}{2}} \qquad (17.2.13c)$$

A consistency condition, first defined by DeBar [1974], imposes a constraint on the formulation of the staggered mesh algorithm: if a body has a uniform velocity and a spatially varying density

before the advection, then the velocity should be uniform and unchanged after the advection. The new mass of a node can be expressed in terms of the quantities used to advect the element-centered mass.

$$M_j^+ = \frac{1}{2}(M_{j-1/2}^+ + M_{j+1/2}^+) \tag{17.2.14}$$

$$M_j^+ = \frac{1}{2}(M_{j-1/2}^- + \rho_{j-1} f_{j-1} - \rho_j f_j + M_{j+1/2}^- + \rho_j f_j - \rho_{j+1} f_{j+1}) \tag{17.2.15a}$$

$$M_j^+ = M_j^- + \frac{1}{2}[(\rho_{j-1} f_{j-1} - \rho_j f_j) + (\rho_j f_j - \rho_{j+1} f_{j+1})] \tag{17.2.15b}$$

The staggered mass fluxes and transport volumes are defined by equating Equation (17.2.14) and Equation (17.2.15).

$$\rho_{j+1/2} f_{j+1/2} = \tilde{f}_{j+1/2} = \frac{1}{2}(\rho_j f_j + \rho_{j+1} f_{j+1}) \tag{17.2.16}$$

The density $\rho_{j+1/2}$ is generally a nonlinear function of the volume $f_{j+1/2}$, hence calculating $f_{j+1/2}$ from Equation (17.2.16) requires the solution of a nonlinear equation for each transport volume. In contrast, the mass flux is explicitly defined by Equation (17.2.16). Most codes, including KRAKEN [Debar 1974], CSQ [Thompson 1975], CTH [McGlaun 1989], and DYNA2D [Hallquist 1980], use mass fluxes with the staggered mesh algorithm because of their simplicity.

The dispersion characteristics of this algorithm are identical to the underlying element-centered algorithm by construction. This is not true, however, for some of the element-centered momentum advection algorithms. There are some difficulties in implementing the staggered mesh method in multi-dimensions. First, the number of edges defining a staggered element equals the number of elements surrounding the corresponding node. On an unstructured mesh, the arbitrary connectivity results in an arbitrary number of edges for each staggered element. Most of the higher order accurate advection algorithms assume a logically regular mesh of quadrilateral elements, making it difficult to use them with the staggered mesh. Vectorization also becomes difficult because of the random number of edges that each staggered element might have. In the ALE calculations of DYNA2D, only the nodes that have a locally logically regular mesh surrounding them can be moved in order to avoid these difficulties [Benson 1992]. These difficulties do not occur in finite difference codes which process logically regular blocks of zones. Another criticism is the staggered mesh algorithm tends to smear out shocks because not all the advected variables are element-centered [Margolin 1989]. This is the primary reason, according to Margolin [1989], that the element-centered algorithm was adopted in SALE [Amdsden, Ruppel, and Hirt 1980].

The SALE Algorithm. SALE advects an element-centered momentum and redistributes its changes to the nodes [Amdsden, Ruppel, and Hirt 1980]. The mean element velocity, $\bar{v}_{j+1/2}$,

specific momentum, $p_{j+½}$, element momentum, $P_{j+½}$, and nodal momentum are defined by Equation (17.2.17).

$$\bar{v}_{j+½} = \frac{1}{2}(v_j + v_{j+1}) \tag{17.2.17a}$$

$$p_{j+½} = \rho_{j+½}\bar{v}_{j+½} \tag{17.2.17b}$$

$$P_{j+½} = M_{j+½}\bar{v}_{j+½} \tag{17.2.17c}$$

Denoting the change in the element momentum $\Delta P_{j+½}$, the change in the velocity at a node is calculated by distributing half the momentum change from the two adjacent elements.

$$\Delta P_{j-½} = p_{j-1} f_{j-1} - p_j f_j \tag{17.2.18a}$$

$$P_j^+ = P_j^- + \frac{1}{2}(\Delta P_{j-½} + \Delta P_{j+½}) \tag{17.2.18b}$$

$$v_j^+ = \frac{P_j^+}{M_j^+} \tag{17.2.18c}$$

This algorithm can also be implemented by advecting the mean velocity, $\bar{v}_{j+½}$ with the transported mass, and the transported momentum $p_j f_j$ is changed to $\bar{v}_j \tilde{f}_j$.

The consistency condition is satisfied regardless of whether masses or volumes are used. Note that the velocity is not updated from the updated values of the adjacent element momenta. The reason for this is the original velocities are not recovered if $f_i = 0$, which indicates that there is an inversion error associated with the algorithm.

The HIS (Half Index Shift) Algorithm. Benson [1992] developed this algorithm based on his analysis of other element-centered advection algorithms. It is designed to overcome the dispersion errors of the SALE algorithm and to preserve the monotonicity of the velocity field. The SALE algorithm is a special case of a general class of algorithms. To sketch the idea behind the HIS algorithm, the discussion is restricted to the scalar advection equation. Two variables, $\Psi_{1,j+½}$ and $\Psi_{2,j+½}$ are defined in terms of a linear transformation of ϕ_j and ϕ_{j+1}. The linear transformation may be a function of the element $j + 1/2$.

$$\begin{Bmatrix} \Psi^-_{1,j+½} \\ \Psi^-_{2,j+½} \end{Bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{Bmatrix} \phi^-_j \\ \phi^-_{j+1} \end{Bmatrix} \tag{17.2.19}$$

This relation is readily inverted.

$$\begin{Bmatrix} \phi_j^+ \\ \phi_{j+1}^+ \end{Bmatrix} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}\begin{Bmatrix} \Psi_{1,j+1/2}^+ \\ \Psi_{2,j+1/2}^+ \end{Bmatrix} \tag{17.2.20}$$

A function is monotonic over an interval if its derivative does not change sign. The sum of two monotonic functions is monotonic, but their difference is not necessarily monotonic. As a consequence, $\Psi_{1,j+1/2}^-$ and $\Psi_{2,j+1/2}^-$ are monotonic over the same intervals as ϕ_j^- if all the coefficients in the linear transformation have the same sign. On the other hand, ϕ_j^+ is not necessarily monotonic even if $\Psi_{1,j+1/2}^+$ and $\Psi_{2,j+1/2}^+$ are monotonic because of the appearance of the negative signs in the inverse matrix. Monotonicity can be maintained by transforming in both directions provided that the transformation matrix is diagonal. Symmetry in the overall algorithm is obtained by using a weighted average of the values of ϕ_j calculated in elements $j + 1/2$ and $j - 1/2$.

A monotonic element-centered momentum advection algorithm is obtained by choosing the identity matrix for the transformation and by using mass weighting for the inverse relationship.

$$\begin{Bmatrix} \Psi_{1,j+1/2} \\ \Psi_{2,j+1/2} \end{Bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{Bmatrix} v_j^- \\ v_{j+1}^- \end{Bmatrix} \tag{17.2.21}$$

To conserve momentum, Ψ is advected with the transport masses.

$$\Psi_{m,j+1/2}^+ = \left(M_{j+1/2}^- \Psi_{m,j+1/2}^- + \Psi_{m,j}^- \tilde{f}_j - \Psi_{m,j+1}^- \tilde{f}_{j+1}\right)/M_{j+1/2}^+ \tag{17.2.22}$$

$$v_j = \frac{1}{2M_j}\left(M_{j+1/2}\Psi_{1,j+1/2} + M_{j-1/2}\Psi_{2,j-1/2}\right) \tag{17.2.23}$$

Dispersion Errors. A von Neumann analysis [Trefethen 1982] characterizes the dispersion errors of linear advection algorithms. Since the momentum advection algorithm modifies the underlying element-centered advection algorithm, the momentum advection algorithm does not necessarily have the same dispersion characteristics as the underlying algorithm. The von Neumann analysis provides a tool to explore the changes in the dispersion characteristics without considering a particular underlying advection algorithm.

The model problem is the linear advection equation with a constant value of c. A class of solutions can be expressed as complex exponentials, where i is $\sqrt{-1}$, ω is the frequency, and χ is the wave number.

$$\frac{\partial \phi}{\partial t} + c\frac{\partial \phi}{\partial x} = 0 \qquad (17.2.24a)$$

$$\phi(x,t) = e^{i(\omega t - \chi x)} \qquad (17.2.24b)$$

For Equation (17.2.24), the dispersion equation is $\omega = c\chi$, but for discrete approximations of the equation and for general hyperbolic equations, the relation is $\omega = \omega\chi$. The phase velocity, c_p, and the group speed, c_g, are defined by Equation (17.2.25).

$$cp = \frac{\omega}{\chi} \qquad (17.2.25a)$$

$$c_g = \frac{\partial \omega}{\partial \chi} \qquad (17.2.25b)$$

The mesh spacing is assumed to have a constant value J, and the time step, h, is also constant. The + and - states in the previous discussions correspond to times n and $n+1$ in the dispersion analysis. An explicit linear advection method that has the form given by Equation (17.2.26) results in a complex dispersion equation, Equation (17.2.27), where Π is a complex polynomial.

$$\phi_j^{n+1} = \phi_j^n + F(c, h, J, ..., \phi_{j-1}^n, \phi_j^n \phi_{j+1}^n, ...) \qquad (17.2.26)$$

$$e^{i\omega h} = 1 + P(e^{i\chi J}) \qquad (17.2.27a)$$

$$\Pi(e^{i\chi J}) = \sum_j \beta_j e^{i\chi j J} \qquad (17.2.27b)$$

The dispersion equation has the general form given in Equation (17.2.28), where Π_r and Π_i denote the real and imaginary parts of Π, respectively.

$$\omega h = \tan^{-1}\left(\frac{\Pi_i}{1 + \Pi_r}\right) \qquad (17.2.28)$$

Recognizing that the relations in the above equations are periodic in ωh and χ^J, the normalized frequency and wave number are defined to simplify the notation.

$$\bar{\omega} = \omega h \qquad \bar{\chi} = \chi^J \qquad (17.2.29)$$

The von Neumann analysis of the SALE algorithm proceeds by first calculating the increment in the cell momentum.

$$P^n_{j+1/2} = \frac{1}{2}\left(v^n_j + v^n_{j+1}\right) \tag{17.2.30a}$$

$$P^n_{j+1/2} = \frac{1}{2}\left(1+e^{-i\bar{\chi}}\right)v^n_j \tag{17.2.30b}$$

$$\Delta P^{n+1}_{j+1/2} = P^{n+1}_{j+1/2} - P^n_{j+1/2} \tag{17.2.30c}$$

$$\Delta P^{n+1}_{j+1/2} = \frac{1}{2}\left(1+e^{-i\bar{\chi}}\right)\Pi v^n_j \tag{17.2.30d}$$

The velocity is updated from the changes in the cell momentum.

$$v^{n+1}_j = v^n_j + \frac{1}{2}\left(\Delta P^{n+1}_{j+1/2} + \Delta P^{n+1}_{j-1/2}\right) \tag{17.2.31a}$$

$$v^{n+1}_j = \frac{1}{4}\left(1+e^{i\bar{\chi}}\right)\left(1+e^{-i\bar{\chi}}\right)\Pi v^n_j \tag{17.2.31b}$$

$$v^{n+1}_j = \frac{1}{2}\left(1+\cos(\bar{\chi})\right)\Pi v^n_j \tag{17.2.31c}$$

The dispersion relation for the SALE advection algorithm is given by Equation (17.2.32).

$$\bar{\omega} = \tan^{-1}\left(\frac{\frac{1}{2}\left(1+\cos(\bar{\chi})\right)\Pi_i}{1+\frac{1}{2}\left(1+\cos(\bar{\chi})\right)\Pi_r}\right) \tag{17.2.32}$$

By comparing Equation (17.2.28) and Equation (17.2.32), the effect of the SALE momentum advection algorithm on the dispersion is to introduce a factor λ, equal to $\frac{1}{2}\left(1+\cos(\bar{\chi})\right)\Pi$, into the spatial part of the advection stencil. For small values of $\bar{\chi}$, λ is close to one, and the dispersion characteristics are not changed, but when $\bar{\chi}$ is π, the phase and group velocity go to zero and the amplification factor is one independent of the underlying advection algorithm. Not only is the wave not transported, it is not damped out. The same effect is found in two dimensions, where λ, has the form $\frac{1}{4}\left(1+\cos(\bar{\chi})+\cos(\bar{\bar{\chi}})+\cos(\bar{\chi})\cos(\bar{\bar{\chi}})\right)$.

In contrast, none of the other algorithms alter the dispersion characteristics of the underlying algorithm. Benson has demonstrated for the element-centered algorithms that the SALE inversion error and the dispersion problem are linked. Algorithms that fall into the same general class as the SALE and HIS algorithms will, therefore, not have dispersion problems [Benson 1992].

Three-Dimensional Momentum Advection Algorithms. The momentum advection algorithms discussed in the previous sections are extended to three dimensions in a straightforward manner. The staggered mesh algorithm requires the construction of a staggered mesh and the appropriate transport masses. Based on the consistency arguments, the appropriate transport masses are given by Equation (17.2.33).

$$\tilde{f}_{j+1/2,k,l} = \frac{1}{8}\sum_{J=j}^{j+1}\sum_{K=k-\frac{1}{2}}^{k+\frac{1}{2}}\sum_{L=l-\frac{1}{2}}^{l+\frac{1}{2}} f_{J,K,L} \tag{17.2.33}$$

The SALE advection algorithm calculates the average momentum of the element from the four velocities at the nodes and distributes 1/8 of the change in momentum to each node.

$$p_{j+\frac{1}{2},k+\frac{1}{2},l+\frac{1}{2}} = \frac{1}{8}\rho_{j+\frac{1}{2},k+\frac{1}{2},l+\frac{1}{2}}\sum_{J=j}^{j+1}\sum_{K=k}^{k+1}\sum_{L=l}^{l+1} v_{JKL} \tag{17.2.34a}$$

$$p_{j+\frac{1}{2},k+\frac{1}{2},l+\frac{1}{2}} = \frac{1}{8}M_{j+\frac{1}{2},k+\frac{1}{2},l+\frac{1}{2}}\sum_{J=j}^{j+1}\sum_{K=k}^{k+1}\sum_{L=l}^{l+1} v_{JKL} \tag{17.2.34b}$$

$$v^{+}_{j,k,l} = \frac{1}{M^{+}_{j,k,l}}\left(M^{-}_{j,k,l}v^{-}_{j,k,l} + \frac{1}{8}\sum_{J=j-\frac{1}{2}}^{j+\frac{1}{2}}\sum_{K=k-\frac{1}{2}}^{k+\frac{1}{2}}\sum_{L=l-\frac{1}{2}}^{l+\frac{1}{2}} \Delta P_{J,K,L} \right) \tag{17.2.34c}$$

The HIS algorithm is also readily extended to three dimensions. The variable definitions are given in Equation (17.2.35) and Equation (17.2.36), where the subscript A refers to the local numbering of the nodes in the element. In an unstructured mesh, the relative orientation of the nodal numbering within the elements may change. The subscript A is always with reference to the numbering in element j,k,l. The subscript $\tilde{A}$ is the local node number in an adjacent element that refers to the same global node number as A.

$$\Psi_{A,j+\frac{1}{2},k+\frac{1}{2},l+\frac{1}{2}} = v_{A,j+\frac{1}{2},k+\frac{1}{2},l+\frac{1}{2}} \tag{17.2.35}$$

$$v^{+}_{j,k,l} = \frac{1}{M^{+}_{j,k,l}}\sum_{J=j-\frac{1}{2}}^{j+\frac{1}{2}}\sum_{K=k-\frac{1}{2}}^{k+\frac{1}{2}}\sum_{L=l-\frac{1}{2}}^{l+\frac{1}{2}} M_{J,K,L}\Psi^{+}_{\tilde{A},J,K,L} \tag{17.2.36}$$

17.3 The Manual Rezone

The central limitation to the simplified ALE formulation is that the topology of the mesh is fixed. For a problem involving large deformations, a mesh that works well at early times may not work at late times regardless of how the mesh is distributed over the material domain. To circumvent this difficulty, a manual rezoning capability has been implemented in LS-DYNA. The general procedure is to 1) interrupt the calculation when the mesh is no longer acceptable, 2) generate a new mesh with INGRID by using the current material boundaries from LS-DYNA (the

topologies of the new and old mesh are unrelated), 3) remap the solution from the old mesh to the new mesh, and 4) restart the calculation.

This chapter will concentrate on the remapping algorithm since the mesh generation capability is documented in the INGRID manual [Stillman and Hallquist 1992]. The remapping algorithm first constructs an interpolation function on the original mesh by using a least squares procedure, and then interpolates for the solution values on the new mesh.

The one point quadrature used in LS-DYNA implies a piecewise constant distribution of the solution variables within the elements. A piecewise constant distribution is not acceptable for a rezoner since it implies that for even moderately large changes in the locations of the nodes (say, displacements on the order of fifty percent of the elements characteristic lengths) that there will be no changes in the values of the element-centered solution variables. A least squares algorithm is used to generate values for the solution variables at the nodes from the element-centered values. The values of the solution variables can then be interpolated from the nodal values, ϕ_A, using the standard trilinear shape functions anywhere within the mesh.

$$\phi(\xi,\eta,\zeta)=\phi_A N_A(\xi,\eta,\zeta) \tag{17.3.1}$$

The objective function for minimization, J, is defined material by material, and each material is remapped independently.

$$J=\frac{1}{2}\int_V(\phi_A N_A-\phi)^2\,dV \tag{17.3.2}$$

The objective function is minimized by setting the derivatives of J with respect to ϕ_A equal to zero.

$$\frac{\partial J}{\partial \phi_A}=\int_V(\phi_B N_B-\phi)N_A dV=0 \tag{17.3.3}$$

The least square values of ϕ_A are calculated by solving the system of linear equations, Equation (17.3.4).

$$M_{AB}\phi_B=\int_V N_A\phi dV \tag{17.3.4a}$$

$$M_{AB}=\int_V N_A N_B dV \tag{17.3.4b}$$

The "mass matrix," M_{AB}, is lumped to give a diagonal matrix. This eliminates the spurious oscillations that occur in a least squares fit around the discontinuities in the solution (e.g., shock waves) and facilitates an explicit solution for ϕ_A. The integral on the right hand side of Equation (17.3.4a) is evaluated using one point integration. By introducing these simplifications,

Equation (17.3.4) is reduced to Equation (17.3.5), where the summation over a is restricted to the elements containing node A.

$$\phi_A = \frac{\sum_\alpha \phi_\alpha V_\alpha}{\sum_\alpha V_\alpha} \tag{17.3.5}$$

The value of ϕ_α is the mean value of ϕ in element a. From this definition, the value of ϕ_α is calculated using Equation (17.3.6).

$$\phi_\alpha = \frac{1}{V_\alpha} \int_{V_\alpha} \phi_A N_A dV \tag{17.3.6}$$

The integrand in Equation (17.3.6) is defined on the old mesh, so that Equation (17.3.6) is actually performed on the region of the old mesh that overlaps element α in the new mesh, where the superscript "*" refers to elements on the old mesh.

$$\phi_\alpha = \frac{1}{V_\alpha} \Sigma_\beta \int_{V_\alpha \cap V_\beta^*} \phi_A N_A dV^* \tag{17.3.7}$$

One point integration is currently used to evaluate Equation (17.3.7), although it would be a trivial matter to add higher order integration. By introducing this simplification, Equation (17.3.7) reduces to interpolating the value of ϕ_α from the least squares fit on the old mesh.

$$\phi_a = \phi_A N_A(\xi^*,\eta^*,\zeta^*) \tag{17.3.8}$$

The isoparametric coordinates in the old mesh that correspond to the spatial location of the new element centroid must be calculated for Equation (17.3.8). The algorithm that is described here is from Shapiro [1990], who references [Thompson and Maffeo 1985, Maffeo 1984, Maffeo 1985] as the motivations for his current strategy, and we follow his notation. The algorithm uses a "coarse filter" and a "fine filter" to make the search for the correct element in the old mesh efficient.

The coarse filter calculates the minimum and maximum coordinates of each element in the old mesh. If the new element centroid, (x_s, y_s, z_s), lies outside of the box defined by the maximum and minimum values of an old element, then the old element does not contain the new element centroid.

Several elements may pass the coarse filter but only one of them contains the new centroid. The fine filter is used to determine if an element actually contains the new centroid. The fine filter algorithm will be explained in detail for the two-dimensional case since it easier to visualize than the three-dimensional case, but the final equations will be given for the three-dimensional case.

The two edges adjacent to each node in Figure 17.2 (taken from [Shapiro 1990]) define four skew coordinate systems. If the coordinates for the new centroid are positive for each

coordinate system, then the new centroid is located within the old element. Because of the overlap of the four positive quarter spaces defined by the skew coordinate systems, only two coordinate systems actually have to be checked. Using the first and third coordinate systems, the coordinates, a_i, are the solution of Equation (17.3.9).

$$V_s = V_1 + a_1 V_{12} + a_2 V_{14} \tag{17.3.9a}$$

$$V_s = V_3 + a_3 V_{32} + a_4 V_{34} \tag{17.3.9b}$$

Two sets of linear equations are generated for the α_i by expanding the vector equations.

$$\begin{bmatrix} x_2 - x_1 & x_4 - x_1 \\ y_2 - y_1 & y_4 - y_1 \end{bmatrix} \begin{Bmatrix} \alpha_1 \\ \alpha_2 \end{Bmatrix} = \begin{Bmatrix} x_s - x_1 \\ y_s - y_1 \end{Bmatrix} \tag{17.3.10a}$$

$$\begin{bmatrix} x_2 - x_3 & x_4 - x_3 \\ y_2 - y_3 & y_4 - y_3 \end{bmatrix} \begin{Bmatrix} \alpha_3 \\ \alpha_4 \end{Bmatrix} = \begin{Bmatrix} x_s - x_3 \\ y_s - y_3 \end{Bmatrix} \tag{17.3.10b}$$

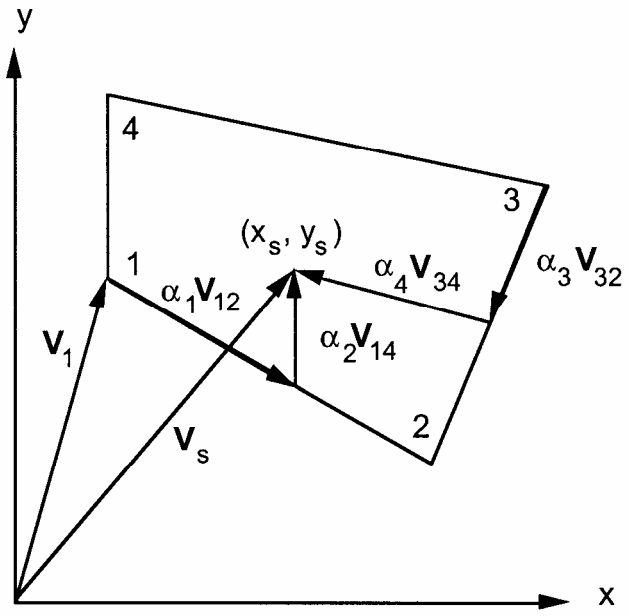


Figure 17.2. Skew Coordinate System

The generalization of Equation (17.3.9) to three dimensions is given by Equation (17.3.11), and it requires the solution of four sets of three equations. The numbering convention for the nodes in Equation (17.3.11) follows the standard numbering scheme used in LS-DYNA for eight node solid elements.

$$V_s = V_1 + a_1 V_{12} + a_2 V_{14} + a_3 V_{15} \tag{17.3.11a}$$

$$V_s = V_3 + a_4 V_{37} + a_5 V_{34} + a_6 V_{32} \tag{17.3.11b}$$

$$V_s = V_6 + a_7 V_{62} + a_8 V_{65} + a_9 V_{67} \tag{17.3.11c}$$

$$V_s = V_8 + a_{10} V_{85} + a_{11} V_{84} + a_{12} V_{87} \tag{17.3.11d}$$

The fine filter sometimes fails to locate the correct element when the mesh is distorted. When this occurs, the element that is closest to the new centroid is located by finding the element for which the sum of the distances between the new centroid and the nodes of the element is a minimum.

Once the correct element is found, the isoparametric coordinates are calculated using the Newton-Raphson method, which usually converges in three or four iterations.

$$\begin{bmatrix} x_A \dfrac{\partial N_A}{\partial \xi} & x_A \dfrac{\partial N_A}{\partial \eta} & x_A \dfrac{\partial N_A}{\partial \zeta} \\ y_A \dfrac{\partial N_A}{\partial \xi} & y_A \dfrac{\partial N_A}{\partial \eta} & y_A \dfrac{\partial N_A}{\partial \zeta} \\ z_A \dfrac{\partial N_A}{\partial \xi} & z_A \dfrac{\partial N_A}{\partial \eta} & z_A \dfrac{\partial N_A}{\partial \zeta} \end{bmatrix} \begin{Bmatrix} \Delta\xi \\ \Delta\eta \\ \Delta\zeta \end{Bmatrix} = \begin{Bmatrix} x_s - x_A N_A \\ y_s - y_A N_A \\ z_s - z_A N_A \end{Bmatrix} \tag{17.3.12}$$

$$\begin{aligned} \xi^{i+1} &= \xi^i + \Delta\xi \\ \eta^{i+1} &= \eta^i + \Delta\eta \\ \zeta^{i+1} &= \zeta^i + \Delta\zeta \end{aligned} \tag{17.3.13}$$

18. STRESS UPDATE OVERVIEW

18.1 Jaumann Stress Rate

Stresses for material which exhibit elastic-plastic and soil-like behavior (hypoelastic) are integrated incrementally in time:

$$\sigma_{ij}(t+dt) = \sigma_{ij}(t) + \dot{\sigma}_{ij}dt \tag{18.1}$$

Here, and in equations which follow, we neglect the contribution of the bulk viscosity to the stress tensor. In Equation (17.1), the dot denotes the material time derivative given by

$$\dot{\sigma}_{ij} = \sigma_{ij}^{\nabla} + \sigma_{ik}\omega_{kj} + \sigma_{jk}\omega_{ki} \tag{18.2}$$

in which

$$\omega_{ij} = \frac{1}{2}\left(\frac{\partial v_i}{\partial x_j} - \frac{\partial v_j}{\partial x_i}\right) \tag{18.3}$$

is the spin tensor and

$$\sigma_{ij}^{\nabla} = C_{ijkl}\dot{\varepsilon}_{kl} \tag{18.4}$$

is the Jaumann (co-rotational) stress rate. In Equation 18.4, C_{ijkl} is the stress dependent constitutive matrix, v_i, is the velocity vector, and $\dot{\varepsilon}_{ij}$ is the strain rate tensor:

$$\omega_{ij} = \frac{1}{2}\left(\frac{\partial v_i}{\partial x_j} - \frac{\partial v_j}{\partial x_i}\right) \tag{18.5}$$

In the implementation of Equation (18.1) we first perform the stress rotation, Equation (18.2), and then call a constitutive subroutine to add the incremental stress components σ_{ij}^{∇}. This may be written as

$$\sigma_{ij}^{n+1} = \sigma_{ij}^{n} + r_{ij}^{n} + \sigma_{ij}^{\nabla n+1/2}\Delta t^{n+1/2} \tag{18.6}$$

where

$$\begin{aligned} \sigma_{ij}^{\nabla n+1/2}\Delta t^{n+1/2} &= C_{ijkl}\Delta\varepsilon_{kl}^{n+1/2} \\ \Delta\varepsilon_{ij}^{n+1/2} &= \dot{\varepsilon}_{ij}^{n+1/2}\Delta t^{n+1/2} \end{aligned} \tag{18.7}$$

and r_{ij}^n gives the rotation of the stress at time t^n to the configuration at t^{n+1}

$$r_{ij}^n = \left(\sigma_{ip}^n \omega_{pj}^{n+1/2} + \sigma_{jp}^n \omega_{pi}^{n+1/2} \right) \Delta t^{n+1/2} \tag{18.8}$$

In the implicit NIKE2D/3D [Hallquist 1981b] codes, which are used for low frequency structural response, we do a half-step rotation, apply the constitutive law, and complete the second half-step rotation on the modified stress. This approach has also been adopted for some element formulations in LS-DYNA when the invariant stress update is active. An exact or second order accurate rotation is performed rather than the approximate one represented by Equation (18.3), which is valid only for small incremental rotations. A typical implicit time step is usually 100 to 1000 or more times larger than the explicit time step; consequently, the direct use of Equation (18.7) could lead to very significant errors.

18.2 Jaumann Stress Rate Used With Equations of State

If pressure is defined by an equation of state as a function of relative volume, V, and energy, E, or temperature, T,

$$p = p(V, E) = p(V, T) \tag{18.9}$$

we update the deviatoric stress components

$$s_{ij}^{n+1} = \sigma_{ij}^n + r_{ij}^n + p^n \delta_{ij} + C_{ijkl} \dot{\varepsilon}_{kl}'^{n+1/2} \Delta t^{n+1/2} \tag{18.10}$$

where $\dot{\varepsilon}_{ij}'^{n+1/2}$ is the deviatoric strain rate tensor:

$$\dot{\varepsilon}_{ij}'^{n+1/2} = \dot{\varepsilon}_{ij} - \frac{1}{3} \dot{\varepsilon}_{kk} \delta \tag{18.11}$$

Before the equation of state, Equation (18.9), is evaluated, we compute the bulk viscosity, q, and update the total internal energy e of the element being processed to a trial value e^*:

$$e^{*n+1} = e^n - \frac{1}{2} \Delta v \left(p^n + q^{n-1/2} + q^{n+1/2} \right) + v^{n+1/2} s_{ij}^{n+1/2} \Delta \varepsilon_{ij}^{n+1/2} \tag{18.12}$$

where v is the element volume and

$$\begin{aligned} \Delta v &= v^{n+1} - v^n, v^{n+1/2} = \frac{1}{2} \left(v^n + v^{n+1} \right) \\ s_{ij}^{n+1/2} &= \frac{1}{2} \left(s_{ij}^n + s_{ij}^{n+1} \right) \end{aligned} \tag{18.13}$$

The time-centering of the viscosity is explained by Noh [1976].

Assume we have an equation of state that is linear in internal energy of the form

$$p^{n+1} = A^{n+1} + B^{n+1}E^{n+1} \tag{18.14}$$

where

$$E^{n+1} = \frac{e^{n+1}}{v_0} \tag{18.15}$$

and v_0 is the initial volume of the element. Noting that

$$e^{n+1} = e^{*n+1} - \frac{1}{2}\Delta v p^{n+1} \tag{18.16}$$

pressure can be evaluated exactly by solving the implicit form

$$p^{n+1} = \frac{\left(A^{n+1} + B^{n+1}E^{*n+1}\right)}{\left(1 + \frac{1}{2}B^{n+1}\frac{\Delta v}{v_0}\right)} \tag{18.17}$$

and the internal energy can be updated in Equation (18.16). If the equation of state is not linear in internal energy, a one-step iteration is used to approximate the pressure

$$p^{*n+1} = p(V^{n+1}, E^{*n+1}) \tag{18.18}$$

Internal energy is updated to $n+1$ using p^{*n+1} in Equation (18.16) and the final pressure is then computed:

$$p^{n+1} = p(V^{n+1}, E^{n+1}) \tag{18.19}$$

This is also the iteration procedure used in KOVEC [Woodruff 1973]. All the equations of state in LS-DYNA are linear in energy except the ratio of polynomials.

18.3 Green-Naghdi Stress Rate

The Green-Naghdi rate is defined as

$$\sigma_{ij}^{\nabla} = \dot{\sigma}_{ij} + \sigma_{ik}\Omega_{kj} + \sigma_{jk}\Omega_{ki} = R_{ik}R_{jl}\dot{\tau}_{kl} \tag{18.20}$$

where Ω_{ij} is defined as

$$\Omega_{ij} = \dot{R}_{ik}R_{ik} \tag{18.21}$$

and R is found by application of the polar decomposition theorem

$$F_{ij} = R_{ik}U_{kj} = V_{ik}R_{kj} \tag{18.22}$$

F_{ij} is the deformation gradient matrix and U_{ij} and V_{ij} are the positive definite right and left stretch tensors:

$$F_{ij} = \frac{\partial x_i}{\partial X_j} \tag{18.23}$$

Stresses are updated for all materials by adding to the rotated Cauchy stress at time n.

$$\tau_{ij}^n = R_{ki}^n R_{lj}^n \sigma_{kl}^n \tag{18.24}$$

the stress increment obtained by an evaluation of the constitutive equations,

$$\Delta\tau_{ij}^{n+1/2} = C_{ijkl}\Delta d_{kl}^{n+1/2} \tag{18.25}$$

where

$$\Delta d_{ij}^{n+1/2} = R_{ki}^{n+1/2} R_{lj}^{n+1/2} \Delta\varepsilon_{kl}^{n+1/2} \tag{18.26}$$

C_{ijkl} = constitutive matrix

$\Delta\varepsilon_{kl}$ = increment in strain

and to obtain the rotated Cauchy stress at $n+1$, i.e.,

$$\tau_{ij}^{n+1} = \tau_{ij}^n + \Delta\tau_{ij}^{n+1/2} \tag{18.27}$$

The desired Cauchy stress at $n+1$ can now be found

$$\sigma_{ij}^{n+1} = R_{ik}^{n+1} R_{jl}^{n+1} \tau_{kl}^{n+1} \tag{18.28}$$

Because we evaluate our constitutive models in the rotated configuration, we avoid the need to transform history variables such as the back stress that arises in kinematic hardening.

In the computation of R, Taylor and Flanagan [1989] did an incremental update in contrast with the direct polar decomposition used in the NIKE3D code. Following their notation the algorithm is given by.

$$z_i = e_{ijk} V_{jm} \dot{\varepsilon}_{mk}$$

$$\varpi_i = e_{ijk}\omega_{jk} - 2\left[V_{ij} - \delta_{ij}V_{kk}\right]^{-1} z_j$$

$$\Omega_{ij} = \frac{1}{2} e_{ijk}\varpi_k$$

$$\left(\delta_{ik} - \tfrac{\Delta t}{2}\Omega_{ik}\right)R_{kj}^{n+1} = \left(\delta_{ik} + \tfrac{\Delta t}{2}\Omega_{ik}\right)R_{kj}^{n} \qquad (18.29)$$

$$\dot{V}_{ij} = (\dot{\varepsilon}_{ik} + \omega_{ik})V_{kj} - V_{ik}\Omega_{kj}$$

$$V_{ij}^{n+1} = V_{ij}^{n} + \Delta t\dot{V}_{ij}$$

We have adopted the PRONTO3D approach in LS-DYNA due to numerical difficulties with the polar decomposition in NIKE3D. We believe the PRONTO3D approach is reliable. Several disadvantages of the PRONTO3D approach include 300+ operations (at least that is the number we got), the requirement of 15 additional variables per integration point, and if we rezone the material in the future the initial geometry will need to be remapped and the 15 variables initialized.

18.4 Elastoplastic Materials

At low stress levels in elastoplastic materials the stresses, σ_{ij}, depends only on the state of strain; however, above a certain stress level, called the yield stress, $\sigma_y(a_i)$, nonrecoverable plastic deformations are obtained. The yield stress changes with increasing plastic deformations, which are measured by internal variables, a_i.

In the uniaxial tension test, a curve like that in Figure 18.1 is generated where logrithmic uniaxial strain is plotted against the uniaxial true stress which is defined as the applied load P divided by the actual cross-sectional area, A.

For the simple von Mises plasticity models the yield stress is pressure independent and the yield surface is a cylinder in principal stress space as shown in Figure 18.2. With isotropic hardening the diameter of the cylinder grows but the shape remains circular. In kinematic hardening the diameter may remain constant but will translate in the plane as a function of the plastic strain tensor, See Figure 18.3.

The equation describing the pressure independent yield surface, F, is a function of the deviatoric stress tensor, s_{ij}, of a form given in Equation (18.30).

$$F(s_{ij}, a_i) = f(s_{ij}) - \sigma_y(a_i) = 0 \qquad (18.30)$$

$f(s_{ij})$ = determines the shape,

$\sigma_y(a_i)$ = determines the translation and size.

The existence of a potential function, g, called the plastic potential, is assumed

$$g = g(s_{ij}) \qquad (18.31)$$

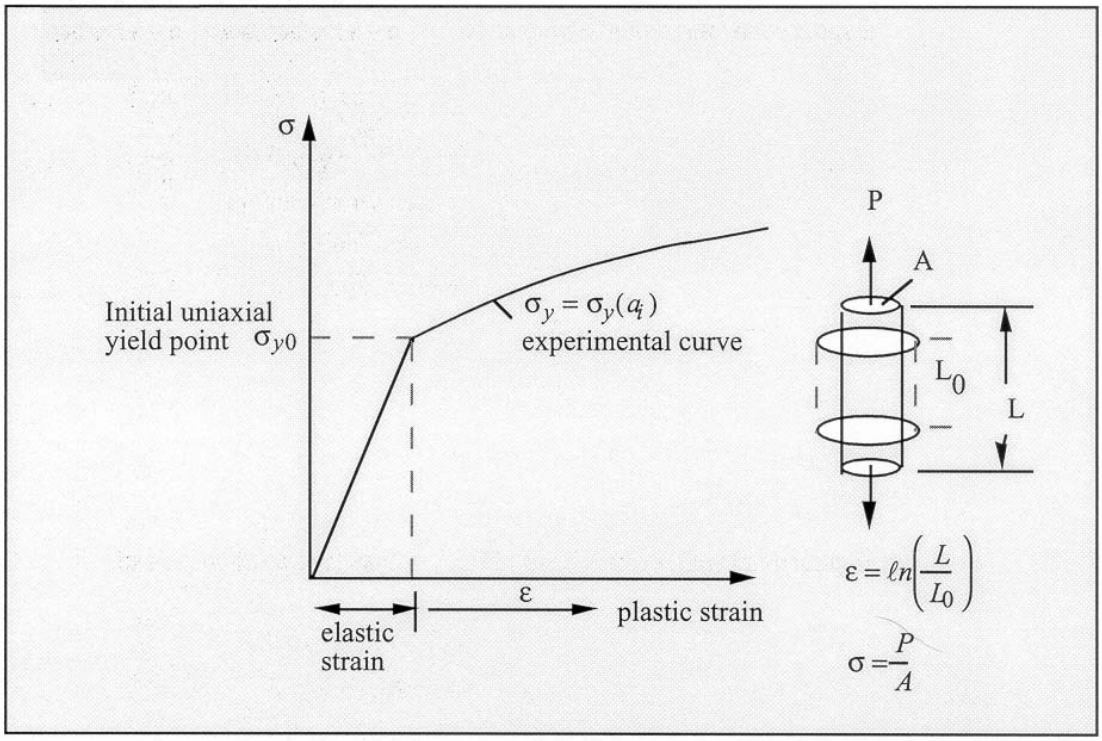


Figure 18.1. The uniaxial tension test demonstrates plastic behavior.

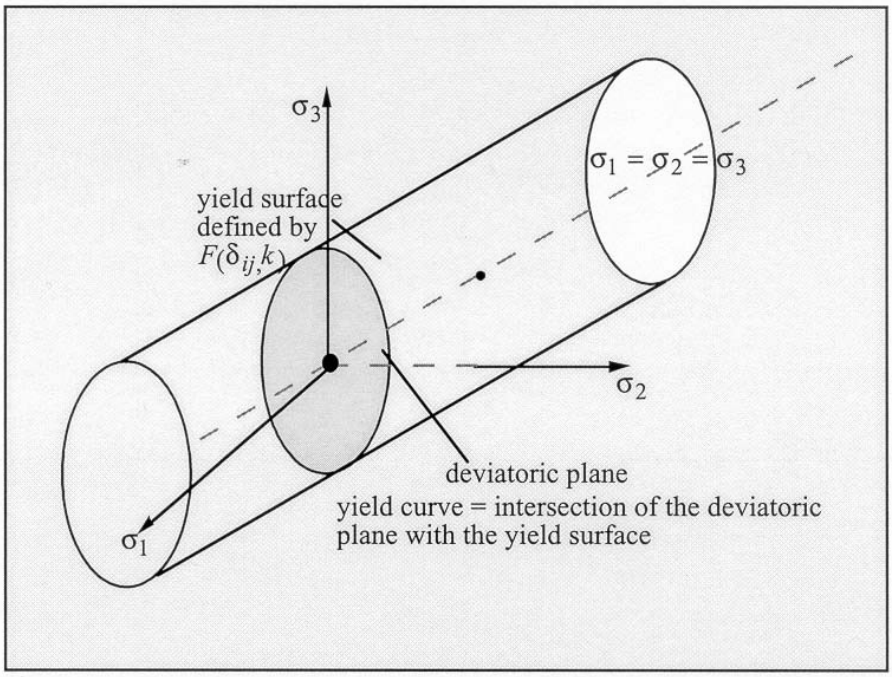


Figure 18.2. The yield surface in principal stress space in pressure independent.

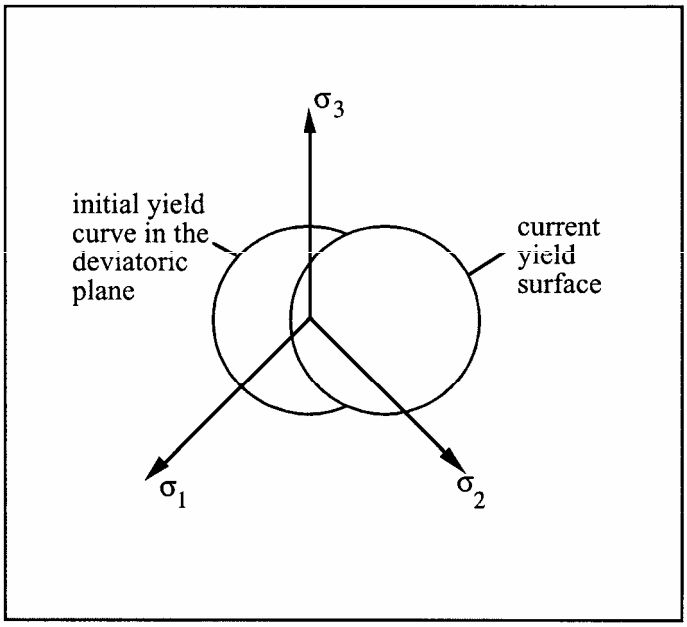


Figure 18.3. With kinematic hardening the yield surface may shift as a function of plastic strain.

Stability and uniqueness require that:

$$d\varepsilon_{ij}^{p} = \lambda \frac{\partial g}{\partial s_{ij}} \tag{18.32}$$

where λ is a proportionality constant.

As depicted in Figure 18.4 the plastic strain increments $d\varepsilon_{ij}^{p}$ are normal to the plastic potential function. This is the normality rule of plasticity.

The plastic potential g is identical with the yield condition $F(s_{ij})$

$$g \equiv F \tag{18.33}$$

Hence:

$$d\varepsilon_{ij}^{p} = \lambda \frac{\partial g}{\partial s_{ij}} \quad \varnothing \quad d\varepsilon_{ij}^{p} = \lambda \cdot \frac{\partial f}{\partial s_{ij}} = \lambda \text{grad}\, f \tag{18.34}$$

and the stress increments ds_{ij} are normal to the plastic flow $\frac{\partial g}{\partial s_{ij}}$.

Post -yielding behavior from uniaxial tension tests typically show the following behaviors illustrated in Figure 18.5:

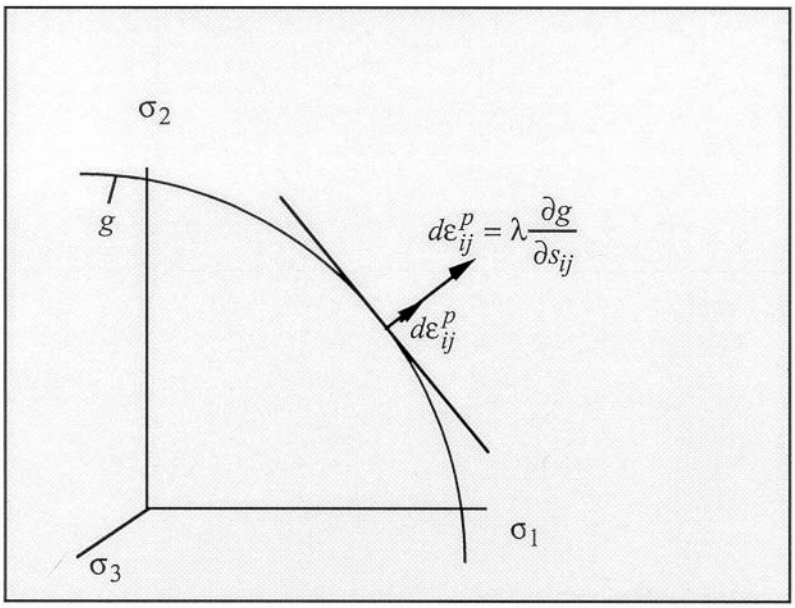


Figure 18.4. The plastic strain is normal to the yield surface.

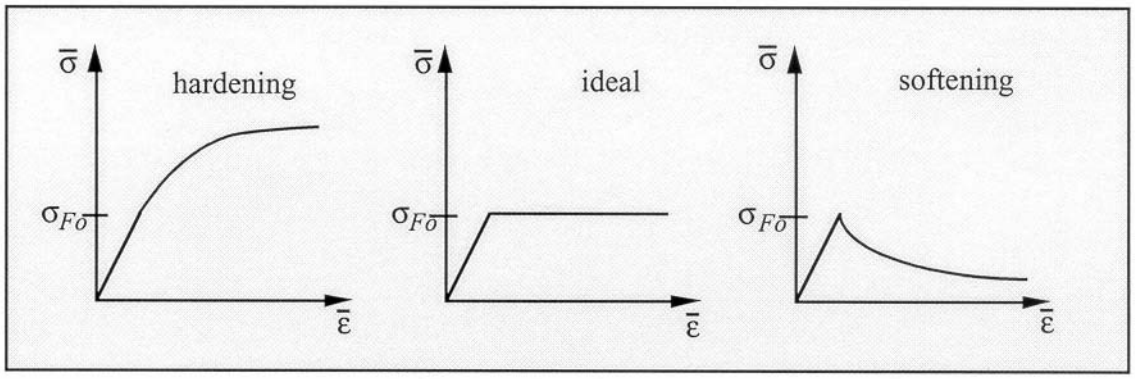


Figure 18.5. Hardening, ideal, and softening plasticity models.

The behavior of these hardening laws are characterized in Table 18.1. below. Although LS-DYNA permits softening to be defined and used, such softening behavior will result in strain localization and nonconvergence with mesh refinement.

18.5 Hyperelastic Materials

Stresses for elastic and hyperelastic materials are path independent; consequently, the stress update is not computed incrementally. The methods described here are well known and the reader is referred to Green and Adkins [1970] and Ogden [1984] for more details.

A retangular cartesian coordinate system is used so that the covariant and contravariant metric tensors in the reference (undeformed) and deformed configuration are:

$$
\begin{aligned}
g_{ij} &= g^{ij} = \delta_{ij} \\
G_{ij} &= \frac{\partial x_k}{\partial X_i}\frac{\partial x_k}{\partial X_j} \\
G^{ij} &= \frac{\partial X_i}{\partial x_k}\frac{\partial X_j}{\partial x_k}
\end{aligned}
\tag{18.35}
$$

The Green-St. Venant strain tensor and the principal strain invariants are defined as

$$
\begin{aligned}
\gamma_{ij} &= \tfrac{1}{2}\left(G_{ij} - \delta_{ij}\right) \\
I_1 &= \delta^{ij} G_{ij} \\
I_2 &= \tfrac{1}{2}\left(\delta^{ir}\delta^{js} G_{ri} G_{sj} - \delta^{ir}\delta^{js} G_{ij} G_{rs}\right) \\
I_3 &= \det G_{ij}
\end{aligned}
\tag{18.36}
$$

	HARDENING	IDEAL	SOFTENING
BEHAVIOR	$\sigma_y(a_i)$ is monotonic increasing	$\sigma_y(a_i)$ is constant	$\sigma_y(a_i)$ is monotonic decreasing
STABILITY	yes	yes	No
UNIQUENESS	yes	yes	No
APPLICATIONS	metals, concrete, rock with small deformations	crude idealization for steel, plastics, etc.	dense sand, concrete with large deformations

Table 18.1. Plastic hardening, ideal plasticity, and softening.

For a compressible elastic material the existence of a strain energy functional, W, is assumed

$$
W = W(I_1, I_2, I_3) \tag{18.37}
$$

which defines the energy per unit undeformed volume. The stress measured in the deformed configuration is given as [Green and Adkins, 1970]:

$$
s^{ij} = \Phi g^{ij} + \Psi B^{ij} + pG^{ij} \tag{18.38}
$$

where

$$\Phi = \frac{2}{\sqrt{I_3}} \frac{\partial W}{\partial I_1}$$

$$\Psi = \frac{2}{\sqrt{I_3}} \frac{\partial W}{\partial I_2} \tag{18.39a}$$

$$p = 2\sqrt{I_3} \frac{\partial W}{\partial I_3}$$

$$B^{ij} = I_1 \delta^{ij} - \delta^{ir} \delta^{js} G_{rs} \tag{18.39b}$$

This stress is related to the second Piola-Kirchhoff stress tensor:

$$S^{ij} = s^{ij} \sqrt{I_3} \tag{18.40}$$

Second Piola-Kirchhoff stresses are transformed to physical (Cauchy) stresses according to the relationship:

$$\sigma_{ij} = \frac{\rho}{\rho_0} \frac{\partial x_i}{\partial X_k} \frac{\partial x_j}{\partial X_l} S_{kl} \tag{18.41}$$

18.6 Layered Composites

The composite models for shell elements in LS-DYNA include models for elastic behavior and inelastic behavior. The approach used here for updating the stresses also applies to the airbag fabric model.

To allow for an arbitrary orientation of the shell elements within the finite element mesh, each ply in the composite has a unique orientation angle, β, which measures the offset from some reference in the element. Each integration point through the shell thickness, typically though not limited to one point per ply, requires the definition of β at that point. The reference is determined by the angle Ψ, which can be defined for each element on the element card, and is measured from the 1-2 element side. Figures 18.6 and 18.7 depict these angles.

We update the stresses in the shell in the local shell coordinate system which is defined by the 1-2 element side and the cross product of the diagonals. Thus to transform the stress tensor into local system determined by the fiber directions entails a transformation that takes place in the plane of the shell.

In the implementation of the material model we first transform the Cauchy stress and velocity strain tensor d_{ij} into the coordinate system of the material denoted by the subscript L

$$\vec{\sigma}_L = q^t \sigma q$$

$$q_L = q^t d q \tag{18.42}$$

$$\sigma_L = \begin{bmatrix} \sigma_{11}\ \sigma_{12}\sigma_{13} \\ \sigma_{21}\ \sigma_{22}\sigma_{23} \\ \sigma_{32}\ \sigma_{32}\sigma_{33} \end{bmatrix} \varepsilon_L = \begin{bmatrix} d_{11}\ d_{12}\ d_{13} \\ d_{21}\ d_{22}\ d_{23} \\ d_{32}\ d_{32}\ d_{33} \end{bmatrix}$$

The Arabic subscripts on the stress and strain (σ and ε) are used to indicate the principal material directions where 1 indicates the fiber direction and 2 indicates the transverse fiber direction (in the plane). The orthogonal 3×3 transformation matrix is given by

$$q = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{18.43}$$

In shell theory we assume a plane stress condition, i.e., that the normal stress, σ_{33}, to the mid-surface is zero. We can now incrementally update the stress state in the material coordinates

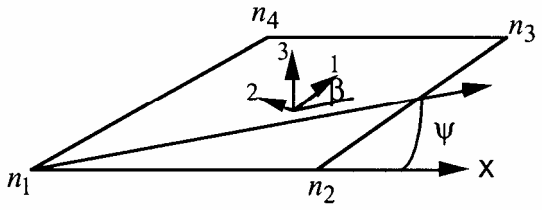


Figure 18.6. Orientation of material directions relative to the 1-2 side.

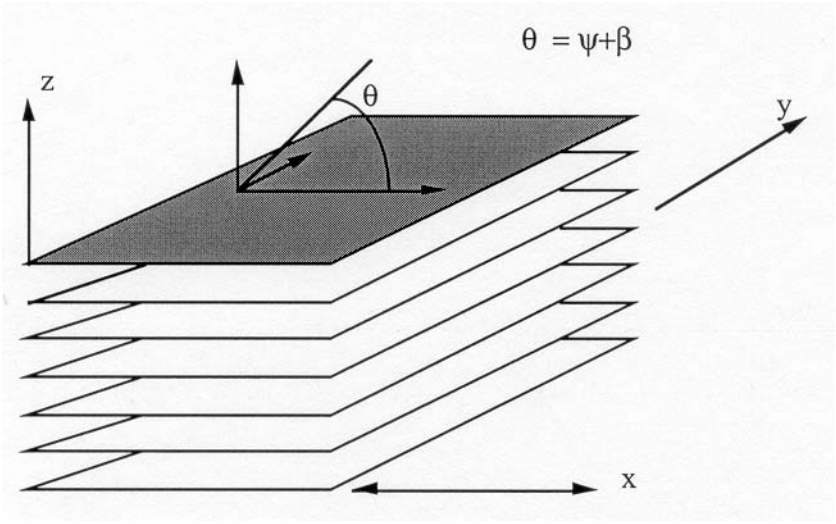


Figure 18.7. A multi-layer laminate can be defined. The angle β_i is defined for the ith lamina.

$$\sigma_L^{n+1} = \sigma_L^n + \Delta\sigma_L^{n+1/2} \tag{18.44}$$

where for an elastic material

$$\Delta\sigma_L^{n+1/2} = \begin{bmatrix} \Delta\sigma_{11} \\ \Delta\sigma_{22} \\ \Delta\sigma_{12} \\ \Delta\sigma_{23} \\ \Delta\sigma_{31} \end{bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} & 0 & 0 & 0 \\ Q_{12} & Q_{22} & 0 & 0 & 0 \\ 0 & 0 & Q_{44} & 0 & 0 \\ 0 & 0 & 0 & Q_{55} & 0 \\ 0 & 0 & 0 & 0 & Q_{66} \end{bmatrix} \begin{bmatrix} d_{11} \\ d_{22} \\ d_{12} \\ d_{23} \\ d_{31} \end{bmatrix}_L \Delta t \tag{18.45}$$

The terms Q_{ij} are referred to as reduced components of the lamina and are defined as

$$Q_{11} = \frac{E_{11}}{1-\nu_{12}\nu_{21}}$$

$$Q_{22} = \frac{E_{22}}{1-\nu_{12}\nu_{21}}$$

$$Q_{12} = \frac{\nu_{12}E_{11}}{1-\nu_{12}\nu_{21}}$$

$$Q_{44} = G_{12} \tag{18.46}$$

$$Q_{55} = G_{23}$$

$$Q_{66} = G_{31}$$

Because of the symmetry properties,

$$\nu_{ji} = \nu_{ij} \frac{E_{jj}}{E_{ii}} \tag{18.47}$$

where ν_{ij} is Poisson's ratio for the transverse strain in jth direction for the material undergoing stress in the ith-direction, E_{ij} are the Young's modulii in the ith direction, and G_{ij} are the shear modulii.

After completion of the stress update we transform the stresses back into the local shell coordinate system.

$$\sigma = q\sigma_L q^t \tag{18.48}$$

18.7 Constraints on Orthotropic Elastic Constants

The inverse of the constitutive matrix C_l is generally defined in terms of the local material axes for an orthotropic material is given in terms of the nine independent elastic constants as

$$C_l^{-1} = \begin{bmatrix} \frac{1}{E_{11}} & -\frac{\upsilon_{21}}{E_{22}} & -\frac{\upsilon_{31}}{E_{33}} & 0 & 0 & 0 \\ -\frac{\upsilon_{12}}{E_{11}} & \frac{1}{E_{22}} & -\frac{\upsilon_{32}}{E_{33}} & 0 & 0 & 0 \\ -\frac{\upsilon_{13}}{E_{11}} & -\frac{\upsilon_{23}}{E_{22}} & \frac{1}{E_{33}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{12}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{23}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{31}} \end{bmatrix} \tag{18.49}$$

As discussed by Jones [1975], the constants for a thermodynamically stable material must satisfy the following inequalities:

$$E_1, E_2, E_3, G_{12}, G_{23}, G_{31} > 0 \tag{18.50a}$$

$$C_{11,}\ C_{22}, C_{33}, C_{44}, C_{55}, C_{66} > 0 \tag{18.50b}$$

$$(1-\nu_{23}\nu_{32}),(1-\nu_{13}\nu_{31}),(1-\nu_{12}\nu_{21}) > 0 \tag{18.50c}$$

$$1-\nu_{12}\nu_{21}-\nu_{23}\nu_{32}-\nu_{31}\nu_{13}-2\nu_{21}\nu_{32}\nu_{13} > 0 \tag{18.50d}$$

Using Equation (18.47) and (18.50b) leads to:

$$|\nu_{21}| < \left(\frac{E_{22}}{E_{11}}\right)^{1/2} \quad |\nu_{12}| < \left(\frac{E_{11}}{E_{22}}\right)^{1/2} \tag{18.51}$$

$$|\nu_{32}| < \left(\frac{E_{33}}{E_{22}}\right)^{1/2} \quad |\nu_{23}| < \left(\frac{E_{22}}{E_{33}}\right)^{1/2}$$

$$|\nu_{13}| < \left(\frac{E_{11}}{E_{33}}\right)^{1/2} \quad |\nu_{31}| < \left(\frac{E_{33}}{E_{11}}\right)^{1/2}$$

18.8 Local Material Coordinate Systems in Solid Elements

In solid elements there is a number of different ways of defining a local coordinate system. Perhaps the most general is by defining a triad for each element that is oriented in the local material directions, See Figure 18.8. In this approach two vectors a and d are defined. The local c direction is found from the cross product, $c = a \times d$, the local b direction is the cross product $b = c \times a$. This triad is stored as history data at each integration point.

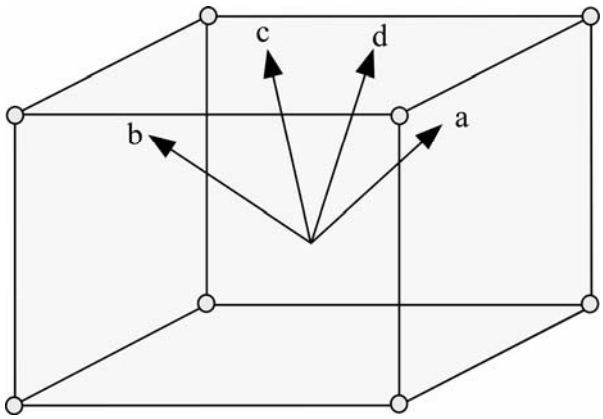


Figure 18.8. Local material directions are defined by a triad which can be input for each solid element.

The biggest concern when dealing with local material directions is that the results are not invariant with element numbering since the orientation of the local triad is fixed with respect to the base of the brick element, nodes 1-4, in Figure 18.9. For Hyperelastic materials where the stress tensor is computed in the initial configuration, this is not a problem, but for materials like the honeycomb foams, the local directions can change due to element distortion causing relative movement of nodes 1-4. In honeycomb foams we assume that the material directions are orthogonal in the deformed configuration since the stress update is performed in the deformed configuration.

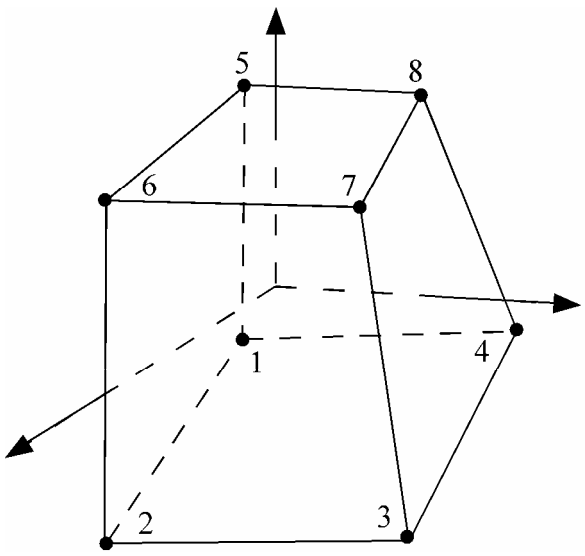


Figure 18.9. The orientation of the triad for the local material directions is stored relative to the base of the solid element. The base is defined by nodes 1-4 of the element connectivity.

18.9 General Erosion Criteria for Solid Elements

Several erosion criteria are available that are independent of the material models. Each one is applied independently, and once any one of them is satisfied, the element is deleted from the calculation. The criteria for failure are:

1. $P \geq P_{\min}$ where P is the pressure (positive in compression), and $P_{\min}$ is the pressure at failure.
2. $\sigma_1 \geq \sigma_{\max}$, where σ_1 is the maximum principal stress, and $\sigma_{\max}$ .is the principal stress at failure.
3. $\sqrt{\frac{3}{2}\sigma'_{ij}\sigma'_{ij}} \geq \bar{\sigma}_{\max}$, where σ'_{ij} are the deviatoric stress components, and $\bar{\sigma}_{\max}$ is the equivalent stress at failure.
4. $\varepsilon_1 \geq \varepsilon_{\max}$, where ε_1 is the maximum principal strain, and $\varepsilon_{\max}$ is the principal strain at failure.
5. $\gamma_1 \geq \gamma_{\max}$, where γ_1 is the shear strain, and $\gamma_{\max}$ is the shear strain at failure.
6. The Tuler-Butcher criterion,

$$\int_0^t [\max(0, \sigma_1 - \sigma_0)]^2 \, dt \geq K_f \tag{18.52}$$

where σ_1 is the maximum principal stress, σ_0 is a specified threshold stress, $\sigma_1 \geq \sigma_0 \geq 0$, and K_f is the stress impulse for failure. Stress values below the threshold value are too low to cause fracture even for very long duration loadings. Typical constants are given in Table 18.2 below [Rajendran, 1989].

These failure models apply to solid elements with one point integration in 2 and 3 dimensions.

Material	σ_0 (Kbar)	λ	K_f
1020 Steel	10.0	2	12.5
OFHC Copper	3.60	2	10.0
C1008	14.0	2	0.38
HY100	15.7	2	61.0
7039-T64	8.60	2	3.00

Table 18.2. Typical constants for the Tuler-Bucher criterion.

18.10 Strain Output to the LS-DYNA Database

The strain tensors that are output to the LS-DYNA database from the solid, shell, and beam elements are integrated in time. These strains are similar to the logarithmic strain measure and are based on an integration of the strain rate tensor. Admittedly, the shear strain components do not integrate as logarithmic strain measures, but in spite of this, we have found that the strains output from LS-DYNA are far more useful than those computed in LS-DYNA. The time integration of the strain tensor in LS-DYNA maintains objectivity in the sense that rigid body motions do not cause spurious straining.

Recall, the spin tensor and strain rate tensor, Equations (18.3) and (18.5), respectively:

$$\omega_{ij} = \frac{1}{2}\left(\frac{\partial v_i}{\partial x_j} - \frac{\partial v_j}{\partial x_i} \right) \tag{18.3}$$

$$\dot{\varepsilon}_{ij} = \frac{1}{2}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \tag{18.5}$$

In updating the strains from time n to $n+1$, the following formula is used:

$$\varepsilon_{ij}^{n+1} = \varepsilon_{ij}^{n} + \rho_{ij}^{n} + \dot{\varepsilon}_{ij}^{n+1/2} \Delta t^{n+1/2} \tag{18.53}$$

where ρ_{ij}^{n} gives the rotational correction that transforms the strain tensor at time t^n into the configuration at t^{n+1}

$$\rho_{ij}^{n} = \left(\varepsilon_{ip}^{n} \omega_{pj}^{n+1/2} + \varepsilon_{jp}^{n} \omega_{pi}^{n+1/2} \right) \Delta t^{n+1/2} \tag{18.54}$$

For shell elements we integrate the strain tensor at the inner and outer integration points and output two tensors per element. When the mid surface strains are plotted in LS-PREPOST, these are the average values.

19. MATERIAL MODELS

LS-DYNA accepts a wide range of material and equation of state models, each with a unique number of history variables. Approximately 150 material models are implemented, and space has been allotted for up to 10 user-specified models.

1 Elastic
2 Orthotropic Elastic
3 Kinematic/Isotropic Elastic-Plastic
4 Thermo-Elastic-Plastic
5 Soil and Crushable/Non-crushable Foam
6 Viscoelastic
7 Blatz - Ko Rubber
8 High Explosive Burn
9 Null Hydrodynamics
10 Isotropic-Elastic-Plastic-Hydrodynamic
11 Temperature Dependent, Elastoplastic, Hydrodynamic
12 Isotropic-Elastic-Plastic
13 Elastic-Plastic with Failure Model
14 Soil and Crushable Foam with Failure Model
15 Johnson/Cook Strain and Temperature Sensitive Plasticity
16 Pseudo TENSOR Concrete/Geological Model
17 Isotropic Elastic-Plastic Oriented Crack Model
18 Power Law Isotropic Plasticity
19 Strain Rate Dependent Isotropic Plasticity
20 Rigid
21 Thermal Orthotropic Elastic
22 Composite Damage Model
23 Thermal Orthotropic Elastic with 12 Curves
24 Piecewise Linear Isotropic Plasticity
25 Inviscid Two Invariant Geologic Cap Model
26 Metallic Honeycomb
27 Compressible Mooney-Rivlin Rubber
28 Resultant Plasticity
29 Force Limited Resultant Formulation
30 Closed-Form Update Shell Plasticity
31 Slightly Compressible Rubber Model
32 Laminated Glass Model
33 Barlat's Anisotropic Plasticity Model
34 Fabric
35 Kinematic/Isotropic Elastic-Plastic Green-Naghdi Rate
36 Barlat's 3-Parameter Plasticity Model
37 Transversely Anisotropic Elastic-Plastic
38 Blatz-Ko Compressible Foam
39 Transversely Anisotropic Elastic-Plastic with FLD
40 Nonlinear Elastic Orthotropic Material

41-50 User Defined Material Models
42 Planar Anisotropic Plasticity Model
48 Strain Rate Dependent Plasticity with Size Dependent Failure
51 Temperature and Rate Dependent Plasticity
52 Sandia's Damage Model
53 Low Density Closed Cell Polyurethane Foam
54-55 Composite Damage Model
57 Low Density Urethane Foam
58 Laminated Composite Fabric
59 Composite Failure
60 Elastic with Viscosity
61 Maxwell/Kelvin Viscoelastic
62 Viscous Foam
63 Isotropic Crushable Foam
64 Strain Rate Sensitive Power-Law Plasticity
65 Modified Zerilli/Armstrong
66 Linear Stiffness/Linear Viscous 3D Discrete Beam
67 Nonlinear Stiffness/Viscous 3D Discrete Beam
68 Nonlinear Plastic/Linear Viscous 3D Discrete Beam
69 Side Impact Dummy Damper, SID Damper
70 Hydraulic/Gas Damper
71 Cable
72 Concrete Damage Model
73 Low Density Viscoelastic Foam
74 Elastic Spring for the Discrete Beam
75 Bilkhu/Dubois Foam Model
76 General Viscoelastic
77 Hyperviscoelastic Rubber
78 Soil/Concrete
79 Hysteretic Soil
80 Ramberg-Osgood Plasticity
81 Plastic with Damage
82 Isotropic Elastic-Plastic with Anisotropic Damage
83 Fu-Chang's Foam with Rate Effects
84-85 Winfrith Concrete
84 Winfrith Concrete Reinforcement
86 Orthotropic-Viscoelastic
87 Cellular Rubber
88 MTS Model
89 Plasticity Polymer
90 Acoustic
91 Soft Tissue
93 Elastic 6DOF Spring Discrete Beam
94 Inelastic Spring Discrete Beam
95 Inelastic 6DOF Spring Discrete Beam
96 Brittle Damage Model

97	General Joint Discrete Beam
100	Spot weld
101	GE Thermoplastics
102	Hyperbolic Sin
103	Anisotropic Viscoplastic
104	Damage 1
105	Damage 2
106	Elastic Viscoplastic Thermal
110	Johnson-Holmquist Ceramic Model
111	Johnson-Holmquist Concrete Model
112	Finite Elastic Strain Plasticity
113	Transformation Induced Plasticity
114	Layered Linear Plasticity
115	Elastic Creep Model
116	Composite Lay-Up Model
117-118	Composite Matrix
119	General Spring and Damper Model
120	Gurson Dilational-Plastic Model
120	Gurson Model with Rc-Dc
121	Generalized Nonlinear 1DOF Discrete Beam
122	Hill 3RC
123	Modified Piecewise Linear Plasticity
124	Tension-Compression Plasticity
126	Metallic Honeycomb
127	Arruda-Boyce rubber
128	Anisotropic heart tissue
129	Lung tissue
130	Special Orthotropic
131	Isotropic Smeared Crack
132	Orthotropic Smeared Crack
133	Barlat YLD2000
134	Viscoelastic Fabric
139	Modified Force Limited
140	Vacuum
141	Rate Sensitive Polymer
142	Transversely Anisotropic Crushable Foam
143	Wood Model
144	Pitzer Crushable Foam
145	Schwer Murray Cap Model
146	1DOF Generalized Spring
147	FHWA Soil Model
148	Gas Mixture
150	CFD
151	EMMI
154	Deshpande-Fleck Foam
156	Muscle

158	Rate Sensitive Composite Fabric
159	Continuous Surface Cap Model
161-162	Composite MSC
163	Modified Crushable Foam
164	Brain Linear Viscoelastic
166	Moment Curvature Beam
169	Arup Adhesive
170	Resultant Anisotropic
175	Viscoelastic Maxwell
176	Quasilinear Viscoelastic
177	Hill Foam
178	Viscoelastic Hill Foam
179	Low Density Synthetic Foam
181	Simplified Rubber/Foam
183	Simplified Rubber with Damage
184	Cohesive Elastic
185	Cohesive TH
191	Seismic Beam
192	Soil Brick
193	Drucker Prager
194	RC Shear Wall
195	Concrete Beam
196	General Spring Discrete Beam
197	Seismic Isolator
198	Jointed Rock

In the table below, a list of the available material models and the applicable element types are given. Some materials include strain rate sensitivity, failure, equations of state, and thermal effects and this is also noted. General applicability of the materials to certain kinds of behavior is suggested in the last column.

Material Number	Material Title	Bricks	Beams	Thin Shells	Thick Shells	Strain-Rate Effects	Failure	Equation-of-State	Thermal Effects	Notes:
										Gn General Cm Composites Cr Ceramics Fl Fluids Fm Foam Gl Glass Hy Hydro-dyn Mt Metal Pl Plastic Rb Rubber Sl Soil/Conc
1	Elastic	Y	Y	Y	Y					Gn, Fl
2	Orthotropic Elastic (Anisotropic - solids)	Y		Y	Y					Cm, Mt
3	Plastic Kinematic/Isotropic	Y	Y	Y	Y	Y	Y			Cm, Mt, Pl
4	Elastic Plastic Thermal	Y	Y	Y	Y				Y	Mt, Pl
5	Soil and Foam	Y								Fm, Sl
6	Linear Viscoelastic	Y	Y	Y		Y				Rb
7	Blatz-Ko Rubber	Y		Y						Rb, Polyurethane
8	High Explosive Burn	Y						Y		Hy
9	Null Material	Y					Y	Y	Y	Fl, Hy
10	Elastic Plastic Hydro(dynamic)	Y					Y	Y		Hy, Mt
11	Steinberg: Temp. Dependent Elastoplastic	Y				Y	Y	Y	Y	Hy, Mt
12	Isotropic Elastic Plastic	Y		Y	Y					Mt
13	Isotropic Elastic Plastic with Failure	Y					Y			Mt
14	Soil and Foam with Failure	Y					Y			Fm, Sl
15	Johnson/Cook Plasticity Model	Y		Y		Y	Y	Y	Y	Hy, Mt
16	Pseudo TENSOR Geological Model	Y				Y	Y	Y		Sl
17	Oriented Crack (Elastoplastic with Fracture)	Y					Y	Y		Hy, Mt, Pl
18	Power Law Plasticity (Isotropic)	Y	Y	Y	Y	Y				Mt, Pl
19	Strain Rate Dependent Plasticity	Y		Y	Y	Y	Y			Mt, Pl
20	Rigid	Y	Y	Y	Y					

Material Number	Material Title	Bricks	Beams	Thin Shells	Thick Shells	Strain-Rate Effects	Failure	Equation-of-State	Thermal Effects	Notes: Gn General, Cm Composites, Cr Ceramics, Fl Fluids, Fm Foam, Gl Glass, Hy Hydro-dyn, Mt Metal, Pl Plastic, Rb Rubber, Sl Soil/Conc
21	Orthotropic Thermal (Elastic)	Y		Y	Y				Y	Gn
22	Composite Damage	Y		Y	Y		Y			Cm
23	Temperature Dependent Orthotropic	Y		Y	Y				Y	Cm
24	Piecewise Linear Plasticity (Isotropic)	Y	Y	Y	Y	Y	Y			Mt, Pl
25	Inviscid Two Invariant Geologic Cap	Y								Sl
26	Honeycomb	Y				Y	Y			Cm, Fm, Sl
27	Mooney-Rivlin Rubber	Y		Y						Rb
28	Resultant Plasticity		Y	Y						Mt
29	Force Limited Resultant Formulation		Y							
30	Closed Form Update Shell Plasticity			Y	Y					Mt
31	Slightly Compressible Rubber	Y								Rb
32	Laminated Glass (Composite)			Y	Y		Y			Cm, Gl
33	Barlat Anisotropic Plasticity	Y		Y	Y					Cr, Mt
34	Fabric			Y						
35	Plastic Green-Naghdi Rate	Y				Y				Mt
36	3-Parameter Barlat Plasticity			Y						Mt
37	Transversely Anisotropic Elastic Plastic			Y	Y					Mt
38	Blatz-Ko Foam	Y		Y						Fm, Pl
39	FLD Transversely Anisotropic			Y	Y					Mt
40	Nonlinear Orthotropic			Y			Y		Y	Cm
41-50	User Defined Materials	Y	Y	Y	Y	Y	Y	Y	Y	Gn
42	Planar Anisotropic Plasticity Model									

Material Number	Material Title	Bricks	Beams	Thin Shells	Thick Shells	Strain-Rate Effects	Failure	Equation-of-State	Thermal Effects	Notes:
										Gn General Cm Composites Cr Ceramics Fl Fluids Fm Foam Gl Glass Hy Hydro-dyn Mt Metal Pl Plastic Rb Rubber Sl Soil/Conc
51	Bamman (Temp/Rate Dependent Plasticity)	Y		Y	Y	Y			Y	Gn
52	Bamman Damage	Y		Y	Y	Y	Y		Y	Mt
53	Closed Cell Foam (Low Density Polyurethane)	Y								Fm
54	Composite Damage with Change Failure			Y			Y			Cm
55	Composite Damage with Tsai-Wu Failure			Y			Y			Cm
56										
57	Low Density Urethane Foam	Y				Y	Y			Fm
58	Laminated Composite Fabric			Y						
59	Composite Failure (Plasticity Based)	Y		Y			Y			Cm, Cr
60	Elastic with Viscosity (Viscous Glass)	Y		Y		Y			Y	Gl
61	Kelvin-Maxwell Viscoelastic	Y				Y				Fm
62	Viscous Foam (Crash Dummy Foam)	Y				Y				Fm
63	Isotropic Crushable Foam	Y				Y				Fm
64	Rate Sensitive Powerlaw Plasticity	Y		Y	Y	Y				Mt
65	Zerilli-Armstrong (Rate/Temp Plasticity)	Y		Y		Y		Y	Y	Mt
66	Linear Elastic Discrete Beam		Y			Y				
67	Nonlinear Elastic Discrete Beam		Y			Y				
68	Nonlinear Plastic Discrete Beam		Y			Y	Y			
69	SID Damper Discrete Beam		Y			Y				
70	Hydraulic Gas Damper Discrete Beam		Y			Y				
71	Cable Discrete Beam (Elastic)		Y							
72	Concrete Damage	Y				Y	Y	Y		Sl

Material Number	Material Title	Bricks	Beams	Thin Shells	Thick Shells	Strain-Rate Effects	Failure	Equation-of-State	Thermal Effects	Notes: Gn General, Cm Composites, Cr Ceramics, Fl Fluids, Fm Foam, Gl Glass, Hy Hydro-dyn, Mt Metal, Pl Plastic, Rb Rubber, Sl Soil/Conc
73	Low Density Viscous Foam	Y				Y	Y			Fm
74	Elastic Spring for the Discrete Beam									
75	Bilkhu/Dubois Foam (Isotropic)	Y				Y				Fm
76	General Viscoelastic (Maxwell Model)	Y				Y				Rb
77	Hyperelastic and Ogden Rubber	Y								Rb
78	Soil Concrete	Y				Y				Sl
79	Hysteretic Soil (Elasto-Perfectly Plastic)	Y					Y			Sl
80	Ramberg Osgood Plasticity									
81	Plasticity with Damage (Elasto-Plastic)	Y	Y	Y	Y	Y	Y			Mt, Pl
82	Isotropic Elastic-Plastic with Anisotropic Damage									
83	Fu Chang Foam	Y				Y	Y			Fm
84	Winfrith Concrete Reinforcement	Y								
85										
86	Orthotropic Viscoelastic			Y		Y				Rb
87	Cellular Rubber	Y				Y				Rb
88	MTS	Y		Y		Y		Y		Mt
89	Plasticity Polymer			Y						
90	Acoustic	Y								Fl
91	Soft Tissue	Y		Y						
93	Elastic 6DOF Spring Discrete Beam		Y							
94	Inelastic Spring Discrete Beam		Y							

Material Number	Material Title	Bricks	Beams	Thin Shells	Thick Shells	Strain-Rate Effects	Failure	Equation-of-State	Thermal Effects	Notes:
										Gn General Cm Composites Cr Ceramics Fl Fluids Fm Foam Gl Glass Hy Hydro-dyn Mt Metal Pl Plastic Rb Rubber Sl Soil/Conc
95	Inelastic 6DOF Spring Discrete Beam		Y							
96	Brittle Damage	Y				Y	Y			
97	General Joint Discrete Beam		Y							
98	Simplified Johnson Cook	Y	Y	Y	Y					
99	Simplified Johnson Cook Orthotropic Damage									
100	Spotweld		Y							
101	GEPLASTIC Strate2000a			Y						
102	Inv Hyperbolic Sin	Y								
103	Anisotropic Viscoplastic	Y		Y						
104	Damage 1	Y		Y						
105	Damage 2	Y		Y						
106	Elastic Viscoplastic Thermal	Y		Y					Y	
107										
108										
109										
110	Johnson Holmquist Ceramics	Y								
111	Johnson Holmquist Concrete	Y								
112	Finite Elastic Strain Plasticity	Y								
113	TRIP			Y	Y				Y	Mt
114	Layered Linear Plasticity			Y	Y					
115	Unified Creep	Y								

Material Number	Material Title	Bricks	Beams	Thin Shells	Thick Shells	Strain-Rate Effects	Failure	Equation-of-State	Thermal Effects	Notes:
										Gn General Cm Composites Cr Ceramics Fl Fluids Fm Foam Gl Glass Hy Hydro-dyn Mt Metal Pl Plastic Rb Rubber Sl Soil/Conc
116	Composite Layup			Y						
117	Composite Matrix			Y						
118	Composite Direct			Y						
119	General Nonlinear 6DOF Discrete Beam		Y			Y	Y			
120	Gurson			Y						
121	Generalized Nonlinear 1DOF Discrete Beam		Y							
122	Hill 3RC									
123	Modified Piecewise Linear Plasticity			Y	Y					
124	Plasticity Compression Tension	Y								
126	Modified Honeycomb	Y								
127	Arruda Boyce Rubber	Y								
128	Heart Tissue	Y								
129	Lung Tissue	Y								
130	Special Orthotropic			Y						
131	Isotropic Smeared Crack	Y					Y			Mt, Cm
132	Orthotropic Smeared Crack	Y					Y			Mt, Cm
133	Barlat YLD2000									
139	Modified Force Limited		Y							
140	Vacuum									
141	Rate Sensitive Polymer									

Material Number	Material Title	Bricks	Beams	Thin Shells	Thick Shells	Strain-Rate Effects	Failure	Equation-of-State	Thermal Effects	Notes:
										Gn General Cm Composites Cr Ceramics Fl Fluids Fm Foam Gl Glass Hy Hydro-dyn Mt Metal Pl Plastic Rb Rubber Sl Soil/Conc
142	Transversely Anisotropic Crushable Foam									
143	Wood									
144	Pitzer Crushable Foam									
145	Schwer Murray Cap Model									
146	1DOF Generalized Spring									
147	FHWA Soil									
147	FHWA Soil Nebraska									
148	Gas Mixture									
150	CFD									
151	EMMI	Y			Y	Y	Y		Y	Mt
154	Deshpande Fleck Foam									
156	Muscle		Y			Y				
158	Rate Sensitive Composite Fabric			Y	Y	Y	Y			Cm
159	CSMC	Y			Y	Y	Y			Sl
161	Composite MSC	Y								
163	Modified Crushable Foam									
164	Brain Linear Viscoelastic									
166	Moment Curvature Beam		Y							
169	Arup Adhesive	Y			Y		Y			Pb
170	Resultant Anisotropic			Y	Y					Pl
175	Viscoelastic Thermal	Y		Y	Y	Y			Y	Rb
176	Quasilinear Viscoelastic									

Material Number	Material Title	Bricks	Beams	Thin Shells	Thick Shells	Strain-Rate Effects	Failure	Equation-of-State	Thermal Effects	Notes:
										Gn General Cm Composites Cr Ceramics Fl Fluids Fm Foam Gl Glass Hy Hydro-dyn Mt Metal Pl Plastic Rb Rubber Sl Soil/Conc
177	Hill Foam									
178	Viscoelastic Hill Foam									
179	Low Density Synthetic Foam									
181	Simplified Rubber									
183	Simplified Rubber with Damage	Y		Y	Y	Y	Y			Rb
184	Cohesive Elastic	Y					Y			Cm, Mt
185	Cohesive TH	Y					Y			Cm, Mt
191	Seismic Beam		Y							
192	Soil Brick	Y								
193	Drucker Prager	Y								
194	RC Shear Wall			Y						
195	Concrete Beam		Y							
196	General Spring Discrete Beam		Y							
197	Seismic Isolator		Y				Y			Mt
198	Jointed Rock	Y					Y			
DS1	Spring Elastic (Linear)		Y							
DS2	Damper Viscous (Linear)		Y			Y				
DS3	Spring Elastoplastic (Isotropic)		Y							
DS4	Spring Nonlinear Elastic		Y			Y				
DS5	Damper Nonlinear Elastic		Y			Y				
DS6	Spring General Nonlinear		Y							

Material Number	Material Title	Bricks	Beams	Thin Shells	Thick Shells	Strain-Rate Effects	Failure	Equation-of-State	Thermal Effects	Notes:
										Gn General Cm Composites Cr Ceramics Fl Fluids Fm Foam Gl Glass Hy Hydro-dyn Mt Metal Pl Plastic Rb Rubber Sl Soil/Conc
DS7	Spring Maxwell (Three Parameter Viscoelastic)		Y			Y				
DS8	Spring Inelastic (Tension or Compression)		Y							
DS13	Spring Trilinear Degrading									
DS14	Spring Squat Shearwall									
DS15	Spring Muscle									
SB1	Seatbelt									
T01	Thermal Isotropic	Y		Y					Y	
T02	Thermal Orthotropic	Y		Y					Y	
T03	Thermal Isotropic (Temp. Dependent)	Y		Y					Y	
T04	Thermal Orthotropic (Temp. Dependent)	Y		Y					Y	
T05	Thermal Isotropic (Phase Change)	Y		Y					Y	
T06	Thermal Isotropic (Temp Dep-Load Curve)	Y		Y					Y	
T11	Thermal User Defined	Y		Y					Y	

Material Model 1: Elastic

In this elastic material we compute the co-rotational rate of the deviatoric Cauchy stress tensor as

$$s_{ij}^{\nabla^{n+1/2}} = 2G\dot{\varepsilon}'^{\,n+1/2}_{ij} \tag{19.1.1}$$

and pressure

$$p^{n+1} = -K \ln V^{n+1} \tag{19.1.2}$$

where G and K are the elastic shear and bulk moduli, respectively, and V is the relative volume, i.e., the ratio of the current volume to the initial volume.

Material Model 2: Orthotropic Elastic

The material law that relates second Piola-Kirchhoff stress S to the Green-St. Venant strain E is

$$S = C \cdot E = T^t C_l T \cdot E \tag{19.2.1}$$

where T is the transformation matrix [Cook 1974].

$$T = \begin{bmatrix} l_1^2 & m_1^2 & n_1^2 & l_1 m_1 & m_1 n_1 & n_1 l_1 \\ l_2^2 & m_2^2 & n_2^2 & l_2 m_2 & m_2 n_2 & n_2 l_2 \\ l_3^2 & m_3^2 & n_3^2 & l_3 m_3 & m_3 n_3 & n_3 l_3 \\ 2l_1 l_2 & 2m_1 m_2 & 2n_1 n_2 & (l_1 m_2 + l_1 m_1) & (m_1 n_2 + m_2 n_1) & (n_1 l_2 + n_2 l_1) \\ 2l_2 l_3 & 2m_2 m_3 & 2n_2 n_3 & (l_2 m_3 + l_3 m_2) & (m_2 n_3 + m_3 n_2) & (n_2 l_3 + n_3 l_2) \\ 2l_3 l_1 & 2m_3 m_1 & 2n_3 n_1 & (l_3 m_1 + l_1 m_3) & (m_3 n_1 + m_1 n_3) & (n_3 l_1 + n_1 l_3) \end{bmatrix} \tag{19.2.2}$$

l_i, m_i, n_i are the direction cosines

$$x_i^{'} = l_i x_1 + m_i x_2 + n_i x_3 \qquad for\ i = 1,2,3 \tag{19.2.3}$$

and $x_i^{'}$ denotes the material axes. The constitutive matrix C_l is defined in terms of the material axes as

$$C_l^{-1} = \begin{bmatrix} \frac{1}{E_{11}} & -\frac{\upsilon_{21}}{E_{22}} & -\frac{\upsilon_{31}}{E_{33}} & 0 & 0 & 0 \\ -\frac{\upsilon_{12}}{E_{11}} & \frac{1}{E_{22}} & -\frac{\upsilon_{32}}{E_{33}} & 0 & 0 & 0 \\ -\frac{\upsilon_{13}}{E_{11}} & -\frac{\upsilon_{23}}{E_{22}} & \frac{1}{E_{33}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{12}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{23}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{31}} \end{bmatrix} \tag{19.2.4}$$

where the subscripts denote the material axes, i.e.,

$$\upsilon_{ij} = \upsilon_{x_i' x_j'} \quad \text{and} \quad E_{ii} = E_{x_i'} \tag{19.2.5}$$

Since C_l is symmetric

$$\frac{\upsilon_{12}}{E_{11}} = \frac{\upsilon_{21}}{E_{22}}, \text{ etc.} \tag{19.2.6}$$

The vector of Green-St. Venant strain components is

$$E^t = \lfloor E_{11}, E_{22}, E_{33}, E_{12}, E_{23}, E_{31}, \rfloor \tag{19.2.7}$$

After computing S_{ij}, we use Equation (18.32) to obtain the Cauchy stress. This model will predict realistic behavior for finite displacement and rotations as long as the strains are small.

Material Model 3: Elastic Plastic with Kinematic Hardening

Isotropic, kinematic, or a combination of isotropic and kinematic hardening may be obtained by varying a parameter, called β between 0 and 1. For β equal to 0 and 1, respectively, kinematic and isotropic hardening are obtained as shown in Figure 19.3.1. Krieg and Key [1976] formulated this model and the implementation is based on their paper.

In isotropic hardening, the center of the yield surface is fixed but the radius is a function of the plastic strain. In kinematic hardening, the radius of the yield surface is fixed but the center translates in the direction of the plastic strain. Thus the yield condition is

$$\phi = \frac{1}{2}\xi_{ij}\,\xi_{ij} - \frac{\sigma_y^{\,2}}{3} = 0 \tag{19.3.1}$$

where

$$\xi_{ij} = s_{ij} - \alpha_{ij} \tag{19.3.2}$$

$$\sigma_y = \sigma_0 + \beta E_p \, \varepsilon_{eff}^p \tag{19.3.3}$$

The co-rotational rate of α_{ij} is

$$\alpha_{ij}^{\nabla} = (1-\beta)\frac{2}{3} E_p \, \dot{\varepsilon}_{ij}^p \tag{19.3.4}$$

Hence,

$$\alpha_{ij}^{n+1} = \alpha_{ij}^n + \left(\alpha_{ij}^{\nabla^{n+1/2}} + \alpha_{ik}^n \, \Omega_{kj}^{n+1/2} + \alpha_{jk}^n \, \Omega_{ki}^{n+1/2} \right) \Delta t^{n+1/2}. \tag{19.3.5}$$

Strain rate is accounted for using the Cowper-Symonds [Jones 1983] model which scales the yield stress by a strain rate dependent factor

$$\sigma_y = \left[1 + \left(\frac{\dot{\varepsilon}}{C} \right)^{\frac{1}{p}} \right] \left(\sigma_0 + \beta E_p \, \varepsilon_{eff}^p \right) \tag{19.3.6}$$

where p and C are user defined input constants and $\dot{\varepsilon}$ is the strain rate defined as:

$$\dot{\varepsilon} = \sqrt{\dot{\varepsilon}_{ij} \dot{\varepsilon}_{ij}} \tag{19.3.7}$$

The current radius of the yield surface, σ_y, is the sum of the initial yield strength, σ_0, plus the growth $\beta E_p \varepsilon_{eff}^p$, where E_p is the plastic hardening modulus

$$E_p = \frac{E_t E}{E - E_t} \tag{19.3.8}$$

and ε_{eff}^p is the effective plastic strain

$$\varepsilon_{eff}^p = \int_0^t \left(\frac{2}{3} \dot{\varepsilon}_{ij}^{\,p} \, \dot{\varepsilon}_{ij}^p \right)^{1/2} dt \tag{19.3.9}$$

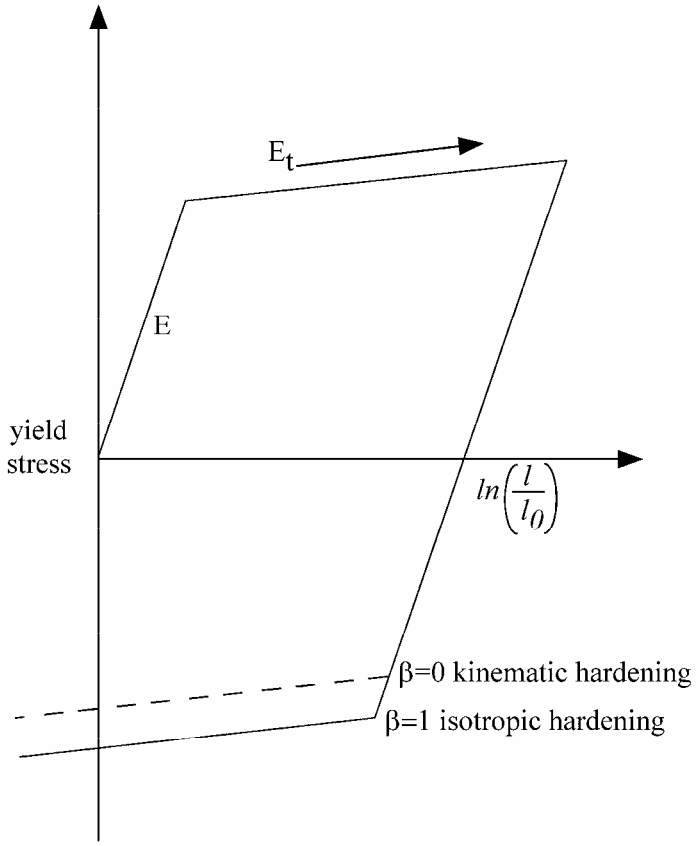


Figure 19.3.1. Elastic-plastic behavior with isotropic and kinematic hardening where l_0 and l are the undeformed and deformed length of uniaxial tension specimen, respectively.

The plastic strain rate is the difference between the total and elastic (right superscript *e*) strain rates:

$$\dot{\varepsilon}_{ij}^{p} = \dot{\varepsilon}_{ij} - \dot{\varepsilon}_{ij}^{e} \tag{19.3.10}$$

In the implementation of this material model, the deviatoric stresses are updated elastically, as described for model 1, but repeated here for the sake of clarity:

$$\sigma_{ij}^{*} = \sigma_{ij}^{n} + C_{ijkl}\Delta\varepsilon_{kl} \tag{19.3.11}$$

where

σ_{ij}^{*} is the trial stress tensor,

σ_{ij}^{n} is the stress tensor from the previous time step,

C_{ijkl} is the elastic tangent modulus matrix,

$\Delta\varepsilon_{kl}$ is the incremental strain tensor.

and, if the yield function is satisfied, nothing else is done. If, however, the yield function is violated, an increment in plastic strain is computed, the stresses are scaled back to the yield surface, and the yield surface center is updated.

Let s_{ij}^* represent the trial elastic deviatoric stress state at n+1

$$s_{ij}^* = \sigma_{ij}^* - \tfrac{1}{3}\sigma_{kk}^* \tag{19.3.12}$$

and

$$\xi_{ij}^* = s_{ij}^* - \alpha_{ij} \quad . \tag{19.3.13}$$

Define the yield function,

$$\phi = \frac{3}{2}\xi_{ij}^*\,\xi_{ij}^* - \sigma_y^2 = \Lambda^2 - \sigma_y^2 \begin{cases} \le 0 \;\; for \text{ elastic or neutral loading} \\ > 0 \;\; for \text{ plastic harding} \end{cases} \tag{19.3.14}$$

For plastic hardening then

$$\varepsilon_{eff}^{p^{n+1}} = \varepsilon_{eff}^{p^n} + \frac{\Lambda - \sigma_y}{3G + E_p} = \varepsilon_{eff}^{p^n} + \Delta\varepsilon_{eff}^p \tag{19.3.15}$$

scale back the stress deviators

$$\sigma_{ij}^{n+1} = \sigma_{ij}^* - \frac{3G\Delta\varepsilon_{eff}^p}{\Lambda}\xi_{ij}^* \tag{19.3.16}$$

and update the center:

$$\alpha_{ij}^{n+1} = \alpha_{ij}^n + \frac{(1-\beta)E_p\,\Delta\varepsilon_{eff}^p}{\Lambda}\,\xi_{ij}^* \tag{19.3.17}$$

Plane Stress Plasticity

The plane stress plasticity options apply to beams, shells, and thick shells. Since the stresses and strain increments are transformed to the lamina coordinate system for the constitutive evaluation, the stress and strain tensors are in the local coordinate system.

The application of the Jaumann rate to update the stress tensor allows for the possibility that the normal stress, σ_{33}, will not be zero. The first step in updating the stress tensor is to compute a trial plane stress update assuming that the incremental strains are elastic. In the above, the normal strain increment $\Delta\varepsilon_{33}$ is replaced by the elastic strain increment

$$\Delta\varepsilon_{33} = -\frac{\sigma_{33} + \lambda(\Delta\varepsilon_{11} + \Delta\varepsilon_{22})}{\lambda + 2\mu} \tag{19.3.18}$$

where λ and μ are Lamé's constants.

When the trial stress is within the yield surface, the strain increment is elastic and the stress update is completed. Otherwise, for the plastic plane stress case, secant iteration is used to solve Equation (19.3.16)for the normal strain increment ($\Delta\varepsilon_{33}$) required to produce a zero normal stress:

$$\sigma_{33}^{i} = \sigma_{33}^{*} - \frac{3G\Delta\varepsilon_{eff}^{p^i}\xi_{33}}{\Lambda} \tag{19.3.19}$$

Here, the superscript i indicates the iteration number.

The secant iteration formula for $\Delta\varepsilon_{33}$ (the superscript p is dropped for clarity) is

$$\Delta\varepsilon_{33}^{i+1} = \Delta\varepsilon_{33}^{i-1} - \frac{\Delta\varepsilon_{33}^{i} - \Delta\varepsilon_{33}^{i-1}}{\sigma_{33}^{i} - \sigma_{33}^{i-1}}\,\sigma_{33}^{i-1} \tag{19.3.20}$$

where the two starting values are obtained from the initial elastic estimate and by assuming a purely plastic increment, i.e.,

$$\Delta\varepsilon_{33}^{1} = -\left(\Delta\varepsilon_{11} - \Delta\varepsilon_{22}\right) \tag{19.3.21}$$

These starting values should bound the actual values of the normal strain increment.

The iteration procedure uses the updated normal stain increment to update first the deviatoric stress and then the other quantities needed to compute the next estimate of the normal stress in Equation (19.3.19). The iterations proceed until the normal stress σ_{33}^{i} is sufficiently small. The convergence criterion requires convergence of the normal strains:

$$\frac{\left|\Delta\varepsilon_{33}^{i} - \Delta\varepsilon_{33}^{i-1}\right|}{\left|\Delta\varepsilon_{33}^{i+1}\right|} < 10^{-4} \tag{19.3.22}$$

After convergence, the stress update is completed using the relationships given in Equations (19.3.16) and (19.3.17)

Material Model 4: Thermo-Elastic-Plastic

This model was adapted from the NIKE2D [Hallquist 1979] code. A more complete description of its formulation is given in the NIKE2D user's manual.

Letting T represent the temperature, we compute the elastic co-rotational stress rate as

$$\sigma_{ij}^{\nabla} = C_{ijkl}\left(\dot{\varepsilon}_{kl} - \dot{\varepsilon}_{kl}^{T}\right) + \dot{\theta}_{ij}\,dT \tag{19.4.1}$$

where

$$\dot{\theta}_{ij} = \frac{dC_{ijkl}}{dT} C^{-1}_{klmn} \dot{\sigma}_{mn} \tag{19.4.2}$$

and $C_{ijk\ell}$ is the temperature dependent elastic constitutive matrix:

$$C_{ijk\ell} = \frac{E}{(1+\upsilon)(1-2\upsilon)} \begin{bmatrix} 1-\upsilon & \upsilon & \upsilon & 0 & 0 & 0 \\ \upsilon & 1-\upsilon & \upsilon & 0 & 0 & 0 \\ \upsilon & \upsilon & 1-\upsilon & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\upsilon}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\upsilon}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\upsilon}{2} \end{bmatrix} \tag{19.4.3}$$

where υ is Poisson's ratio. The thermal strain rate can be written in terms of the coefficient of thermal expansion α as:

$$\dot{\varepsilon}^T_{ij} = \alpha \dot{T} \delta_{ij} \tag{19.4.4}$$

When treating plasticity, we use a procedure analogous to that for material 3. We update the stresses elastically and check to see if we violate the isotropic yield function

$$\phi = \frac{1}{2} s_{ij} s_{ij} - \frac{\sigma_y(T)^2}{3} \tag{19.4.5}$$

where

$$\sigma_y(T) = \sigma_o(T) + E_p(T) \varepsilon^p_{eff} \tag{19.4.6}$$

The initial yield, σ_o, and plastic hardening modulus, E_p, are temperature dependent. If the behavior is elastic we do nothing; otherwise, we scale back the stress deviators by the factor f_s:

$$s^{n+1}_{ij} = f_s s^*_{ij} \tag{19.4.7}$$

where

$$f_s = \frac{\sigma_y}{\left(\frac{3}{2} s^*_{ij} s^*_{ij}\right)^{1/2}} \tag{19.4.8}$$

and update the plastic strain by the increment

$$\Delta\varepsilon_{eff}^{p} = \frac{(1-f_s)\left(\frac{3}{2}s_{ij}^{*}\,s_{ij}^{*}\right)^{1/2}}{G+3\,E_p} \tag{19.4.9}$$

Material Model 5: Soil and Crushable Foam

This model, due to Krieg [1972], provides a simple model for foam and soils whose material properties are not well characterized. We believe the other foam models in LS-DYNA are superior in their performance and are recommended over this model which simulates the crushing through the volumetric deformations. If the yield stress is too low, this foam model gives nearly fluid like behavior.

A pressure-dependent flow rule governs the deviatoric behavior:

$$\phi_s = \frac{1}{2}s_{ij}s_{ij} - \left(a_0 + a_1 p + a_2 p^2\right) \tag{19.5.1}$$

where a_0, a_1, and a_2 are user-defined constants. Volumetric yielding is determined by a tabulated curve of pressure versus volumetric strain. Elastic unloading from this curve is assumed to a tensile cutoff as illustrated in Figure 19.5.1.

Implementation of this model is straightforward. One history variable, the maximum volumetric strain in compression, is stored. If the new compressive volumetric strain exceeds the stored value, loading is indicated. When the yield condition is violated, the updated trial stresses, s_{ij}^{*}, are scaled back using a simple radial return algorithm:

$$s_{ij}^{n+1} = \left(\frac{a_0 + a_1 p + a_2 p^2}{\frac{1}{2}s_{ij}\,s_{ij}}\right)^{1/2} s_{ij}^{*} \tag{19.5.2}$$

If the hydrostatic tension exceeds the cutoff value, the pressure is set to the cutoff value and the deviatoric stress tensor is zeroed.

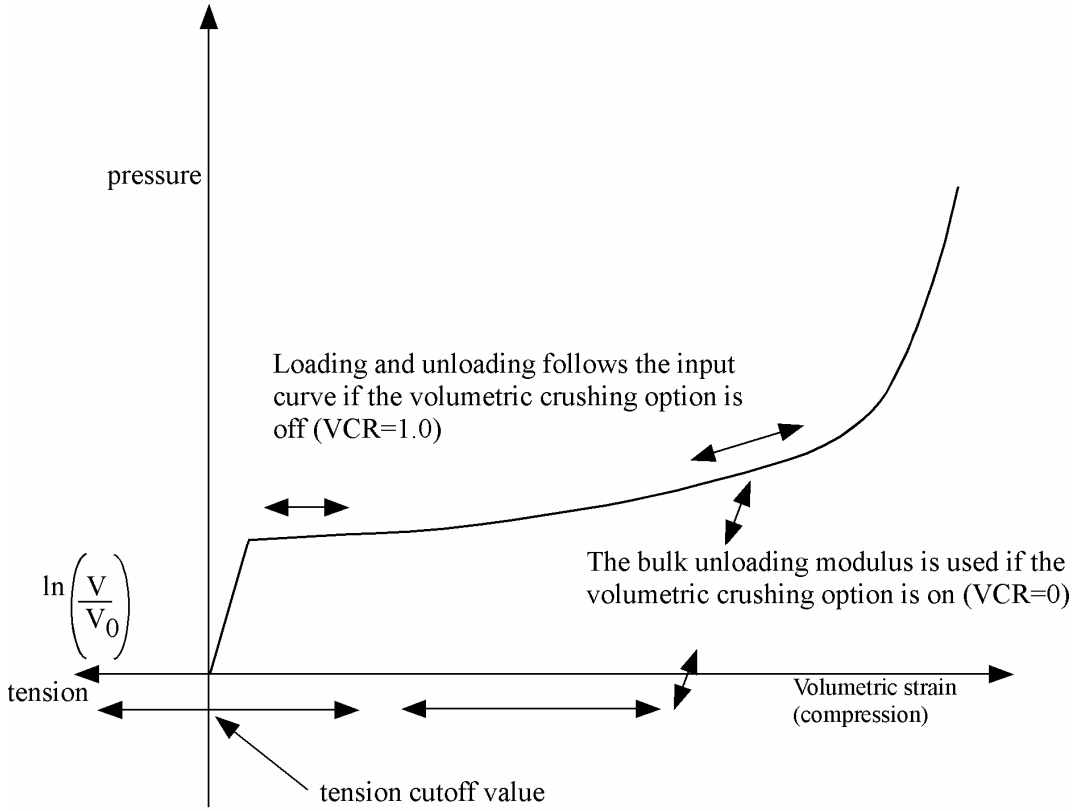


Figure 19.5.1. Volumetric strain versus pressure curve for soil and crushable foam model.

Material Model 6: Viscoelastic

In this model, linear viscoelasticity is assumed for the deviatoric stress tensor [Herrmann and Peterson 1968]:

$$s_{ij} = 2\int_0^t \phi(t-\tau)\frac{\partial \varepsilon'_{ij}(\tau)}{\partial \tau}\,d\tau \qquad (19.6.1)$$

where

$$\phi(t) = G_\infty + (G_0 - G_\infty)e^{-\beta t} \qquad (19.6.2)$$

is the shear relaxation modulus. A recursion formula is used to compute the new value of the hereditary integral at time t^{n+1} from its value at time t^n. Elastic bulk behavior is assumed:

$$p = K \ln V \qquad (19.6.3)$$

where pressure is integrated incrementally.

Material Model 7: Continuum Rubber

The hyperelastic continuum rubber model was studied by Blatz and Ko [1962]. In this model, the second Piola-Kirchhoff stress is given by

$$S_{ij} = G\left(V^{-1} C_{ij} - V^{-\frac{1}{1-2\upsilon}} \delta_{ij} \right) \tag{19.7.1}$$

where G is the shear modulus, V is the relative volume, υ is Poisson's ratio, and C_{ij} is the right Cauchy-Green strain:

$$C_{ij} = \frac{\partial x_k}{\partial X_i} \frac{\partial x_k}{\partial X_j} \tag{19.7.2}$$

after determining S_{ij} , it is transformed into the Cauchy stress tensor, σ_{ij} :

$$\sigma_{ij} = \frac{\rho}{\rho_0} \frac{\partial x_i}{\partial X_h} \frac{\partial x_j}{\partial X_l} S_{kl} \quad \sigma_{ij} = \frac{\rho}{\rho_0} \frac{\partial x_i}{\partial X_k} \frac{\partial x_j}{\partial X_l} S_{kl} \tag{19.7.3}$$

where ρ_0 and ρ are the initial and current density, respectively. The default value of υ is 0.463.

Material Model 8: Explosive Burn

Burn fractions, which multiply the equations of states for high explosives, control the release of chemical energy for simulating detonations. In the initialization phase, a lighting time t_1 is computed for each element by dividing the distance from the detonation point to the center of the element by the detonation velocity D. If multiple detonation points are defined, the closest point determines t_1. The burn fraction F is taken as the maximum

$$F = \max\left(F_1, F_2 \right) \tag{19.8.1}$$

where

$$F_1 = \begin{cases} \dfrac{2\left(t - t_l\right) D}{3\left(\dfrac{v_e}{A_{e_{\max}}} \right)} & if\ t > t_l \\ 0 & if\ t \le t_l \end{cases} \tag{19.8.2}$$

$$F_2 = \frac{1 - V}{1 - V_{CJ}} \tag{19.8.3}$$

where V_{CJ} is the Chapman-Jouguet relative volume and t is current time. If F exceeds 1, it is reset to 1. This calculation of the burn fraction usually requires several time steps for F to reach unity, thereby spreading the burn front over several elements. After reaching unity, F is held constant. This burn fraction calculation is based on work by Wilkins [1964] and is also discussed by Giroux [1973].

As an option, the high explosive material can behave as an elastic perfectly-plastic solid prior to detonation. In this case we update the stress tensor, to an elastic trial stress, $*s_{ij}^{n+1}$,

$$*s_{ij}^{n+1} = s_{ij}^{n} + s_{ip}\Omega_{pj} + s_{jp}\Omega_{pi} + 2G\dot{\varepsilon}'_{ij}\,dt \tag{19.8.4}$$

where G is the shear modulus, and $\dot{\varepsilon}'_{ij}$ is the deviatoric strain rate. The von Mises yield condition is given by:

$$\phi = J_2 - \frac{\sigma_y^2}{3} \tag{19.8.5}$$

where the second stress invariant, J_2, is defined in terms of the deviatoric stress components as

$$J_2 = \frac{1}{2} s_{ij} s_{ij} \tag{19.8.6}$$

and the yield stress is σ_y. If yielding has occurred, i.e., $\phi > 0$, the deviatoric trial stress is scaled to obtain the final deviatoric stress at time $n+1$:

$$s_{ij}^{n+1} = \frac{\sigma_y}{\sqrt{3J_2}} * s_{ij}^{n+1} \tag{19.8.7}$$

If $\phi \le 0$, then

$$s_{ij}^{n+1} = *s_{ij}^{n+1} \tag{19.8.8}$$

Before detonation pressure is given by the expression

$$p^{n+1} = K\left(\frac{1}{V^{n+1}} - 1\right) \tag{19.8.9}$$

where K is the bulk modulus. Once the explosive material detonates:

$$s_{ij}^{n+1} = 0 \tag{19.8.10}$$

and the material behaves like a gas.

The shadow burn option should be active when computing the lighting time if there exist elements within the mesh for which there is no direct line of sight from the detonation points. The shadow burn option is activated in the control section. The lighting time is based on the shortest distance through the explosive material. If inert obstacles exist within the explosive material, the lighting time will account for the extra time required for the detonation wave to travel around the obstacles. The lighting times also automatically accounts for variations in the detonation velocity if different explosives are used. No additional input is required for the shadow option but care must be taken when setting up the input. This option works for two and three-dimensional solid elements. It is recommended that for best results:

1. Keep the explosive mesh as uniform as possible with elements of roughly the same dimensions.
2. Inert obstacle such as wave shapers within the explosive must be somewhat larger than the characteristic element dimension for the automatic tracking to function properly. Generally, a factor of two should suffice. The characteristic element dimension is found by checking all explosive elements for the largest diagonal
3. The detonation points should be either within or on the boundary of the explosive. Offset points may fail to initiate the explosive.
4. Check the computed lighting times in the post processor LS-PREPOST. The lighting times may be displayed at time=0., state 1, by plotting component 7 (a component normally reserved for plastic strain) for the explosive material. The lighting times are stored as negative numbers. The negative lighting time is replaced by the burn fraction when the element ignites.

Line detonations may be approximated by using a sufficient number of detonation points to define the line. Too many detonation points may result in significant initialization cost.

Material Model 9: Null Material

For solid elements equations of state can be called through this model to avoid deviatoric stress calculations. A pressure cutoff may be specified to set a lower bound on the pressure. This model has been very useful when combined with the reactive high explosive model where material strength is often neglected. The null material should not be used to delete solid elements.

An optional viscous stress of the form

$$\sigma_{ij} = \mu \dot{\varepsilon}'_{ij}$$

is computed for nonzero μ where $\dot{\varepsilon}'_{ij}$ is the deviatoric strain rate.

Sometimes it is advantageous to model contact surfaces via shell elements which are not part of the structure, but are necessary to define areas of contact within nodal rigid bodies or between nodal rigid bodies. Beams and shells that use this material type are completely bypassed in the element processing. The Young's modulus and Poisson's ratio are used only for setting the contact interface stiffnesses, and it is recommended that reasonable values be input.

Material Model 10: Elastic-Plastic-Hydrodynamic

For completeness we give the entire derivation of this constitutive model based on radial return plasticity.

The pressure, p, deviatoric strain rate, $\dot{\varepsilon}'_{ij}$, deviatoric stress rate, $\dot{s}_{ij}$, volumetric strain rate, and $\dot{\varepsilon}_v$, are defined in Equation (19.10.1):

$$p = -\frac{1}{3}\sigma_{ij}\,\delta_{ij} \qquad \dot{\varepsilon}'_{ij} = \dot{\varepsilon}_{ij} - \frac{1}{3}\dot{\varepsilon}_v$$

$$s_{ij} = \sigma_{ij} + p\delta_{ij} \qquad \dot{\varepsilon}_v = \dot{\varepsilon}_{ij}\,\delta_{ij} \tag{19.10.1}$$

$$s^{\nabla}_{ij} = 2\mu\,\dot{\varepsilon}'_{ij} = 2G\dot{\varepsilon}'_{ij}$$

The Jaumann rate of the deviatoric stress, s^{∇}_{ij}, is given by:

$$s^{\nabla}_{ij} = \dot{s}_{ij} - s_{ip}\Omega_{pj} - s_{jp}\Omega_{pi} \tag{19.10.2}$$

First we update s^n_{ij} to s^{n+1}_{ij} elastically

$$^{*}s^{n+1}_{ij} = s^n_{ij} + s_{ip}\Omega_{pj} + s_{jp}\Omega_{pi} + 2G\dot{\varepsilon}'_{ij}\,dt = \underbrace{s^n_{ij} + R_{ij}}_{s^{R^n}_{ij}} + \underbrace{2G\dot{\varepsilon}'_{ij}\,dt}_{2G\Delta\varepsilon'_{ij}} \tag{19.10.3}$$

where the left superscript, *, denotes a trial stress value. The effective trial stress is defined by

$$s^{*} = \left(\frac{3}{2}\,{}^{*}s^{n+1}_{ij}\,{}^{*}s^{n+1}_{ij}\right)^{1/2} \tag{19.10.4}$$

and if s^{*} exceeds yield stress σ_y, the Von Mises flow rule:

$$\phi = \frac{1}{2}s_{ij}\,s_{ij} - \frac{\sigma_y{}^2}{3} \le 0 \tag{19.10.5}$$

is violated and we scale the trial stresses back to the yield surface, i.e., a radial return

$$s^{n+1}_{ij} = \frac{\sigma_y}{s^{*}}\;{}^{*}s^{n+1}_{ij} = m\,{}^{*}s^{n+1}_{ij} \tag{19.10.6}$$

The plastic strain increment can be found by subtracting the deviatoric part of the strain increment that is elastic, $\frac{1}{2G}\left(s_{ij}^{n+1}-s_{ij}^{R^n}\right)$, from the total deviatoric increment, $\Delta\varepsilon'_{ij}$, i.e.,

$$\Delta\varepsilon_{ij}^{p}=\Delta\varepsilon'_{ij}\ -\frac{1}{2G}\left(s_{ij}^{n+1}-s_{ij}^{R^n}\right) \tag{19.10.7}$$

Recalling that,

$$\Delta\varepsilon'_{ij}\ =\frac{{}^{*}s_{ij}^{n+1}-s_{ij}^{R^n}}{2G} \tag{19.10.8}$$

and substituting Equation (19.10.8) into (19.10.7) we obtain,

$$\Delta\varepsilon_{ij}^{p}=\frac{\left({}^{*}s_{ij}^{n+1}-s_{ij}^{n+1}\right)}{2G} \tag{19.10.9}$$

Substituting Equation (19.10.6)

$$s_{ij}^{n+1}=m\ {}^{*}s_{ij}^{n+1}$$

into Equation (19.10.9) gives,

$$\Delta\varepsilon_{ij}^{p}=\left(\frac{1-m}{2G}\right){}^{*}s_{ij}^{n+1}=\frac{1-m}{2Gm}s_{ij}^{n+1}=d\lambda s_{ij}^{n+1} \tag{19.10.10}$$

By definition an increment in effective plastic strain is

$$\Delta\varepsilon^{p}=\left(\frac{2}{3}\Delta\varepsilon_{ij}^{p}\ \Delta\varepsilon_{ij}^{p}\right)^{1/2} \tag{19.10.11}$$

Squaring both sides of Equation (19.10.10) leads to:

$$\Delta\varepsilon_{ij}^{p}\ \Delta\varepsilon_{ij}^{p}=\left(\frac{1-m}{2G}\right)^{2}\ {}^{*}s_{ij}^{n+1}\ {}^{*}s_{ij}^{n+1} \tag{19.10.12}$$

or from Equations (19.10.4) and (19.10.11):

$$\frac{3}{2}\Delta\varepsilon^{p^2}=\left(\frac{1-m}{2G}\right)^{2}\frac{2}{3}s^{*2} \tag{19.10.13}$$

Hence,

$$\therefore \Delta\varepsilon^p = \frac{1-m}{3G} s^* = \frac{s^* - \sigma_y}{3G} \tag{19.10.14}$$

where we have substituted for m from Equation (19.10.6)

$$m = \frac{\sigma_y}{s^*}$$

If isotropic hardening is assumed then:

$$\sigma_y^{n+1} = \sigma_y^n + E^p \Delta\varepsilon^p \tag{19.10.15}$$

and from Equation (19.10.14)

$$\Delta\varepsilon^p = \frac{\left(s^* - \sigma_y^{n+1}\right)}{3G} = \frac{\left(s^* - \sigma_y^n - E^p \Delta\varepsilon^p\right)}{3G} \tag{19.10.16}$$

Thus,

$$\left(3G + E^p\right)\Delta\varepsilon^p = \left(s^* - \sigma_y^n\right)$$

and solving for the incremental plastic strain gives

$$\Delta\varepsilon^p = \frac{\left(s^* - \sigma_y^n\right)}{\left(3G + E^p\right)} \tag{19.10.17}$$

The algorithm for plastic loading can now be outlined in five simple stress. If the effective trial stress exceeds the yield stress then

1. Solve for the plastic strain increment:

$$\Delta\varepsilon^p = \frac{\left(s^* - \sigma_y^n\right)}{\left(3G + E^p\right)}$$

2. Update the plastic strain:

$$\varepsilon^{p^{n+1}} = \varepsilon^{p^n} + \Delta\varepsilon^p .$$

3. Update the yield stress:

$$\sigma_y^{n+1} = \sigma_y^n + E^p \Delta\varepsilon^p$$

4. Compute the scale factor using the yield strength at time $n+1$:

$$m = \frac{\sigma_y^{n+1}}{s^*}$$

5. Radial return the deviatoric stresses to the yield surface:

$$s_{ij}^{n+1} = m^* s_{ij}^{n+1}$$

Material Model 11: Elastic-Plastic With Thermal Softening

Steinberg and Guinan [1978] developed this model for treating plasticity at high strain rates (10^5 s^{-1}) where enhancement of the yield strength due to strain rate effects is saturated out.

Both the shear modulus G and yield strength σ_y increase with pressure but decrease with temperature. As a melt temperature is reached, these quantities approach zero. We define the shear modulus before the material melts as

$$G = G_0 \left[1 + bpV^{1/3} - h\left(\frac{E - E_c}{3R'} - 300 \right) \right] e^{-\frac{fE}{E_m - E}} \qquad (19.11.1)$$

where G_0, b, h, and f are input parameters, E_c is the cold compression energy:

$$E_c(X) = \int_0^x p\,dx - \frac{900\,R'\exp(ax)}{(1-X)^{2(\gamma_o - a - \frac{1}{2})}}, \qquad (19.11.2)$$

where

$$X = 1 - V, \qquad (19.11.3)$$

and E_m is the melting energy:

$$E_m(X) = E_c(X) + 3R'T_m(X) \qquad (19.11.4)$$

which is a function of the melting temperature $T_m(X)$:

$$T_m(X) = \frac{T_{mo}\exp(2aX)}{(1-X)^{2(\gamma_o - a - 1/3)}} \qquad (19.11.5)$$

and the melting temperature T_{mo} at $\rho = \rho_0$. The constants γ_0 and a are input parameters. In the above equation, R' is defined by

$$R' = \frac{R\rho_o}{A} \tag{19.11.6}$$

where R is the gas constant and A is the atomic weight. The yield strength σ_y is given by:

$$\sigma_y = \sigma_0' \left[1 + b' p V^{1/3} - h\left(\frac{E - E_c}{3R'} - 300 \right) \right] e^{-\frac{fE}{E_m - E}} \tag{19.11.7}$$

If E_m exceeds E_i. Here, σ_0' is given by:

$$\sigma_0' = \sigma_0 \left[1 + \beta \left(\gamma_i + \bar{\varepsilon}^p \right) \right]^n \tag{19.11.8}$$

where γ_1 is the initial plastic strain, and b' and σ_0' are input parameters. Where σ_0' exceeds $\sigma_{\max}$, the maximum permitted yield strength, σ_0' is set to equal to $\sigma_{\max}$. After the material melts, σ_y and G are set to zero.

LS-DYNA fits the cold compression energy to a ten-term polynomial expansion:

$$E_c = \sum_{i=0}^{9} EC_i \eta^i \tag{19.11.9}$$

where EC_i is the ith coefficient and $\eta = \frac{\rho}{\rho_o}$. The least squares method is used to perform the fit [Kreyszig 1972]. The ten coefficients may also be specified in the input.

Once the yield strength and shear modulus are known, the numerical treatment is similar to that for material model 10.

Material Model 12: Isotropic Elastic-Plastic

The von Mises yield condition is given by:

$$\phi = J_2 - \frac{\sigma_y^2}{3} \tag{19.12.1}$$

where the second stress invariant, J_2, is defined in terms of the deviatoric stress components as

$$J_2 = \frac{1}{2} s_{ij} s_{ij} \tag{19.12.2}$$

and the yield stress, σ_y, is a function of the effective plastic strain, ε_{eff}^p, and the plastic hardening modulus, E_p:

$$\sigma_y = \sigma_0 + E_p \varepsilon_{eff}^p \tag{19.12.3}$$

The effective plastic strain is defined as:

$$\varepsilon_{eff}^p = \int_0^t d\varepsilon_{eff}^p \tag{19.12.4}$$

where $d\varepsilon_{eff}^p = \sqrt{\frac{2}{3} d\varepsilon_{ij}^p d\varepsilon_{ij}^p}$

and the plastic tangent modulus is defined in terms of the input tangent modulus, E_t, as

$$E_p = \frac{EE_t}{E - E_t} \tag{19.12.5}$$

Pressure is given by the expression

$$p^{n+1} = K\left(\frac{1}{V^{n+1}} - 1\right) \tag{19.12.6}$$

where K is the bulk modulus. This is perhaps the most cost effective plasticity model. Only one history variable, ε_{eff}^p, is stored with this model.

This model is not recommended for shell elements. In the plane stress implementation, a one-step radial return approach is used to scale the Cauchy stress tensor to if the state of stress exceeds the yield surface. This approach to plasticity leads to inaccurate shell thickness updates and stresses after yielding. This is the only model in LS-DYNA for plane stress that does not default to an iterative approach.

Material Model 13: Isotropic Elastic-Plastic with Failure

This highly simplistic failure model is occasionally useful. Material model 12 is called to update the stress tensor. Failure is initially assumed to occur if either

$$p^{n+1} < p_{min} \tag{19.13.1}$$

or

$$\varepsilon_{eff}^p > \varepsilon_{\max}^p \tag{19.13.2}$$

where p_{min} and $\varepsilon^p_{\max}$ are user-defined parameters. Once failure has occurred, pressure may never be negative and the deviatoric components are set to zero:

$$s_{ij} = 0 \tag{19.13.3}$$

for all time. The failed element can only carry loads in compression.

Material Model 14: Soil and Crushable Foam With Failure

This material model provides the same stress update as model 5. However, if pressure ever reaches its cutoff value, failure occurs and pressure can never again go negative. In material model 5, the pressure is limited to its cutoff value in tension.

Material Model 15: Johnson and Cook Plasticity Model

Johnson and Cook express the flow stress as

$$\sigma_y = \left(A + B\,\bar{\varepsilon}^{p^n}\right)\left(1 + c\ \ln \dot{\varepsilon}^*\right)\left(1 - T*^m\right) \tag{19.15.1}$$

where A, B, C, n, *and* m are user defined input constants, and:

$\bar{\varepsilon}^p$ = effective plastic strain

$\dot{\varepsilon}* = \dfrac{\dot{\bar{\varepsilon}}^p}{\dot{\varepsilon}_0}$ effective plastic strain rate for $\dot{\varepsilon}_0 = 1\,s^{-1}$

$$T^* = \frac{T - T_{room}}{T_{melt} - T_{room}}$$

Constants for a variety of materials are provided in Johnson and Cook [1983].

Due to the nonlinearity in the dependence of flow stress on plastic strain, an accurate value of the flow stress requires iteration for the increment in plastic strain. However, by using a Taylor series expansion with linearization about the current time, we can solve for σ_y with sufficient accuracy to avoid iteration.

The strain at fracture is given by

$$\varepsilon^f = \left[D_1 + D_2 \exp D_3\,\sigma^*\right]\left[1 + D_4\ \ln \varepsilon^*\right]\left[1 + D_5\,T^*\right] \tag{19.15.2}$$

where D_i, $i=1,\ldots,5$ are input constants and σ^* is the ratio of pressure divided by effective stress:

$$\sigma^* = \frac{p}{\sigma_{eff}} \tag{19.15.3}$$

Fracture occurs when the damage parameter

$$D = \sum \frac{\Delta \bar{\varepsilon}^p}{\varepsilon^f} \tag{19.15.4}$$

reaches the value 1.

A choice of three spall models is offered to represent material splitting, cracking, and failure under tensile loads. The pressure limit model limits the minimum hydrostatic pressure to the specified value, $p \geq p_{min}$. If pressures more tensile than this limit are calculated, the pressure is reset to p_{min}. This option is not strictly a spall model since the deviatoric stresses are unaffected by the pressure reaching the tensile cutoff and the pressure cutoff value p_{min} remains unchanged throughout the analysis. The maximum principal stress spall model detects spall if the maximum principal stress, σ_{max}, exceeds the limiting value σ_p. Once spall is detected with this model, the deviatoric stresses are reset to zero and no hydrostatic tension is permitted. If tensile pressures are calculated, they are reset to 0 in the spalled material. Thus, the spalled material behaves as rubble. The hydrostatic tension spall model detects spall if the pressure becomes more tensile than the specified limit, p_{min}. Once spall is detected, the deviatoric stresses are set to zero and the pressure is required to be compressive. If hydrostatic tension is calculated then the pressure is reset to 0 for that element.

In addition to the above failure criterion, this material model also supports a shell element deletion criterion based on the maximum stable time step size for the element, Δt_{max}. Generally, Δt_{max} goes down as the element becomes more distorted. To assure stability of time integration, the global LS-DYNA time step is the minimum of the Δt_{max} values calculated for all elements in the model. Using this option allows the selective deletion of elements whose time step Δt_{max} has fallen below the specified minimum time step, Δt_{crit}. Elements which are severely distorted often indicate that material has failed and supports little load, but these same elements may have very small time steps and therefore control the cost of the analysis. This option allows these highly distorted elements to be deleted from the calculation, and, therefore, the analysis can proceed at a larger time step, and, thus, at a reduced cost. Deleted elements do not carry any load, and are deleted from all applicable slide surface definitions. Clearly, this option must be judiciously used to obtain accurate results at a minimum cost.

Material type 15 is applicable to the high rate deformation of many materials including most metals. Unlike the Steinberg-Guinan model, the Johnson-Cook model remains valid down to lower strain rates and even into the quasistatic regime. Typical applications include explosive metal forming, ballistic penetration, and impact.

Material Model 16: Pseudo Tensor

This model can be used in two major modes - a simple tabular pressure-dependent yield surface, and a potentially complex model featuring two yield versus pressure functions with the means of migrating from one curve to the other. For both modes, load curve N1 is taken to be a strain rate multiplier for the yield strength. Note that this model must be used with equation-of-state type 8 or 9.

Response Mode I. Tabulated Yield Stress Versus Pressure

This model is well suited for implementing standard geologic models like the Mohr-Coulomb yield surface with a Tresca limit, as shown in Figure 19.16.1. Examples of converting conventional triaxial compression data to this type of model are found in (Desai and Siriwardane, 1984). Note that under conventional triaxial compression conditions, the LS-DYNA input corresponds to an ordinate of $\sigma_1 - \sigma_3$ rather than the more widely used $\frac{\sigma_1 - \sigma_3}{2}$, where σ_1 is the maximum principal stress and σ_3 is the minimum principal stress.

This material combined with equation-of-state type 9 (saturated) has been used very successfully to model ground shocks and soil-structure interactions at pressures up to 100kbar.

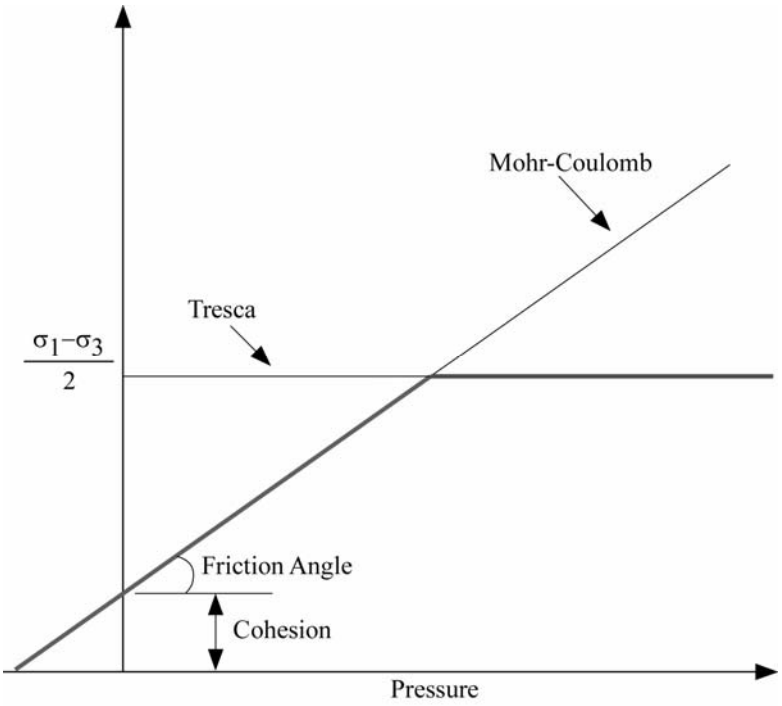


Figure 19.16.1. Mohr-Coulomb surface with a Tresca limit.

To invoke Mode I of this model, set a_0, a_1, a_2, b_1, a_{0f}, and a_{1f} to zero. The tabulated values of pressure should then be specified on cards 4 and 5, and the corresponding values of yield stress should be specified on cards 6 and 7. The parameters relating to reinforcement properties, initial yield stress, and tangent modulus are not used in this response mode, and should be set to zero.

Simple tensile failure

Note that a_{1f} is reset internally to 1/3 even though it is input as zero; this defines a material failure curve of slope 3*p*, where *p* denotes pressure (positive in compression). In this case the yield strength is taken from the tabulated yield vs. pressure curve until the maximum principal stress (σ_1) in the element exceeds the tensile cut-off (σ_{cut}). For every time step that

$\sigma_1 > \sigma_{cut}$ the yield strength is scaled back by a fraction of the distance between the two curves until after 20 time steps the yield strength is defined by the failure curve. The only way to inhibit this feature is to set σ_{cut} arbitrarily large.

Response Mode II. Two-Curve Model with Damage and Failure

This approach uses two yield versus pressure curves of the form

$$\sigma_y = a_0 + \frac{p}{a_1 + a_2 p} \tag{19.16.1}$$

The upper curve is best described as the maximum yield strength curve and the lower curve is the material failure curve. There are a variety of ways of moving between the two curves and each is discussed below.

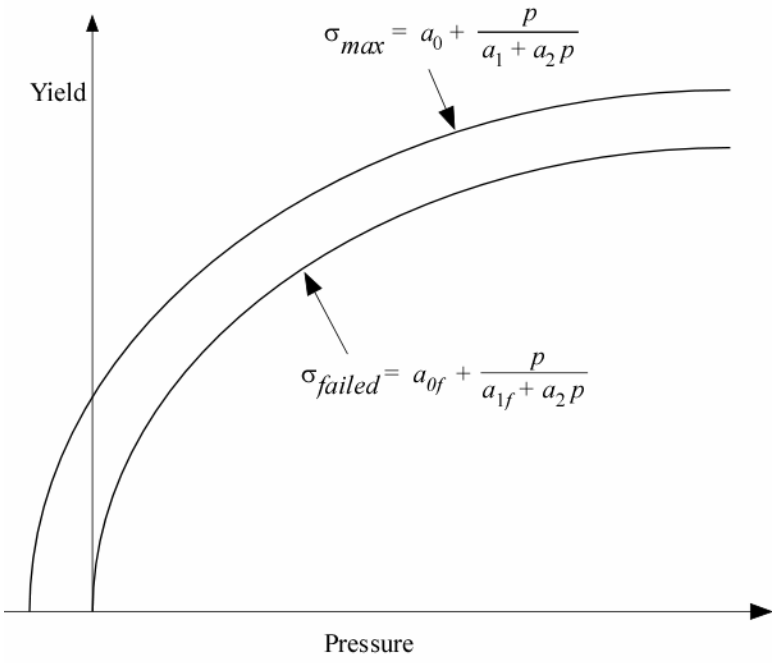


Figure 19.16.2. Two-curve concrete model with damage and failure.

MODE II.A: Simple tensile failure

Define a_0, a_1, a_2, a_{0f} and a_{1f}, set b_1 to zero, and leave cards 4 through 7 blank. In this case the yield strength is taken from the maximum yield curve until the maximum principal stress (σ_1) in the element exceeds the tensile cut-off (σ_{cut}). For every time step that $\sigma_1 > \sigma_{cut}$ the yield strength is scaled back by a fraction of the distance between the two curves until after 20 time steps the yield strength is defined by the failure curve.

Mode II.B: Tensile failure plus plastic strain scaling

Define a_0, a_1, a_2, a_{0f} and a_{1f}, set b_1 to zero, and user cards 4 through 7 to define a scale factor, η, versus effective plastic strain. LS-DYNA evaluates η at the current effective plastic strain and then calculated the yield stress as

$$\sigma_{yield} = \sigma_{failed} + \eta(\sigma_{\max} - \sigma_{failed}) \tag{19.16.2}$$

where $\sigma_{\max}$ and σ_{failed} are found as shown in Figure 19.16.2. This yield strength is then subject to scaling for tensile failure as described above. This type of model allows the description of a strain hardening or softening material such as concrete.

Mode II.C: Tensile failure plus damage scaling

The change in yield stress as a function of plastic strain arises from the physical mechanisms such as internal cracking, and the extent of this cracking is affected by the hydrostatic pressure when the cracking occurs. This mechanism gives rise to the "confinement" effect on concrete behavior. To account for this phenomenon, a "damage" function was defined and incorporated. This damage function is given the form:

$$\lambda = \int_0^{\varepsilon^p} \left(1 + \frac{p}{\sigma_{cut}}\right)^{-b_1} d\varepsilon^p \tag{19.16.3}$$

Define a_0, a_1, a_2, a_{0f} and a_{1f}, and b_1. Cards 4 though 7 now give η as a function of λ and scale the yield stress as

$$\sigma_{yield} = \sigma_{failed} + \eta\left(\sigma_{\max} - \sigma_{failed}\right) \tag{19.16.4}$$

and then apply any tensile failure criteria.

Mode II Concrete Model Options

Material Type 16 Mode II provides the option of automatic internal generation of a simple "generic" model for concrete. If a_0 is negative, then σ_{cut} is assumed to be the unconfined concrete compressive strength, f_c' and $-a_0$ is assumed to be a conversion factor from LS-DYNA pressure units to psi. (For example, if the model stress units are MPa, a_0 should be set to –145.) In this case the parameter values generated internally are

$$\sigma_{cut} = 1.7\left(\frac{f_c'^2}{-a_0}\right)^{\frac{1}{3}}$$
$$a_0 = \frac{f_c'}{4}$$
$$a_1 = \frac{1}{3} \qquad (19.16.5)$$
$$a_2 = \frac{1}{3f_c'}$$
$$a_{0f} = 0$$
$$a_{1f} = 0.385$$

Note that these a_{0f} and a_{1f} defaults will be overwritten by non-zero entries on Card 3. If plastic strain or damage scaling is desired, Cards 5 through 8 and b_1 should be specified in the input. When a_0 is input as a negative quantity, the equation-of-state can be given as 0 and a trilinear EOS Type 8 model will be automatically generated from the unconfined compressive strength and Poisson's ratio. The EOS 8 model is a simple pressure versus volumetric strain model with no internal energy terms, and should give reasonable results for pressures up to 5kbar (approximately 72,500 psi).

Mixture model

A reinforcement fraction, f_r, can be defined along with properties of the reinforcing material. The bulk modulus, shear modulus, and yield strength are then calculated from a simple mixture rule, i.e., for the bulk modulus the rule gives:

$$K = (1 - f_r)K_m + f_r K_r \qquad (19.16.6)$$

where K_m and K_r are the bulk moduli for the geologic material and the reinforcing material, respectively. This feature should be used with caution. It gives an isotropic effect in the material instead of the true anisotropic material behavior. A reasonable approach would be to use the mixture elements only where reinforcing material exists and plain elements elsewhere. When the mixture model is being used, the strain rate multiplier for the principal material is taken from load curve N1 and the multiplier for the reinforcement is taken from load curve N2.

Material Model 17: Isotropic Elastic-Plastic With Oriented Cracks

This is an isotropic elastic-plastic material which includes a failure model with an oriented crack. The von Mises yield condition is given by:

$$\phi = J_2 - \frac{\sigma_y^2}{3} \qquad (19.17.1)$$

where the second stress invariant, J_2, is defined in terms of the deviatoric stress components as

$$J_2 = \frac{1}{2} s_{ij} s_{ij} \tag{19.17.2}$$

and the yield stress, σ_y, is a function of the effective plastic strain, ε_{eff}^p, and the plastic hardening modulus, E_p:

$$\sigma_y = \sigma_0 + E_p \varepsilon_{eff}^p \tag{19.17.3}$$

The effective plastic strain is defined as:

$$\varepsilon_{eff}^p = \int_0^t d\varepsilon_{eff}^p \tag{19.17.4}$$

where $$d\varepsilon_{eff}^p = \sqrt{\frac{2}{3} d\varepsilon_{ij}^p d\varepsilon_{ij}^p}$$

and the plastic tangent modulus is defined in terms of the input tangent modulus, E_t, as

$$E_p = \frac{E E_t}{E - E_t} \tag{19.17.5}$$

Pressure in this model is found from evaluating an equation of state. A pressure cutoff can be defined such that the pressure is not allowed to fall below the cutoff value.

The oriented crack fracture model is based on a maximum principal stress criterion. When the maximum principal stress exceeds the fracture stress, σ_f, the element fails on a plane perpendicular to the direction of the maximum principal stress. The normal stress and the two shear stresses on that plane are then reduced to zero. This stress reduction is done according to a delay function that reduces the stresses gradually to zero over a small number of time steps. This delay function procedure is used to reduce the ringing that may otherwise be introduced into the system by the sudden fracture.

After a tensile fracture, the element will not support tensile stress on the fracture plane, but in compression will support both normal and shear stresses. The orientation of this fracture surface is tracked throughout the deformation, and is updated to properly model finite deformation effects. If the maximum principal stress subsequently exceeds the fracture stress in another direction, the element fails isotropically. In this case the element completely loses its ability to support any shear stress or hydrostatic tension, and only compressive hydrostatic stress states are possible. Thus, once isotropic failure has occurred, the material behaves like a fluid.

This model is applicable to elastic or elastoplastic materials under significant tensile or shear loading when fracture is expected. Potential applications include brittle materials such as ceramics as well as porous materials such as concrete in cases where pressure hardening effects are not significant.

Material Model 18: Power Law Isotropic Plasticity

Elastoplastic behavior with isotropic hardening is provided by this model. The yield stress, σ_y, is a function of plastic strain and obeys the equation:

$$\sigma_y = k\,\varepsilon^n = k\left(\varepsilon_{yp} + \bar{\varepsilon}^p\right)^n \tag{19.18.1}$$

where ε_{yp} is the elastic strain to yield and $\bar{\varepsilon}^p$ is the effective plastic strain (logarithmic).

A parameter, SIGY, in the input governs how the strain to yield is identified. If SIGY is set to zero, the strain to yield if found by solving for the intersection of the linearly elastic loading equation with the strain hardening equation:

$$\begin{aligned} \sigma &= E\,\varepsilon \\ \sigma &= k\,\varepsilon^n \end{aligned} \tag{19.18.2}$$

which gives the elastic strain at yield as:

$$\varepsilon_{yp} = \left(\frac{E}{k}\right)^{\left[\frac{1}{n-1}\right]} \tag{19.18.3}$$

If SIGY yield is nonzero and greater than 0.02 then:

$$\varepsilon_{yp} = \left(\frac{\sigma_y}{k}\right)^{\left[\frac{1}{n}\right]} \tag{19.18.4}$$

Strain rate is accounted for using the Cowper-Symonds model which scales the yield stress with the factor

$$1 + \left(\frac{\dot{\varepsilon}}{C}\right)^{1/p} \tag{19.18.5}$$

where $\dot{\varepsilon}$ is the strain rate. A fully viscoplastic formulation is optional with this model which incorporates the Cowper-Symonds formulation within the yield surface. An additional cost is incurred but the improvement allows for dramatic results.

Material Model 19: Strain Rate Dependent Isotropic Plasticity

In this model, a load curve is used to describe the yield strength σ_0 as a function of effective strain rate $\dot{\bar{\varepsilon}}$ where

$$\dot{\bar{\varepsilon}} = \left(\frac{2}{3}\dot{\varepsilon}'_{ij}\,\dot{\varepsilon}'_{ij}\right)^{1/2} \tag{19.19.1}$$

and the prime denotes the deviatoric component. The yield stress is defined as

$$\sigma_y = \sigma_0 \left(\dot{\bar{\varepsilon}} \right) + E_p \, \bar{\varepsilon}^p \qquad (19.19.2)$$

where $\bar{\varepsilon}^p$ is the effective plastic strain and E_h is given in terms of Young's modulus and the tangent modulus by

$$E_p = \frac{E \, E_t}{E - E_t} \ . \qquad (19.19.3)$$

Both Young's modulus and the tangent modulus may optionally be made functions of strain rate by specifying a load curve ID giving their values as a function of strain rate. If these load curve ID's are input as 0, then the constant values specified in the input are used.

Note that all load curves used to define quantities as a function of strain rate must have the same number of points at the same strain rate values. This requirement is used to allow vectorized interpolation to enhance the execution speed of this constitutive model.

This model also contains a simple mechanism for modeling material failure. This option is activated by specifying a load curve ID defining the effective stress at failure as a function of strain rate. For solid elements, once the effective stress exceeds the failure stress the element is deemed to have failed and is removed from the solution. For shell elements the entire shell element is deemed to have failed if all integration points through the thickness have an effective stress that exceeds the failure stress. After failure the shell element is removed from the solution.

In addition to the above failure criterion, this material model also supports a shell element deletion criterion based on the maximum stable time step size for the element, Δt_{max}. Generally, Δt_{max} goes down as the element becomes more distorted. To assure stability of time integration, the global LS-DYNA time step is the minimum of the Δt_{max} values calculated for all elements in the model. Using this option allows the selective deletion of elements whose time step Δt_{max} has fallen below the specified minimum time step, Δt_{crit}. Elements which are severely distorted often indicate that material has failed and supports little load, but these same elements may have very small time steps and therefore control the cost of the analysis. This option allows these highly distorted elements to be deleted from the calculation, and, therefore, the analysis can proceed at a larger time step, and, thus, at a reduced cost. Deleted elements do not carry any load, and are deleted from all applicable slide surface definitions. Clearly, this option must be judiciously used to obtain accurate results at a minimum cost.

Material Type 20: Rigid

The rigid material type 20 provides a convenient way of turning one or more parts comprised of beams, shells, or solid elements into a rigid body. Approximating a deformable body as rigid is a preferred modeling technique in many real world applications. For example, in sheet metal forming problems the tooling can properly and accurately be treated as rigid. In the design of restraint systems the occupant can, for the purposes of early design studies, also be treated as rigid. Elements which are rigid are bypassed in the element processing and no storage

is allocated for storing history variables; consequently, the rigid material type is very cost efficient.

Two unique rigid part IDs may not share common nodes unless they are merged together using the rigid body merge option. A rigid body may be made up of disjoint finite element meshes, however. LS-DYNA assumes this is the case since this is a common practice in setting up tooling meshes in forming problems.

All elements which reference a given part ID corresponding to the rigid material should be contiguous, but this is not a requirement. If two disjoint groups of elements on opposite sides of a model are modeled as rigid, separate part ID's should be created for each of the contiguous element groups if each group is to move independently. This requirement arises from the fact that LS-DYNA internally computes the six rigid body degrees-of-freedom for each rigid body (rigid material or set of merged materials), and if disjoint groups of rigid elements use the same part ID, the disjoint groups will move together as one rigid body.

Inertial properties for rigid materials may be defined in either of two ways. By default, the inertial properties are calculated from the geometry of the constituent elements of the rigid material and the density specified for the part ID. Alternatively, the inertial properties and initial velocities for a rigid body may be directly defined, and this overrides data calculated from the material property definition and nodal initial velocity definitions.

Young's modulus, E, and Poisson's ratio, υ are used for determining sliding interface parameters if the rigid body interacts in a contact definition. Realistic values for these constants should be defined since unrealistic values may contribute to numerical problem in contact.

Material Model 21: Thermal Orthotropic Elastic

In the implementation for three-dimensional continua a total Lagrangian formulation is used. In this approach the material law that relates second Piola-Kirchhoff stress S to the Green-St. Venant strain E is

$$S = C \cdot E = T^t C_l T \cdot E \tag{19.21.1}$$

where T is the transformation matrix [Cook 1974].

$$T = \begin{bmatrix} l_1^2 & m_1^2 & n_1^2 & l_1 m_1 & m_1 n_1 & n_1 l_1 \\ l_2^2 & m_2^2 & n_2^2 & l_2 m_2 & m_2 n_2 & n_2 l_2 \\ l_3^2 & m_3^2 & n_3^2 & l_3 m_3 & m_3 n_3 & n_3 l_3 \\ 2l_1 l_2 & 2m_1 m_2 & 2n_1 n_2 & (l_1 m_2 + l_2 m_1) & (m_1 n_2 + m_2 n_1) & (n_1 l_2 + n_2 l_1) \\ 2l_2 l_3 & 2m_2 m_3 & 2n_2 n_3 & (l_2 m_3 + l_3 m_2) & (m_2 n_3 + m_3 n_2) & (n_2 l_3 + n_3 l_2) \\ 2l_3 l_1 & 2m_3 m_1 & 2n_3 n_1 & (l_3 m_1 + l_1 m_3) & (m_3 n_1 + m_1 n_3) & (n_3 l_1 + n_1 l_3) \end{bmatrix} \tag{19.21.2}$$

l_i, m_i, n_i are the direction cosines

$$x_i^{'} = l_i x_1 + m_i x_2 + n_i x_3 \qquad for\ i = 1, 2, 3 \tag{19.21.3}$$

and $x_i^{'}$ denotes the material axes. The constitutive matrix C_l is defined in terms of the material axes as

$$C_l^{-1} = \begin{bmatrix} \frac{1}{E_{11}} & -\frac{\upsilon_{21}}{E_{22}} & -\frac{\upsilon_{31}}{E_{33}} & 0 & 0 & 0 \\ -\frac{\upsilon_{12}}{E_{11}} & \frac{1}{E_{22}} & -\frac{\upsilon_{32}}{E_{33}} & 0 & 0 & 0 \\ -\frac{\upsilon_{13}}{E_{11}} & -\frac{\upsilon_{23}}{E_{22}} & \frac{1}{E_{33}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{12}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{23}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{31}} \end{bmatrix} \tag{19.21.4}$$

where the subscripts denote the material axes, i.e.,

$$\upsilon_{ij} = \upsilon_{x_i' x_j'} \quad \text{and} \quad E_{ii} = E_{x_i'} \tag{19.21.5}$$

Since C_l is symmetric

$$\frac{\upsilon_{12}}{E_{11}} = \frac{\upsilon_{21}}{E_{22}}, \text{ etc.} \tag{19.21.6}$$

The vector of Green-St. Venant strain components is

$$E^t = \lfloor E_{11}, E_{22}, E_{33}, E_{12}, E_{23}, E_{31}, \rfloor \tag{19.21.7}$$

which include the local thermal strains which are integrated in time:

$$\begin{aligned} \varepsilon_{aa}^{n+1} &= \varepsilon_{aa}^{n} + \alpha_a \left(T^{n+1} - T^n \right) \\ \varepsilon_{bb}^{n+1} &= \varepsilon_{bb}^{n} + \alpha_b \left(T^{n+1} - T^n \right) \\ \varepsilon_{cc}^{n+1} &= \varepsilon_{cc}^{n} + \alpha_c \left(T^{n+1} - T^n \right) \end{aligned} \tag{19.21.8}$$

After computing S_{ij} we use Equation (18.32) to obtain the Cauchy stress. This model will predict realistic behavior for finite displacement and rotations as long as the strains are small.

For shell elements, the stresses are integrated in time and are updated in the corotational coordinate system. In this procedure the local material axes are assumed to remain orthogonal in the deformed configuration. This assumption is valid if the strains remain small.

Material Model 22: Chang-Chang Composite Failure Model

Five material parameters are used in the three failure criteria. These are [Chang and Chang 1987a, 1987b]:

- S_1, longitudinal tensile strength
- S_2, transverse tensile strength
- S_{12}, shear strength
- C_2, transverse compressive strength
- α, nonlinear shear stress parameter.

S_1, S_2, S_{12}, and C_2 are obtained from material strength measurement. α is defined by material shear stress-strain measurements. In plane stress, the strain is given in terms of the stress as

$$\begin{aligned} \varepsilon_1 &= \frac{1}{E_1}(\sigma_1 - \upsilon_1\sigma_2) \\ \varepsilon_2 &= \frac{1}{E_2}(\sigma_2 - \upsilon_2\sigma_1) \\ 2\varepsilon_{12} &= \frac{1}{G_{12}}\tau_{12} + \alpha\tau_{12}^3 \end{aligned} \qquad (19.22.1)$$

The third equation defines the nonlinear shear stress parameter α.

A fiber matrix shearing term augments each damage mode:

$$\bar{\tau} = \frac{\frac{\tau_{12}^2}{2G_{12}} + \frac{3}{4}\alpha\tau_{12}^4}{\frac{S_{12}^2}{2G_{12}} + \frac{3}{4}\alpha S_{12}^4} \qquad (19.22.2)$$

which is the ratio of the shear stress to the shear strength.

The matrix cracking failure criteria is determined from

$$F_{\text{matrix}} = \left(\frac{\sigma_2}{S_2}\right)^2 + \bar{\tau} \qquad (19.22.3)$$

where failure is assumed whenever $F_{\text{matrix}} > 1$. If $F_{\text{matrix}} > 1$, then the material constants E_2, G_{12}, υ_1, and υ_2 are set to zero.

The compression failure criteria is given as

$$F_{comp} = \left(\frac{\sigma_2}{2S_{12}} \right)^2 + \left[\left(\frac{C_2}{2S_{12}} \right)^2 - 1 \right] \frac{\sigma_2}{C_2} + \overline{\tau} \qquad (19.22.4)$$

where failure is assumed whenever $F_{comb} > 1$. If $F_{comb} > 1$, then the material constants E_2, υ_1, and υ_2 are set to zero.

The final failure mode is due to fiber breakage.

$$F_{fiber} = \left(\frac{\sigma_1}{S_1} \right)^2 + \overline{\tau} \qquad (19.22.5)$$

Failure is assumed whenever $F_{fiber} > 1$. If $F_{fiber} > 1$, then the constants E_1, E_2, G_{12}, υ_1, and υ_2 are set to zero.

Material Model 23: Thermal Orthotropic Elastic with 12 Curves

In the implementation for three-dimensional continua a total Lagrangian formulation is used. In this approach the material law that relates second Piola-Kirchhoff stress S to the Green-St. Venant strain E is

$$S = C \cdot E = T^t C_l T \cdot E \qquad (19.23.1)$$

where T is the transformation matrix [Cook 1974].

$$T = \begin{bmatrix} l_1^2 & m_1^2 & n_1^2 & l_1 m_1 & m_1 n_1 & n_1 l_1 \\ l_2^2 & m_2^2 & n_2^2 & l_2 m_2 & m_2 n_2 & n_2 l_2 \\ l_3^2 & m_3^2 & n_3^2 & l_3 m_3 & m_3 n_3 & n_3 l_3 \\ 2l_1 l_2 & 2m_1 m_2 & 2n_1 n_2 & (l_1 m_2 + l_2 m_1) & (m_1 n_2 + m_2 n_1) & (n_1 l_2 + n_2 l_1) \\ 2l_2 l_3 & 2m_2 m_3 & 2n_2 n_3 & (l_2 m_3 + l_3 m_2) & (m_2 n_3 + m_3 n_2) & (n_2 l_3 + n_3 l_2) \\ 2l_3 l_1 & 2m_3 m_1 & 2n_3 n_1 & (l_3 m_1 + l_1 m_3) & (m_3 n_1 + m_1 n_3) & (n_3 l_1 + n_1 l_3) \end{bmatrix} \qquad (19.23.2)$$

l_i, m_i, n_i are the direction cosines

$$x_i^{'} = l_i x_1 + m_i x_2 + n_i x_3 \qquad for\ i = 1, 2, 3 \qquad (19.23.3)$$

and $x_i^{'}$ denotes the material axes. The temperature dependent constitutive matrix C_l is defined in terms of the material axes as

$$C_l^{-1} = \begin{bmatrix} \frac{1}{E_{11}(T)} & -\frac{\upsilon_{21}(T)}{E_{22}(T)} & -\frac{\upsilon_{31}(T)}{E_{33}(T)} & 0 & 0 & 0 \\ -\frac{\upsilon_{12}(T)}{E_{11}(T)} & \frac{1}{E_{22}(T)} & -\frac{\upsilon_{32}(T)}{E_{33}(T)} & 0 & 0 & 0 \\ -\frac{\upsilon_{13}(T)}{E_{11}(T)} & -\frac{\upsilon_{23}(T)}{E_{22}(T)} & \frac{1}{E_{33}(T)} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{12}(T)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{23}(T)} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{31}(T)} \end{bmatrix} \quad (19.23.4)$$

where the subscripts denote the material axes, i.e.,

$$\upsilon_{ij} = \upsilon_{x_i' x_j'} \quad \text{and} \quad E_{ii} = E_{x_i'} \quad (19.23.5)$$

Since C_l is symmetric

$$\frac{\upsilon_{12}}{E_{11}} = \frac{\upsilon_{21}}{E_{22}}, \text{ etc.} \quad (19.23.6)$$

The vector of Green-St. Venant strain components is

$$E^t = \lfloor E_{11}, E_{22}, E_{33}, E_{12}, E_{23}, E_{31}, \rfloor \quad (19.23.7)$$

which include the local thermal strains which are integrated in time:

$$\begin{aligned} \varepsilon_{aa}^{n+1} &= \varepsilon_{aa}^{n} + \alpha_a (T^{n+\frac{1}{2}}) \left[T^{n+1} - T^n \right] \\ \varepsilon_{bb}^{n+1} &= \varepsilon_{bb}^{n} + \alpha_b (T^{n+\frac{1}{2}}) \left[T^{n+1} - T^n \right] \\ \varepsilon_{cc}^{n+1} &= \varepsilon_{cc}^{n} + \alpha_c (T^{n+\frac{1}{2}}) \left[T^{n+1} - T^n \right] \end{aligned} \quad (19.23.8)$$

After computing S_{ij} we use Equation (16.32) to obtain the Cauchy stress. This model will predict realistic behavior for finite displacement and rotations as long as the strains are small.

For shell elements, the stresses are integrated in time and are updated in the corotational coordinate system. In this procedure the local material axes are assumed to remain orthogonal in the deformed configuration. This assumption is valid if the strains remain small.

Material Model 24: Piecewise Linear Isotropic Plasticity

This plasticity treatment in this model is quite similar to Model 10, but unlike 10, it includes strain rate effects and does not use an equation of state. Deviatoric stresses are determined that satisfy the yield function

$$\phi = \frac{1}{2} s_{ij}\, s_{ij} - \frac{\sigma_y^{\,2}}{3} \le 0 \qquad (19.10.1)$$

where

$$\sigma_y = \beta\left[\sigma_0 + f_h\left(\varepsilon_{eff}^p\right)\right] \qquad (19.24.1)$$

where the hardening function $f_h\left(\varepsilon_{eff}^{\,p}\right)$ can be specified in tabular form as an option. Otherwise, linear hardening of the form

$$f_h\left(\varepsilon_{eff}^p\right) = E_p\left(\varepsilon_{eff}^p\right) \qquad (19.10.3)$$

is assumed where E_p and ε_{eff}^p are given in Equations (19.3.6) and (19.3.7), respectively. The parameter β accounts for strain rate effects. For complete generality a table defining the yield stress versus plastic strain may be defined for various levels of effective strain rate.

In the implementation of this material model, the deviatoric stresses are updated elastically (see material model 1), the yield function is checked, and if it is satisfied the deviatoric stresses are accepted. If it is not, an increment in plastic strain is computed:

$$\Delta\varepsilon_{eff}^p = \frac{\left(\frac{3}{2} s_{ij}^*\, s_{ij}^*\right)^{1/2} - \sigma_y}{3G + E_p} \qquad (19.10.4)$$

is the shear modulus and E_p is the current plastic hardening modulus. The trial deviatoric stress state s_{ij}^* is scaled back:

$$s_{ij}^{n+1} = \frac{\sigma_y}{\left(\frac{3}{2} s_{ij}^*\, s_{ij}^*\right)^{1/2}} s_{ij}^* \qquad (19.10.5)$$

For shell elements, the above equations apply, but with the addition of an iterative loop to solve for the normal strain increment, such that the stress component normal to the mid surface of the shell element approaches zero.

Three options to account for strain rate effects are possible:

I. Strain rate may be accounted for using the Cowper-Symonds model which scales the yield stress with the factor

$$\beta = 1 + \left(\frac{\dot{\varepsilon}}{C} \right)^{1/p} \tag{19.24.2}$$

where $\dot{\varepsilon}$ is the strain rate.

II. For complete generality a load curve, defining β, which scales the yield stress may be input instead. In this curve the scale factor versus strain rate is defined.

III. If different stress versus strain curves can be provided for various strain rates, the option using the reference to a table definition can be used. See Figure 19.24.1.

A fully viscoplastic formulation is optional which incorporates the different options above within the yield surface. An additional cost is incurred over the simple scaling but the improvement is results can be dramatic.

If a table ID is specified a curve ID is given for each strain rate, see Section 23. Intermediate values are found by interpolating between curves. Effective plastic strain versus yield stress is expected. If the strain rate values fall out of range, extrapolation is not used; rather, either the first or last curve determines the yield stress depending on whether the rate is low or high, respectively.

Material Model 25: Kinematic Hardening Cap Model

The implementation of an extended two invariant cap model, suggested by Stojko [1990], is based on the formulations of Simo, et al. [1988, 1990] and Sandler and Rubin [1979]. In this model, the two invariant cap theory is extended to include nonlinear kinematic hardening as suggested by Isenberg, Vaughn, and Sandler [1978]. A brief discussion of the extended cap model and its parameters is given below.

The cap model is formulated in terms of the invariants of the stress tensor. The square root of the second invariant of the deviatoric stress tensor, $\sqrt{J_{2D}}$ is found from the deviatoric stresses **s** as

$$\sqrt{J_{2D}} \equiv \sqrt{\frac{1}{2} s_{ij} s_{ij}}$$

and is the objective scalar measure of the distortional or shearing stress. The first invariant of the stress, J_1, is the trace of the stress tensor.

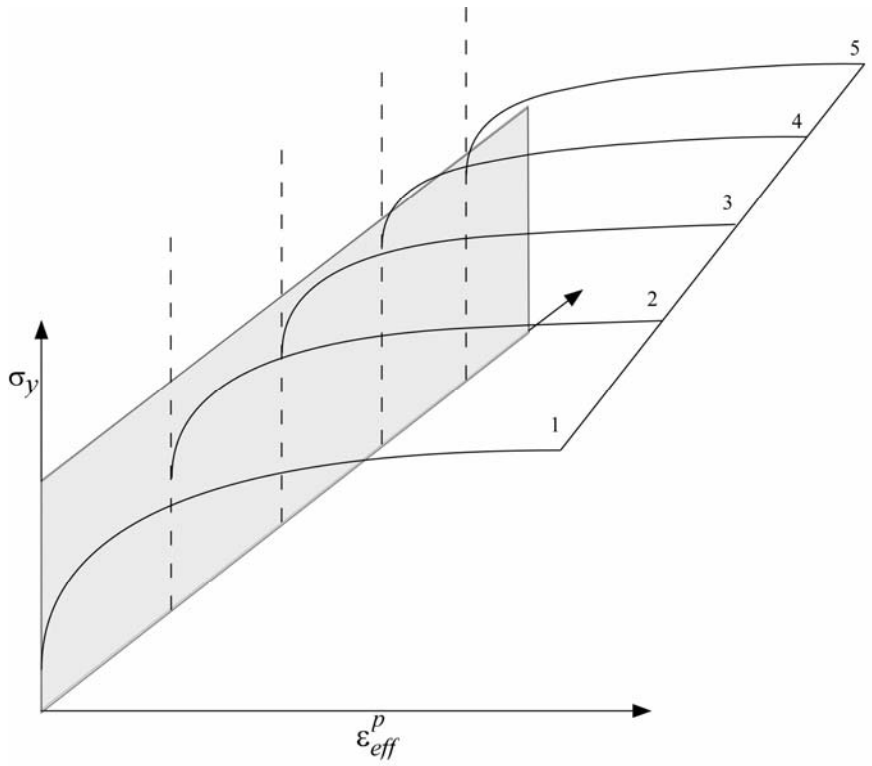


Figure 19.25.1. Rate effects may be accounted for by defining a table of curves.

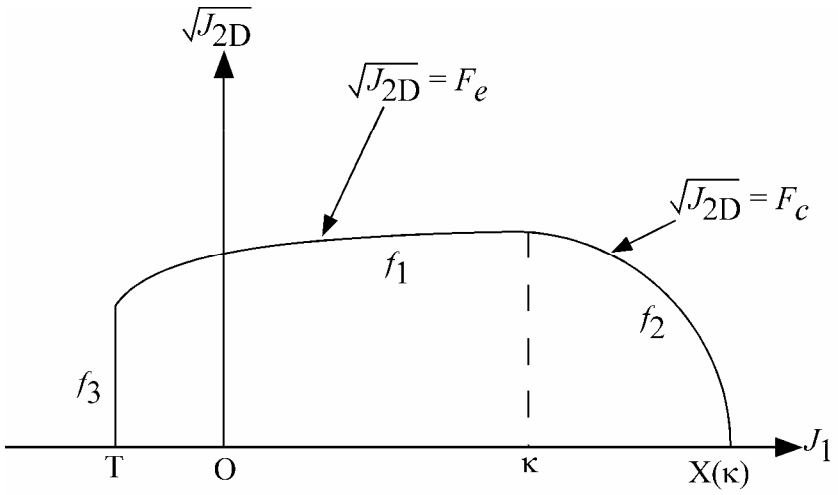


Figure 19.25.2. The yield surface of the two-invariant cap model in pressure $\sqrt{J_{2D}} - J_1$ space Surface f_1 is the failure envelope, f_2 is the cap surface, and f_3 is the tension cutoff.

The cap model consists of three surfaces in $\sqrt{J_{2D}} - J_1$ space, as shown in Figure 19.25.1. First, there is a failure envelope surface, denoted f_1 in the figure. The functional form of f_1 is

$$f_1 = \sqrt{J_{2D}} - \min\left(F_e\left(J_1\right), T_{mises}\right),$$

where F_e is given by

$$F_e\left(J_1\right) \equiv \alpha - \gamma \exp\left(-\beta J_1\right) + \theta J_1$$

and $T_{mises} \equiv \left|X\left(\kappa_n\right) - L\left(\kappa_n\right)\right|$. This failure envelope surface is fixed in $\sqrt{J_{2D}} - J_1$ space, and therefore does not harden unless kinematic hardening is present. Next, there is a cap surface, denoted f_2 in the figure, with f_2 given by

$$f_2 = \sqrt{J_{2D}} - F_c\left(J_1, \kappa\right)$$

where F_c is defined by

$$F_c\left(J_1, \kappa\right) \equiv \frac{1}{R}\sqrt{\left[X\left(\kappa\right) - L\left(\kappa\right)\right]^2 - \left[J_1 - L\left(\kappa\right)\right]^2},$$

$X\left(\kappa\right)$ is the intersection of the cap surface with the J_1 axis

$$X\left(\kappa\right) = \kappa + RF_e\left(\kappa\right),$$

and $L\left(\kappa\right)$ is defined by

$$L\left(\kappa\right) \equiv \begin{cases} \kappa \; if \; \kappa > 0 \\ 0 \; if \; \kappa \le 0 \end{cases}.$$

The hardening parameter κ is related to the plastic volume change ε_v^p through the hardening law

$$\varepsilon_v^p = W\left\{1 - \exp\left[-D\left(X\left(\kappa\right) - X_0\right)\right]\right\}$$

Geometrically, κ is seen in the figure as the J_1 coordinate of the intersection of the cap surface and the failure surface. Finally, there is the tension cutoff surface, denoted f_3 in the figure. The function f_3 is given by

$$f_3 + T - J_1,$$

where T is the input material parameter which specifies the maximum hydrostatic tension sustainable by the material. The elastic domain in $\sqrt{J_{2D}} - J_1$ space is then bounded by the failure envelope surface above, the tension cutoff surface on the left, and the cap surface on the right.

An additive decomposition of the strain into elastic and plastic parts is assumed:

$$\varepsilon = \varepsilon + \varepsilon^P ,$$

where ε^e is the elastic strain and ε^p is the plastic strain. Stress is found from the elastic strain using Hooke's law,

$$\sigma = C\left(\varepsilon - \varepsilon^P\right),$$

where σ is the stress and C is the elastic constitutive tensor.

The yield condition may be written

$$f_1(\sigma) \le 0$$
$$f_2(\sigma, \kappa) \le 0$$
$$f_3(\sigma) \le 0$$

and the plastic consistency condition requires that

$$\dot{\lambda}_k f_k = 0$$
$$k = 1, 2, 3$$
$$\dot{\lambda}_k \ge 0$$

where λ_k is the plastic consistency parameter for surface k. If $f_k < 0$, then $\dot{\lambda}_k = 0$ and the response is elastic. If $f_k > 0$, then surface k is active and $\dot{\lambda}_k$ is found from the requirement that $\dot{f}_k = 0$.

Associated plastic flow is assumed, so using Koiter's flow rule the plastic strain rate is given as the sum of contribution from all of the active surfaces,

$$\dot{\varepsilon}^p = \sum_{k=1}^{3} \dot{\lambda}_k \frac{\partial f_k}{\partial s}$$

One of the major advantages of the cap model over other classical pressure-dependent plasticity models is the ability to control the amount of dilatency produced under shear loading. Dilatency is produced under shear loading as a result of the yield surface having a positive slope in $\sqrt{J_{2D}} - J_1$ space, so the assumption of plastic flow in the direction normal to the yield surface produces a plastic strain rate vector that has a component in the volumetric (hydrostatic)

direction (see Figure 19.25.1). In models such as the Drucker-Prager and Mohr-Coulomb, this dilatency continues as long as shear loads are applied, and in many cases produces far more dilatency than is experimentally observed in material tests. In the cap model, when the failure surface is active, dilatency is produced just as with the Drucker-Prager and Mohr-Columb models. However, the hardening law permits the cap surface to contract until the cap intersects the failure envelope at the stress point, and the cap remains at that point. The local normal to the yield surface is now vertical, and therefore the normality rule assures that no further plastic volumetric strain (dilatency) is created. Adjustment of the parameters that control the rate of cap contractions permits experimentally observed amounts of dilatency to be incorporated into the cap model, thus producing a constitutive law which better represents the physics to be modeled.

Another advantage of the cap model over other models such as the Drucker-Prager and Mohr-Coulomb is the ability to model plastic compaction. In these models all purely volumetric response is elastic. In the cap model, volumetric response is elastic until the stress point hits the cap surface. Therefore, plastic volumetric strain (compaction) is generated at a rate controlled by the hardening law. Thus, in addition to controlling the amount of dilatency, the introduction of the cap surface adds another experimentally observed response characteristic of geological material into the model.

The inclusion of kinematic hardening results in hysteretic energy dissipation under cyclic loading conditions. Following the approach of Isenberg, et al., [1978] a nonlinear kinematic hardening law is used for the failure envelope surface when nonzero values of and N are specified. In this case, the failure envelope surface is replaced by a family of yield surfaces bounded by an initial yield surface and a limiting failure envelope surface. Thus, the shape of the yield surfaces described above remains unchanged, but they may translate in a plane orthogonal to the J axis.

Translation of the yield surfaces is permitted through the introduction of a "back stress" tensor, α. The formulation including kinematic hardening is obtained by replacing the stress σ with the translated stress tensor $\eta \equiv \sigma - \alpha$ in all of the above equation. The history tensor α is assumed deviatoric, and therefore has only 5 unique components. The evolution of the back stress tensor is governed by the nonlinear hardening law

$$\overset{\nabla}{\alpha} = \bar{c}\bar{F}\left(\sigma,\alpha\right)\dot{\varepsilon}^{p}$$

where $\bar{c}$ is a constant, $\bar{F}$ is a scalar function of σ and α and $\dot{\varepsilon}^{p}$ is the rate of deviator plastic strain. The constant may be estimated from the slope of the shear stress - plastic shear strain curve at low levels of shear stress.

The function $\bar{F}$ is defined as

$$\bar{F} \equiv \max\left(0{,}1 - \frac{(\sigma-\alpha)\bullet\alpha}{2NF_e\left(J_1\right)}\right)$$

where N is a constant defining the size of the yield surface. The value of N may be interpreted as the radial distant between the outside of the initial yield surface and the inside of the limit surface. In order for the limit surface of the kinematic hardening cap model to correspond with

the failure envelope surface of the standard cap model, the scalar parameter a must be replaced $\alpha - N$ in the definition F_e.

The cap model contains a number of parameters which must be chosen to represent a particular material, and are generally based on experimental data. The parameters α, β, θ, and γ are usually evaluated by fitting a curve through failure data taken from a set of triaxial compression tests. The parameters *W*, *D*, and X_0 define the cap hardening law. The value *W* represents the void fraction of the uncompressed sample and *D* governs the slope of the initial loading curve in hydrostatic compression. The value of *R* is the ration of major to minor axes of the quarter ellipse defining the cap surface. Additional details and guidelines for fitting the cap model to experimental data are found in [Chen and Baladi, 1985].

Material Model 26: Crushable Foam

This orthotropic material model does the stress update in the local material system denoted by the subscripts, *a*, *b*, and *c*. The material model requires the following input parameters:

- *E*, Young's modulus for the fully compacted material;
- ν, Poisson's ratio for the compacted material;
- σ_y, yield stress for fully compacted honeycomb;
- LCA, load curve number for sigma-aa versus either relative volume or volumetric strain (see Figure 19.26.1.);
- LCB, load curve number for sigma-bb versus either relative volume or volumetric strain (default: LCB = LCA);
- LCC, the load curve number for sigma-cc versus either relative volume or volumetric strain (default: LCC = LCA);
- LCS, the load curve number for shear stress versus either relative volume or volumetric strain (default LCS = LCA);
- V_f, relative volume at which the honeycomb is fully compacted;
- E_{aau}, elastic modulus in the uncompressed configuration;
- E_{bbu}, elastic modulus in the uncompressed configuration;
- E_{ccu}, elastic modulus in the uncompressed configuration;
- G_{abu}, elastic shear modulus in the uncompressed configuration;
- G_{bcu}, elastic shear modulus in the uncompressed configuration;
- G_{cau}, elastic shear modulus in the uncompressed configuration;
- LCAB, load curve number for sigma-ab versus either relative volume or volumetric strain (default: LCAB = LCS);
- LCBC, load curve number for sigma-bc versus either relative volume or volumetric strain default: LCBC = LCS);
- LCCA, load curve number for sigma-ca versus either relative volume or volumetric strain (default: LCCA = LCS);
- LCSR, optional load curve number for strain rate effects.

The behavior before compaction is orthotropic where the components of the stress tensor are uncoupled, i.e., an a component of strain will generate resistance in the local a direction with no coupling to the local b and c directions. The elastic moduli vary linearly with the relative volume from their initial values to the fully compacted values:

$$E_{aa} = E_{aau} + \beta(E - E_{aau})$$

$$E_{bb} = E_{bbu} + \beta(E - E_{bbu}) \qquad (19.26.1)$$

$$E_{cc} = E_{ccu} + \beta(E - E_{ccu})$$
$$G_{ab} = G_{abu} + \beta(G - G_{abu})$$

$$G_{bc} = G_{bcu} + \beta(G - G_{bcu})$$

$$G_{ca} = G_{cau} + \beta(G - G_{cau})$$

where

$$\beta = \max\left[\min\left(\frac{1-V_{\min}}{1-V_f}, 1\right), 0\right] \qquad (19.26.2)$$

and G is the elastic shear modulus for the fully compacted honeycomb material

$$G = \frac{E}{2(1+\nu)} \qquad (19.26.3)$$

The relative volume V is defined as the ratio of the current volume over the initial volume; typically, $V = 1$ at the beginning of a calculation. The relative volume, $V_{\min}$, is the minimum value reached during the calculation.

The load curves define the magnitude of the average stress as the material changes density (relative volume). Each curve related to this model must have the same number of points and the same abscissa values. There are two ways to define these curves: as a function of relative volume *V*, or as a function of volumetric strain defined as:

$$\varepsilon_V = 1 - V \qquad (19.26.4)$$

In the former, the first value in the curve should correspond to a value of relative volume slightly less than the fully compacted value. In the latter, the first value in the curve should be less than or equal to zero corresponding to tension and should increase to full compaction. **When**

defining the curves, care should be taken that the extrapolated values do not lead to negative yield stresses.

At the beginning of the stress update we transform each element's stresses and strain rates into the local element coordinate system. For the uncompacted material, the trial stress components are updated using the elastic interpolated moduli according to:

$$\begin{aligned}
\sigma_{aa}^{n+1^{trial}} &= \sigma_{aa}^{n} + E_{aa}\Delta\varepsilon_{aa} \\
\sigma_{bb}^{n+1^{trial}} &= \sigma_{bb}^{n} + E_{bb}\Delta\varepsilon_{bb} \\
\sigma_{cc}^{n+1^{trial}} &= \sigma_{cc}^{n} + E_{cc}\Delta\varepsilon_{cc} \\
\sigma_{ab}^{n+1^{trial}} &= \sigma_{ab}^{n} + 2G_{ab}\Delta\varepsilon_{ab} \\
\sigma_{bc}^{n+1^{trial}} &= \sigma_{bc}^{n} + 2G_{bc}\Delta\varepsilon_{bc} \\
\sigma_{ca}^{n+1^{trial}} &= \sigma_{ca}^{n} + 2G_{ca}\Delta\varepsilon_{ca} = 1
\end{aligned} \tag{19.26.5}$$

Then we independently check each component of the updated stresses to ensure that they do not exceed the permissible values determined from the load curves, e.g., if

$$\left|\sigma_{ij}^{n+1^{trial}}\right| > \lambda\sigma_{ij}(V_{\min})$$

then

$$\sigma_{ij}^{n+1} = \sigma_{ij}(V_{\min})\frac{\lambda\sigma_{ij}^{n+1^{trial}}}{\left|\sigma_{ij}^{n+1^{trial}}\right|} \tag{19.26.6}$$

The parameter λ is either unity or a value taken from the load curve number, LCSR, that defines λ as a function of strain rate. Strain rate is defined here as the Euclidean norm of the deviatoric strain rate tensor.

For fully compacted material we assume that the material behavior is elastic-perfectly plastic and updated the stress components according to

$$s_{ij}^{trial} = s_{ij}^{n} + 2G\Delta\varepsilon_{ij}^{dev^{n+1/2}} \tag{19.26.7}$$

where the deviatoric strain increment is defined as

$$\Delta\varepsilon_{ij}^{dev} = \Delta\varepsilon_{ij} - \tfrac{1}{3}\Delta\varepsilon_{kk}\delta_{ij} \tag{19.26.8}$$

We next check to see if the yield stress for the fully compacted material is exceeded by comparing

$$s_{eff}^{trial} = \left(\tfrac{3}{2}s_{ij}^{trial}s_{ij}^{trial}\right)^{1/2} \tag{19.26.9}$$

the effective trial stress, to the yield stress σ_y. If the effective trial stress exceeds the yield stress, we simply scale back the stress components to the yield surface:

$$s_{ij}^{n+1} = \frac{\sigma_y}{s_{eff}^{trial}} s_{ij}^{trial} \tag{19.26.10}$$

We can now update the pressure using the elastic bulk modulus, *K*:

$$\begin{aligned} p^{n+1} &= p^n - K\Delta\varepsilon_{kk}^{n+\frac{1}{2}} \\ K &= \frac{E}{3(1-2\nu)} \end{aligned} \tag{19.26.11}$$

and obtain the final value for the Cauchy stress

$$\sigma_{ij}^{n+1} = s_{ij}^{n+1} - p^{n+1}\delta_{ij} \tag{19.26.12}$$

After completing the stress update, we transform the stresses back to the global configuration.

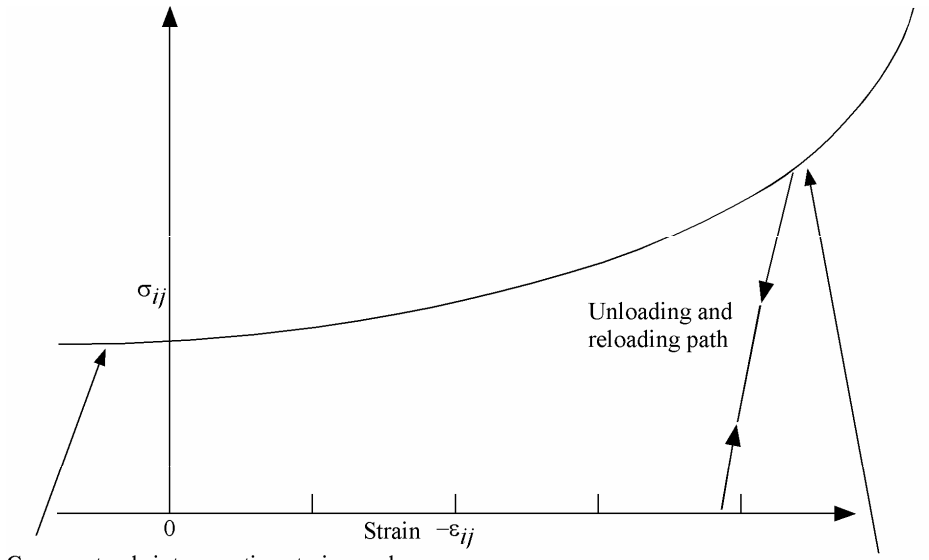


Figure 19.26.1. Stress quantity versus volumetric strain. Note that the "yield stress" at a volumetric strain of zero is nonzero. In the load curve definition, the "time" value is the volumetric strain and the "function" value is the yield stress.

Material Model 27: Incompressible Mooney-Rivlin Rubber

The Mooney-Rivlin material model is based on a strain energy function, W, as follows

$$W = A(I_1 - 3) + B(I_2 - 3) + C(\frac{1}{I_3^2} - 1) + D(I_3 - 1)^2 \qquad (19.27.1)$$

A and B are user defined constants, whereas C* and D are related to A and B as follows

$$C = \frac{1}{2}A + B \qquad (19.27.2)$$

$$D = \frac{A(5\upsilon - 2) + B(11\upsilon - 5)}{2(1 - 2\upsilon)} \qquad (19.27.3)$$

The derivation of the constants C and D is straightforward [Feng, 1993] and is included here since we were unable to locate it in the literature. The principal components of Cauchy stress, σ_i, are given by [Ogden, 1984]

$$J\sigma_i = \lambda_i \frac{\partial W}{\partial \lambda_i} \qquad (19.27.4)$$

For uniform dilation

$$\lambda_1 = \lambda_2 = \lambda_3 = \lambda \qquad (19.27.5)$$

thus the pressure, p, is obtained (please note the sign convention),

$$p = \sigma_1 = \sigma_2 = \sigma_3 = \frac{2}{\lambda^3}\left(\lambda^2 \frac{\partial W}{\partial I_1} + 2\lambda^4 \frac{\partial W}{\partial I_2} + \lambda^6 \frac{\partial W}{\partial I_3} \right) \qquad (19.27.6)$$

The relative volume, V, can be defined in terms of the stretches as:

$$V = \lambda^3 = \frac{new\ volume}{old\ volume} \qquad (19.27.7)$$

For small volumetric deformations the bulk modulus, K, can be defined as the ratio of the pressure over the volumetric strain as the relative volume approaches unity:

$$K = \lim_{V \to 1}\left(\frac{p}{V - 1} \right) \qquad (19.27.8)$$

* Please observe the difference between the constant, C, and the right Cauchy Green tensor **C**, which will be denoted either by boldface tensorial notation or through its tensorial components, $C_{ij,}$ throughout the report.

The partial derivatives of W lead to:

$$\begin{aligned}
\frac{\partial W}{\partial I_1} &= A \\
\frac{\partial W}{\partial I_2} &= B \\
\frac{\partial W}{\partial I_3} &= -2CI_3^{-3} + 2D\left(I_3 - 1\right) = -2C\lambda^{-18} + 2D\left(\lambda^6 - 1\right) \\
p &= \frac{2}{\lambda^3}\left\{A\lambda^2 + 2\lambda^4 B + \lambda^6\left[-2C\lambda^{-18} + 2D\left(\lambda^6 - 1\right)\right]\right\} \\
&= \frac{2}{\lambda^3}\left\{A\lambda^2 + 2\lambda^4 B - 2C\lambda^{-12} + 2D\left(\lambda^{12} - \lambda^6\right)\right\}
\end{aligned} \tag{19.27.9}$$

In the limit as the stretch ratio approaches unity, the pressure must approach zero:

$$\lim_{\lambda \to 1} p = 0 \tag{19.27.10}$$

Therefore, $A + 2B - 2C = 0$ and

$$\therefore C = 0.5A + B \tag{19.27.11}$$

To solve for D we note that:

$$K = \lim_{V \to 1}\left(\frac{p}{V-1}\right) = \lim_{\lambda \to 1}\frac{\frac{2}{\lambda^3}\left\{A\lambda^2 + 2\lambda^4 B - 2C\lambda^{-12} + 2D\left(\lambda^{12} - \lambda^6\right)\right\}}{\lambda^3 - 1} \tag{19.27.12}$$

$$= 2\lim_{\lambda \to 1}\frac{A\lambda^2 + 2\lambda^4 B - 2C\lambda^{-12} + 2D\left(\lambda^{12} - \lambda^6\right)}{\lambda^6 - \lambda^3}$$

$$= 2\lim_{\lambda \to 1}\frac{2A\lambda + 8\lambda^3 B + 24C\lambda^{-13} + 2D\left(12\lambda^{11} - 6\lambda^5\right)}{6\lambda^5 - 3\lambda^2}$$

$$= \frac{2}{3}\left(2A + 8B + 24C + 12D\right) = \frac{2}{3}\left(14A + 32B + 12D\right)$$

We therefore obtain:

$$14A+32B+12D=\frac{3}{2}K=\frac{3}{2}\left(\frac{2G(1+\upsilon)}{3(1-2\upsilon)}\right)=\frac{2(A+B)(1+\upsilon)}{(1-2\upsilon)} \qquad (19.27.13)$$

$$D=\frac{A(5\upsilon-2)+B(11\upsilon-5)}{2(1-2\upsilon)} \qquad (19.27.14)$$

The invariants I_1-I_3 are related to the right Cauchy-Green tensor **C** as

$$I_1=C_{ii} \qquad (19.27.15)$$

$$I_2=\frac{1}{2}C_{ii}{}^2-\frac{1}{2}C_{ij}C_{ij} \qquad (19.27.16)$$

$$I_3=\det(C_{ij}) \qquad (19.27.17)$$

The second Piola-Kirchhoff stress tensor, **S**, is found by taking the partial derivative of the strain energy function with respect to the Green-Lagrange strain tensor, **E**.

$$S_{ij}=\frac{\partial W}{\partial E_{ij}}=2\frac{\partial W}{\partial C_{ij}}=2\left(A\frac{\partial I_1}{\partial C_{ij}}+B\frac{\partial I_2}{\partial C_{ij}}+\left(2D(I_3-1)-\frac{2C}{I_3^2}\right)\frac{\partial I_3}{\partial C_{ij}}\right) \qquad (19.27.18)$$

The derivatives of the invariants I_1-I_3 are

$$\frac{\partial I_1}{\partial C_{ij}}=\delta_{ij}$$

$$\frac{\partial I_2}{\partial C_{ij}}=I_1\delta_{ij}-C_{ij} \qquad (19.27.19)$$

$$\frac{\partial I_3}{\partial C_{ij}}=I_3C_{ij}^{-1}$$

Inserting Equation (19.27.19) into Equation (19.27.18) yields the following expression for the second Piola-Kirchhoff stress:

$$S_{ij}=2A\delta_{ij}+2B\left(I_1\delta_{ij}-C_{ij}\right)-4C\frac{1}{I_3^2}C_{ij}^{-1}+4D(I_3-1)I_3C_{ij}^{-1} \qquad (19.27.20)$$

Equation (19.27.20) can be transformed into the Cauchy stress by using the push forward operation

$$\sigma_{ij} = \frac{1}{J} F_{ik} S_{kl} F_{jl} \tag{19.27.21}$$

where $J = \det(F_{ij})$.

19.27.1 Stress Update for Shell Elements

As a basis for discussing the algorithmic tangent stiffness for shell elements in Section 19.27.3, the corresponding stress update as it is done in LS-DYNA is shortly recapitulated in this section. When dealing with shell elements, the stress (as well as constitutive matrix) is typically evaluated in corotational coordinates after which it is transformed back to the standard basis according to

$$\sigma_{ij} = R_{ik} R_{jl} \hat{\sigma}_{kl} .$$

Here R_{ij} is the rotation matrix containing the corotational basis vectors. The so-called corotated stress $\hat{\sigma}_{ij}$ is evaluated using Equation 19.27.21 with the exception that the deformation gradient is expressed in the corotational coordinates, i.e.,

$$\hat{\sigma}_{ij} = \frac{1}{J} \hat{F}_{ik} S_{kl} \hat{F}_{jl} \tag{19.27.22}$$

where S_{ij} is evaluated using Equation (19.27.20). The corotated deformation gradient is incrementally updated with the aid of a time increment Δt, the corotated velocity gradient $\hat{L}_{ij}$, and the angular velocity $\hat{\Omega}_{ij}$ with which the embedded coordinate system is rotating.

$$\hat{F}_{ij} = (\delta_{ik} + \Delta t \hat{L}_{ik} - \Delta t \hat{\Omega}_{ik}) \hat{F}_{kj} \tag{19.27.23}$$

The primary reason for taking a corotational approach is to facilitate the maintenance of a vanishing normal stress through the thickness of the shell, something that is achieved by adjusting the corresponding component of the corotated velocity gradient $\hat{L}_{33}$ accordingly. The problem can be stated as to determine $\hat{L}_{33}$ such that when updating the deformation gradient through Equation (19.27.23) and subsequently the stress through Equation (19.27.22), $\hat{\sigma}_{33} = 0$. To this end, it is assumed that

$$\hat{L}_{33} = \alpha(\hat{L}_{11} + \hat{L}_{22}) ,$$

for some parameter α that is determined in the following three step procedure. In the first two steps, $\alpha = 0$ and $\alpha = -1$, respectively, resulting in two trial normal stresses $\hat{\sigma}_{33}^{(0)}$ and $\hat{\sigma}_{33}^{(-1)}$.

Then it is assumed that the actual normal stress depends linearly on α, meaning that the latter can be determined from

$$0 = \sigma_{33}^{(\alpha)} = \sigma_{33}^{(0)} + \alpha(\sigma_{33}^{(0)} - \sigma_{33}^{(-1)}).$$

In LS-DYNA, α is given by

$$\alpha = \begin{cases} \dfrac{\hat{\sigma}_{33}^{(0)}}{\hat{\sigma}_{33}^{(-1)} - \hat{\sigma}_{33}^{(0)}}, & \text{if } \left|\hat{\sigma}_{33}^{(-1)} - \hat{\sigma}_{33}^{(0)}\right| \geq 10^{-4} \\ -1, & \text{otherwise} \end{cases},$$

and the stresses are determined from this value of α. Finally, to make sure that the normal stress through the thickness vanishes, it is set to 0 (zero) before exiting the stress update routine.

19.27.2 Derivation of the Continuum Tangent Stiffness

This section will describe the derivation of the continuum tangent stiffness for the Mooney-Rivlin material. For solid elements, the continuum tangent stiffness is chosen in favor of an algorithmic (consistent) tangential modulus as the constitutive equation at hand is smooth and a consistent tangent modulus is not required for good convergence properties. For shell elements however, this stiffness must ideally be modified in order to account for the zero normal stress condition. This modification, and its consequences, are discussed in the next section.
The continuum tangent modulus in the reference configuration is per definition,

$$E_{ijkl}^{PK} = \frac{\partial S_{ij}}{\partial E_{kl}} = 2\frac{\partial S_{ij}}{\partial C_{kl}} \tag{19.27.24}$$

Splitting up the differentiation of Equation (19.27.20) we get

$$\frac{\partial\left(I_1\delta_{ij} - C_{ij}\right)}{\partial C_{kl}} = \delta_{kl}\delta_{ij} - \frac{1}{2}\left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right)$$

$$\frac{\partial\left(\dfrac{1}{I_3^2}C_{ij}^{-1}\right)}{\partial C_{kl}} = -\frac{2}{I_3^2}C_{kl}^{-1}C_{ij}^{-1} - \frac{1}{2I_3^2}\left(C_{kj}^{-1}C_{il}^{-1} + C_{lj}^{-1}C_{ik}^{-1}\right) \tag{19.27.25}$$

$$\frac{\partial\left(I_3(I_3-1)C_{ij}^{-1}\right)}{\partial C_{kl}} = I_3(2I_3-1)C_{kl}^{-1}C_{ij}^{-1} - \frac{1}{2}I_3(I_3-1)(C_{kj}^{-1}C_{il}^{-1} + C_{lj}^{-1}C_{ik}^{-1})$$

Since LS-DYNA needs the tangential modulus for the Cauchy stress, it is a good idea to transform the terms in Equation (19.27.25) before summing them up. The push forward operation for the fourth-order tensor E_{ijkl}^{PK} is

$$E_{ijkl}^{TC} = \frac{1}{J} F_{ia} F_{jb} F_{kc} F_{ld} E_{abcd}^{PK} \tag{19.27.26}$$

Since the right Cauchy-Green tensor is $\mathbf{C}=\mathbf{F}^t\mathbf{F}$ and the left Cauchy-Green tensor is $\mathbf{b}=\mathbf{F}\mathbf{F}^t$, and the determinant and trace of the both stretches are equal, the transformation is in practice carried out by interchanging

$$C_{ij}^{-1} \to \delta_{ij}, \qquad \delta_{ij} \to b_{ij}$$

The end result is then

$$JE_{ijkl}^{TC} = 4B\left[b_{kl}b_{ij} - \frac{1}{2}\left(b_{ik}b_{jl} + b_{il}b_{jk}\right)\right] + \frac{4C}{I_3^2}\left[4\delta_{ij}\delta_{kl} + \left(\delta_{kj}\delta_{il} + \delta_{lj}\delta_{im}\right)\right] + 8DI_3\left[\left(2I_3 - 1\right)\delta_{ij}\delta_{kl} - \frac{1}{2}\left(I_3 - 1\right)\left(\delta_{kj}\delta_{il} + \delta_{lj}\delta_{ik}\right)\right] \tag{19.27.27}$$

19.27.3 The Algorithmic Tangent Stiffness for Shell Elements

The corotated tangent stiffness matrix is given by Equation (19.27.27) with the exception that the left Cauchy-Green tensor and deformation gradient are given in corotational coordinates, i.e.,

$$J\hat{E}_{ijkl}^{TC} = 4B\left[\hat{b}_{kl}\hat{b}_{ij} - \frac{1}{2}\left(\hat{b}_{ik}\hat{b}_{jl} + \hat{b}_{il}\hat{b}_{jk}\right)\right] + \frac{4C}{I_3^2}\left[4\delta_{ij}\delta_{kl} + \left(\delta_{kj}\delta_{il} + \delta_{lj}\delta_{im}\right)\right] + 8DI_3\left[\left(2I_3 - 1\right)\delta_{ij}\delta_{kl} - \frac{1}{2}\left(I_3 - 1\right)\left(\delta_{kj}\delta_{il} + \delta_{lj}\delta_{ik}\right)\right] \tag{19.27.28}$$

Using this exact expression for the tangent stiffness matrix in the context of shell elements is not adequate since it does not take into account that the normal stress is zero and it must be modified appropriately. To this end, we assume that the tangent moduli in Equation (19.27.28) relates the corotated rate-of-deformation tensor $\hat{D}_{ij}$ to the corotated rate of stress $\overset{\bullet}{\hat{\sigma}}_{ij}$,

$$\overset{\bullet}{\hat{\sigma}}_{ij} = \hat{E}_{ijkl}^{TC}\hat{D}_{kl} \tag{19.27.29}$$

Even though this is not completely true, we believe that attempting a more thorough treatment would hardly be worth the effort. The objective can now be stated as to find a modified tangent stiffness matrix $\hat{E}_{ijkl}^{TC\text{alg}}$ such that

$$\overset{\bullet}{\hat{\sigma}}_{ij}^{\text{alg}} = \hat{E}_{ijkl}^{TC\text{alg}}\hat{D}_{kl} \tag{19.27.30}$$

where $\hat{\sigma}_{ij}^{\text{alg}}$ is the stress as it is evaluated in LS-DYNA. The stress update, described in Section 19.27.1, is performed in a rather ad hoc way which probably makes the stated objective unachievable. Still we attempt to extract relevant information from it that enables us to come somewhat close.

An example of a modification of this tangent moduli is due to Hughes and Liu [1981] and given by

$$\hat{E}_{ijkl}^{TC\text{alg}} = \hat{E}_{ijkl}^{TC} - \frac{\hat{E}_{ij33}^{TC}\hat{E}_{33kl}^{TC}}{\hat{E}_{3333}^{TC}}.$$

This matrix is derived by eliminating the thickness strain $\hat{D}_{33}$ from the equation $\dot{\hat{\sigma}}_{33} = 0$ in Equation (19.27.30) as an unknown. This modification is unfortunately not consistent with how the stresses are updated in LS-DYNA. When consulting Section 19.27.1, it is suggested that $\hat{D}_{33}$ instead can be eliminated from

$$\hat{D}_{33} = \alpha(\hat{D}_{11} + \hat{D}_{22}) \tag{19.27.31}$$

using the α determined from the stress update. Unfortunately, by the time when the tangent stiffness matrix is calculated, the exact value of α is not known. From experimental observations however, we have found that α is seldom far from being equal to -1. The fact that $\alpha = -1$ represents incompressibility strengthen this hypothesis. This leads to a modified tangent stiffness $\hat{E}_{ijkl}^{TC\text{alg}}$ that is equal to $\hat{E}_{ijkl}^{TC}$ except for the following modifications,

$$\begin{aligned} \hat{E}_{iijj}^{TC\text{alg}} &= \hat{E}_{iijj}^{TC} - \hat{E}_{ii33}^{TC} - \hat{E}_{33jj}^{TC} + \hat{E}_{3333}^{TC}, \\ \hat{E}_{33ij}^{TC\text{alg}} &= \hat{E}_{ij33}^{TC\text{alg}} = 0, \ i \neq j \end{aligned} \tag{19.27.32}$$

To preclude the obvious singularity, a small positive value is assigned to $\hat{E}_{3333}^{TC\text{alg}}$,

$$\hat{E}_{3333}^{TC\text{alg}} = 10^{-4}\left(\left|\hat{E}_{1111}^{TC\text{alg}}\right| + \left|\hat{E}_{2222}^{TC\text{alg}}\right|\right).$$

As with the Hughes-Liu modification, this modification preserves symmetry and positive definiteness of the tangent moduli, which together with the stress update "consistency" makes it intuitively attractive.

Material Model 28: Resultant Plasticity

This plasticity model, based on resultants as illustrated in Figure 19.29.1, is very cost effective but not as accurate as through-thickness integration. This model is available only with the C^0 triangular, Belytschko-Tsay shell, and the Belytschko beam element since these elements, unlike the Hughes-Liu elements, lend themselves very cleanly to a resultant formulation.

In applying this model to shell elements the resultants are updated incrementally using the midplane strains ε^m and curvatures κ:

$$\Delta n = \Delta t\, C\, \varepsilon^m \tag{19.28.1}$$

$$\Delta m = \Delta t \frac{h^3}{12} C\, \kappa \tag{19.28.2}$$

where the plane stress constitutive matrix is given in terms of Young's Modulus E and Poisson's ratio ν as:

$$\bar{m} = m_{xx}^2 - m_{xx}m_{yy} + m_{yy}^2 + 3m_{xy}^2 \tag{19.28.3}$$

Defining

$$\bar{n} = n_{xx}^2 - n_{xx}n_{yy} + n_{yy}^2 + 3n_{xy}^2 \tag{19.28.4}$$

$$\bar{m} = m_{xx}^2 - m_{xx}m_{yy} + m_{yy}^2 + 3m_{xy}^2 \tag{19.28.5}$$

$$\bar{m}\bar{n} = m_{xx}n_{xx} - \tfrac{1}{2}m_{xx}n_{yy} - \tfrac{1}{2}n_{xx}m_{yy} + m_y n_y + 3m_{xy}n_{xy} \tag{19.28.6}$$

the Ilyushin yield function becomes

$$f\left(m,n\right) = \bar{n} + \frac{4\left|\bar{m}\bar{n}\right|}{h\sqrt{3}} + \frac{16\bar{m}}{h^2} \le n_y^2 = h^2\sigma_y^2 \tag{19.28.7}$$

In our implementation we update the resultants elastically and check to see if the yield condition is violated:

$$f(m,n) > n_y^2 \tag{19.28.8}$$

If so, the resultants are scaled by the factor α:

$$\alpha = \sqrt{\frac{n_y^2}{f\left(m,n\right)}} \tag{19.28.9}$$

We update the yield stress incrementally:

$$\sigma_y^{n+1} = \sigma_y^n + E^P \Delta\varepsilon_{plastic}^{eff} \tag{19.28.10}$$

where E^P is the plastic hardening modulus which in incremental plastic strain is approximated by

$$\Delta\varepsilon_{plastic}^{eff} = \frac{\sqrt{f(m,n)} - n_y}{h\left(3G + E^p\right)} \tag{19.28.11}$$

Kennedy, et. al., report that this model predicts results that may be too stiff; users of this model should proceed cautiously.

In applying this material model to the Belytschko beam, the flow rule changes to

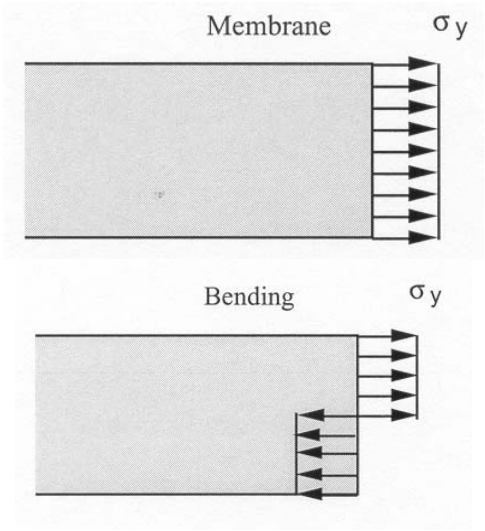
$$f(m,n) = \hat{f}_x^2 + \frac{4\hat{m}_y^2}{3I_{yy}} + \frac{4\hat{m}_z^2}{3I_{zz}} \le n_y^2 = A^2\sigma_y^2 \tag{19.28.12}$$

have been updated elastically according to Equations (4.16)-(4.18). The yield condition is checked [Equation (19.28.8)], and if it is violated, the resultants are scaled as described above.

This model is frequently applied to beams with non-rectangular cross sections. The accuracy of the results obtained should be viewed with some healthy suspicion. No work hardening is available with this model.

Material Model 29: FORCE LIMITED Resultant Formulation

This material model is available for the Belytschko beam element only. Plastic hinges form at the ends of the beam when the moment reaches the plastic moment. The moment-versus-rotation relationship is specified by the user in the form of a load curve and scale factor. The point pairs of the load curve are *(plastic rotation in radians, plastic moment).* Both quantities should be positive for all points, with the first point pair being *(zero, initial plastic moment).* Within this constraint any form of characteristic may be used including flat or falling curves. Different load curves and scale factors may be specified at each node and about each of the local *s* and *t* axes.



$$n_y = \int_{-\frac{h}{2}}^{\frac{h}{2}} \sigma_y d\zeta = h\sigma_y$$

$$m_y = \int_{-\frac{h}{2}}^{\frac{h}{2}} \sigma_y \zeta d\zeta = \frac{h^2}{4}\sigma_y$$

Figure 19.29.1. Full section yield using resultant plasticity.

Axial collapse occurs when the compressive axial load reaches the collapse load. The collapse load-versus-collapse deflection is specified in the form of a load curve. The points of the load curve are *(true strain, collapse force).* Both quantities should be entered as positive for all points, and will be interpreted as compressive i.e., collapse does not occur in tension. The first point should be the pair *(zero, initial collapse load).*

The collapse load may vary with end moment and with deflection. In this case, several load-deflection curves are defined, each corresponding to a different end moment. Each load curve should have the same number of point pairs and the same deflection values. The end moment is defined as the average of the absolute moments at each end of the beam, and is always positive.

It is not possible to make the plastic moment vary with axial load.

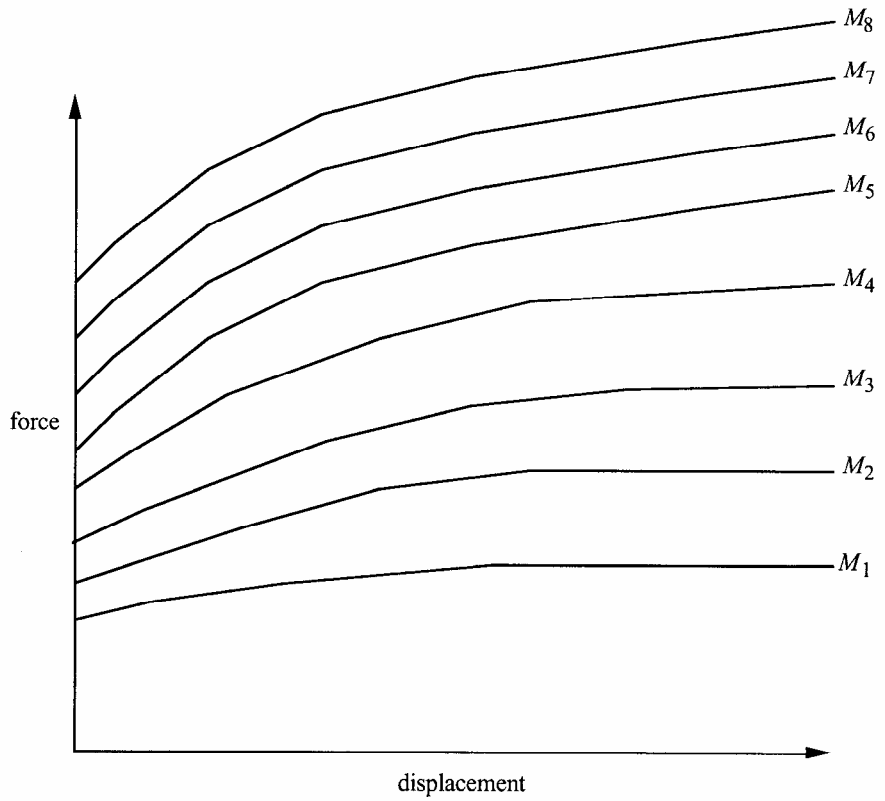


Figure 19.29.2. The force magnitude is limited by the applied end moment. For an intermediate value of the end moment, LS-DYNA interpolates between the curves to determine the allowable force.

A co-rotational technique and moment-curvature relations are used to compute the internal forces. The co-rotational technique is treated in Section 4 in and will not be treated here as we will focus solely on the internal force update and computing the tangent stiffness. For this we use the notation

E = Young's modulus

G = Shear modulus

A = Cross sectional area

A_s = Effective area in shear

l^n = Reference length of beam

l^{n+1} = Current length of beam

I_{yy} = Second moment of inertia about y

I_{zz} = Second moment of inertia about z

J = Polar moment of inertia

$\mathbf{e}_i$ = ith local base vector in the current configuration

$\mathbf{y}_I$ = nodal vector in y direction at node I in the current configuration

$\mathbf{z}_I$ = nodal vector i z direction at node I in the current configuration

We emphasize that the local y and z base vectors in the reference configuration always coincide with the corresponding nodal vectors. The nodal vectors in the current configuration are updated using the Hughes-Winget formula while the base vectors are computed from the current geometry of the element and the current nodal vectors.

19.29.1 Internal Forces

Elastic Update

In the local system for a beam connected by nodes *I* and *J*, the axial force is updated as

$$f_a^{el} = f_a^n + K_a^{el}\delta \tag{19.29.1}$$

where

$$K_a^{el} = \frac{EA}{l^n} \tag{19.29.2}$$

$$\delta = l^{n+1} - l^n . \tag{19.29.3}$$

The torsional moment is updated as

$$m_t^{el} = m_t^n + K_t^{el}\theta_t \tag{19.29.4}$$

where

$$K_t^{el} = \frac{GJ}{l^n} \tag{19.29.5}$$

$$\theta_t = \frac{1}{2}\mathbf{e}_1^T(\mathbf{y}_I \times \mathbf{y}_J + \mathbf{z}_I \times \mathbf{z}_J) . \tag{19.29.6}$$

The bending moments are updated as

$$\mathbf{m}_y^{el} = \mathbf{m}_y^n + \mathbf{A}_y^{el}\boldsymbol{\theta}_y \tag{19.29.7}$$

$$\mathbf{m}_z^{el} = \mathbf{m}_z^n + \mathbf{A}_z^{el}\boldsymbol{\theta}_z \tag{19.29.8}$$

where

$$\mathbf{A}_*^{el} = \frac{1}{1+\phi_*}\frac{EI_{**}}{l^n}\begin{pmatrix} 4+\phi_* & 2-\phi_* \\ 2-\phi_* & 4+\phi_* \end{pmatrix} \tag{19.29.9}$$

$$\phi_* = \frac{12EI_{**}}{GA_s l^n t^n} \tag{19.29.10}$$

$$\boldsymbol{\theta}_y^T = -\mathbf{e}_3^T\begin{pmatrix} \mathbf{y}_I \times \mathbf{z}_I & \mathbf{y}_J \times \mathbf{z}_J \end{pmatrix} \tag{19.29.11}$$

$$\boldsymbol{\theta}_z^T = \mathbf{e}_2^T\begin{pmatrix} \mathbf{y}_I \times \mathbf{z}_I & \mathbf{y}_J \times \mathbf{z}_J \end{pmatrix}. \tag{19.29.12}$$

In the following we refer to $\mathbf{A}_*^{el}$ as the (elastic) moment-rotation matrix.

Plastic Correction

After the elastic update the state of force is checked for yielding as follows. As a preliminary note we emphasize that whenever yielding does not occur the elastic stiffnesses and forces are taken as the new stiffnesses and forces.

The yield moments in direction *i* at node *I* as functions of plastic rotations are denoted $m_{iI}^Y(\theta_{iI}^P)$. This function is given by the user but also depends on whether a plastic hinge has been created. The theory for plastic hinges is given in the LS-DYNA Keyword User's Manual [Hallquist 2003] and is not treated here. Whenever the elastic moment exceeds the plastic moment, the plastic rotations are updated as

$$\theta_{iI}^{P(n+1)} = \theta_{iI}^{P(n)} + \frac{\left|m_{iI}^{el}\right| - m_{iI}^Y}{\max(0.001, A_{i(II)}^{el} + \dfrac{\partial m_{iI}^Y}{\partial \theta_{iI}^P})} \tag{19.29.13}$$

and the moment is reduced to the yield moment

$$m_{iI}^{n+1} = m_{iI}^Y(\theta_{iI}^{P(n+1)})\,\mathrm{sgn}(m_{iI}^{el}). \tag{19.29.14}$$

The corresponding diagonal component in the moment-rotation matrix is reduced as

$$A_{i(II)}^{n+1} = A_{i(II)}^{el}(1-\alpha\frac{A_{i(II)}^{el}}{\max(0.001, A_{i(II)}^{el}+\frac{\partial m_{iI}^{Y}}{\partial \theta_{iI}^{P}})}) \qquad (19.29.15)$$

where $\alpha \le 1$ is a parameter chosen such that the moment-rotation matrix remains positive definite.

The yield moment in torsion is given by $m_t^Y(\theta_t^P)$ and is provided by the user. If the elastic torsional moment exceeds this value, the plastic torsional rotation is updated as

$$\theta_t^{P(n+1)} = \theta_t^{P(n)} + \frac{\left|m_t^{el}\right| - m_t^Y}{\max(0.001, K_t^{el}+\frac{\partial m_t^Y}{\partial \theta_t^P})} \qquad (19.29.16)$$

and the moment is reduced to the yield moment

$$m_t^{n+1} = m_t^Y(\theta_t^{P(n+1)})\,\mathrm{sgn}(m_t^{el}). \qquad (19.29.17)$$

The torsional stiffness is modified as

$$K_t^{n+1} = K_t^{el}(1-\alpha\frac{K_t^{el}}{K_t^{el}+\frac{\partial m_t^Y}{\partial \theta_t^P}}) \qquad (19.29.18)$$

where again $\alpha \le 1$ is chosen so that the stiffness is positive.

Axial collapse is modeled by limiting the axial force by $f_a^Y(\varepsilon, m)$, i.e., a function of the axial strains and the magnitude of bending moments. If the axial elastic force exceeds this value it is reduced to yield

$$f_a^{n+1} = f_a^Y(\varepsilon^{n+1}, m^{n+1})\,\mathrm{sgn}(f_a^{el}) \qquad (19.29.19)$$

and the axial stiffness is given by

$$K_a^{n+1} = \max(0.05K_a^{el}, \frac{\partial f_a^Y}{\partial \varepsilon}). \qquad (19.29.20)$$

We neglect the influence of change in bending moments when computing this parameter.

Damping

Damping is introduced by adding a viscous term to the internal force on the form

$$\mathbf{f}_v = \mathbf{D}\frac{d}{dt}\begin{bmatrix}\delta\\ \theta_t\\ \boldsymbol{\theta}_y\\ \boldsymbol{\theta}_z\end{bmatrix} \tag{19.29.21}$$

$$\mathbf{D} = \gamma\begin{bmatrix}K_a^{el} & & & \\ & K_t^{el} & & \\ & & \mathbf{A}_y^{el} & \\ & & & \mathbf{A}_z^{el}\end{bmatrix} \tag{19.29.22}$$

where γ is a damping parameter.

Transformation

The internal force vector in the global system is obtained through the transformation

$$\mathbf{f}_g^{n+1} = \mathbf{S}\mathbf{f}_l^{n+1} \tag{19.29.23}$$

where

$$\mathbf{S} = \begin{bmatrix}-\mathbf{e}_1 & \mathbf{0} & -\mathbf{e}_3/l^{n+1} & -\mathbf{e}_3/l^{n+1} & \mathbf{e}_2/l^{n+1} & \mathbf{e}_2/l^{n+1}\\ \mathbf{0} & -\mathbf{e}_1 & \mathbf{e}_2 & \mathbf{0} & \mathbf{e}_3 & \mathbf{0}\\ \mathbf{e}_1 & \mathbf{0} & \mathbf{e}_3/l^{n+1} & \mathbf{e}_3/l^{n+1} & -\mathbf{e}_2/l^{n+1} & -\mathbf{e}_2/l^{n+1}\\ \mathbf{0} & \mathbf{e}_1 & \mathbf{0} & \mathbf{e}_2 & \mathbf{0} & \mathbf{e}_3\end{bmatrix} \tag{19.29.24}$$

$$\mathbf{f}_l^{n+1} = \begin{bmatrix}f_a^{n+1}\\ m_t^{n+1}\\ \mathbf{m}_y^{n+1}\\ \mathbf{m}_z^{n+1}\end{bmatrix}. \tag{19.29.25}$$

19.29.2 Tangent Stiffness

Derivation

The tangent stiffness is derived from taking the variation of the internal force

$$\delta\mathbf{f}_g^{n+1} = \delta\mathbf{S}\mathbf{f}_l^{n+1} + \mathbf{S}\delta\mathbf{f}_l^{n+1} \tag{19.29.26}$$

which can be written

$$\delta \mathbf{f}_g^{n+1} = \mathbf{K}^{geo}\delta\mathbf{u} + \mathbf{K}^{mat}\delta\mathbf{u} \tag{19.29.27}$$

where

$$\delta\mathbf{u} = \left(\delta\mathbf{x}_I^T \quad \delta\boldsymbol{\omega}_I^T \quad \delta\mathbf{x}_J^T \quad \delta\boldsymbol{\omega}_J^T\right)^T. \tag{19.29.28}$$

There are two contributions to the tangent stiffness, one geometrical and one material contribution. The geometrical contribution is given (approximately) by

$$\mathbf{K}^{geo} = \mathbf{R}(\mathbf{f}_l^{n+1} \otimes \mathbf{I})\mathbf{W} - \frac{1}{l^{n+1}L^{n+1}}\mathbf{T}\mathbf{f}_l^{n+1}\mathbf{L} \tag{19.29.29}$$

where

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 & \mathbf{0} & \mathbf{R}_3/l^{n+1} & \mathbf{R}_3/l^{n+1} & -\mathbf{R}_2/l^{n+1} & -\mathbf{R}_2/l^{n+1} \\ \mathbf{0} & \mathbf{R}_1 & -\mathbf{R}_2 & \mathbf{0} & -\mathbf{R}_3 & \mathbf{0} \\ -\mathbf{R}_1 & \mathbf{0} & -\mathbf{R}_3/l^{n+1} & -\mathbf{R}_3/l^{n+1} & \mathbf{R}_2/l^{n+1} & \mathbf{R}_2/l^{n+1} \\ \mathbf{0} & -\mathbf{R}_1 & \mathbf{0} & -\mathbf{R}_2 & \mathbf{0} & -\mathbf{R}_3 \end{bmatrix} \tag{19.29.30}$$

$$\mathbf{W} = \left(-\mathbf{R}_1/l^{n+1} \quad \mathbf{e}_1\mathbf{e}_1^T/2 \quad \mathbf{R}_1/l^{n+1} \quad \mathbf{e}_1\mathbf{e}_1^T/2\right) \tag{19.29.31}$$

$$\mathbf{T} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & -\mathbf{e}_3 & -\mathbf{e}_3 & \mathbf{e}_2 & \mathbf{e}_2 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{e}_3 & \mathbf{e}_3 & -\mathbf{e}_2 & -\mathbf{e}_2 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{19.29.32}$$

$$\mathbf{L} = \left(-\mathbf{e}_1^T \quad \mathbf{0} \quad \mathbf{e}_1^T \quad \mathbf{0}\right) \tag{19.29.33}$$

and $\mathbf{I}$ is the 3 by 3 identity matrix. We use $\otimes$ as the outer matrix product and define

$$\mathbf{R}_i\mathbf{v} = \mathbf{e}_i \times \mathbf{v}. \tag{19.29.34}$$

The material contribution can be written as

$$\mathbf{K}^{mat} = \mathbf{S}\mathbf{K}\mathbf{S}^T \tag{19.29.35}$$

where

$$\mathbf{K} = \begin{bmatrix} K_a^{n+1} & & & \\ & K_t^{n+1} & & \\ & & \mathbf{A}_y^{n+1} & \\ & & & \mathbf{A}_z^{n+1} \end{bmatrix} + \frac{1}{\Delta t}\mathbf{D}\,. \qquad (19.29.36)$$

Material Model 30: Closed-Form Update Shell Plasticity

This section presents the mathematical details of the shape memory alloy material in LS-DYNA. The description closely follows the one of Auricchio and Taylor [1997] with appropriate modifications for this particular implementation.

19.30.1 Mathematical Description of the Material Model

The Kirchhoff stress $\boldsymbol{\tau}$ in the shape memory alloy can be written

$$\boldsymbol{\tau} = p\mathbf{i} + \mathbf{t} \qquad (19.30.1)$$

where $\mathbf{i}$ is the second order identity tensor and

$$\begin{aligned} p &= K(\theta - 3\alpha\xi_S\varepsilon_L) \qquad (19.30.2)\\ \mathbf{t} &= 2G(\mathbf{e} - \xi_S\varepsilon_L\mathbf{n})\,. \end{aligned}$$

Here K and G are bulk and shear modulii, θ and $\mathbf{e}$ are volumetric and shear logarithmic strains and α and ε_L are constant material parameters. There is an option to define the bulk and shear modulii as functions of the martensite fraction according to

$$\begin{aligned} K &= K_A + \xi_S(K_S - K_A)\\ G &= G_A + \xi_S(G_S - G_A) \end{aligned}$$

in case the stiffness of the martensite differs from that of the austenite. Furthermore, the unit vector $\mathbf{n}$ is defined as

$$\mathbf{n} = \mathbf{e}/(\|\mathbf{e}\| + 10^{-12})\,, \qquad (19.30.3)$$

and a loading function is introduced as

$$F = 2G\|\mathbf{e}\| + 3\alpha K\theta - \beta\xi_S \qquad (19.30.4)$$

where

$$\beta = (2G + 9\alpha^2 K)\varepsilon_L\,.$$

For the evolution of the martensite fraction ξ_S in the material, the following rule is adopted

$$\left.\begin{array}{l} F - R_s^{AS} > 0 \\ \dot{F} > 0 \\ \xi_S < 1 \end{array}\right\} \Rightarrow \quad \xi_S = -(1-\xi_S)\frac{\dot{F}}{F - R_f^{AS}} \qquad (19.30.5)$$

$$\left.\begin{array}{l} F - R_s^{SA} < 0 \\ \dot{F} < 0 \\ \xi_S > 0 \end{array}\right\} \Rightarrow \quad \xi_S = \xi_S \frac{\dot{F}}{F - R_f^{SA}}.$$

Here R_s^{AS}, R_f^{AS}, R_s^{SA} and R_f^{SA} are constant material parameters. The Cauchy stress is finally obtained as

$$\boldsymbol{\sigma} = \boldsymbol{\tau} / J \qquad (19.30.6)$$

where J is the Jacobian of the deformation.

19.30.2 Algorithmic Stress Update

For the stress update we assume that the martensite fraction ξ_S^n and the value of the loading function F^n is known from time t_n and the deformation gradient at time t_{n+1}, $\mathbf{F}$, is known. We form the left Cauchy-Green tensor as $\mathbf{B} = \mathbf{F}\mathbf{F}^T$ which is diagonalized to obtain the principal values and directions $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_i)$ and $\mathbf{Q}$. The volumetric and principal shear logarithmic strains are given by

$$\theta = \log(J)$$
$$e_i = \log(\lambda_i / J^{1/3})$$

where

$$J = \lambda_1 \lambda_2 \lambda_3$$

is the total Jacobian of the deformation. Using Equation (19.30.4) with $\xi_S = \xi_S^n$, a value F^{trial} of the loading function can be computed. The discrete counterpart of Equation (19.30.5) becomes

$$\left.\begin{array}{l} F^{\text{trial}} - R_s^{AS} > 0 \\ F^{\text{trial}} - F^n > 0 \\ \xi_S^n < 1 \end{array}\right\} \Rightarrow \quad \Delta\xi_S = -(1-\xi_S^n - \Delta\xi_S)\frac{F^{\text{trial}} - \beta\Delta\xi_S - \min(\max(F^n, R_s^{AS}), R_f^{AS})}{F^{\text{trial}} - \beta\Delta\xi_S - R_f^{AS}}$$

(19.30.7)

$$\left.\begin{array}{l} F^{\text{trial}} - R_s^{SA} < 0 \\ F^{\text{trial}} - F^n < 0 \\ \xi_S^n > 0 \end{array}\right\} \quad \Rightarrow \quad \Delta\xi_S = (\xi_S^n + \Delta\xi_S)\frac{F^{\text{trial}} - \beta\Delta\xi_S - \min(\max(F^n, R_f^{SA}), R_s^{SA})}{F^{\text{trial}} - \beta\Delta\xi_S - R_f^{SA}}$$

If none of the two conditions to the left are satisfied, set $\xi_S^{n+1} = \xi_S^n$, $F^{n+1} = F^{\text{trial}}$ and compute the stress $\boldsymbol{\sigma}^{n+1}$ using Equations (19.30.1), (19.30.2), (19.30.3), (19.30.6) and $\xi_S = \xi_S^n$. When phase transformation occurs according to a condition to the left, the corresponding equation to the right is solved for $\Delta\xi_S$. If the bulk and shear modulii are constant this is an easy task. Otherwise F^{trial} as well as β depends on this parameter and makes things a bit more tricky. We have that

$$F^{\text{trial}} = F_n^{\text{trial}}(1 + \frac{E_S - E_A}{E_n}\Delta\xi_S)$$
$$\beta = \beta_n(1 + \frac{E_S - E_A}{E_n}\Delta\xi_S)$$

where E_S and E_A are Young's modulii for martensite and austenite, respectively. The subscript n is introduced for constant quantities evaluated at time t_n. To simplify the upcoming expressions, these relations are written

$$\begin{array}{l} F^{\text{trial}} = F_n^{\text{trial}} + \Delta F^{\text{trial}}\Delta\xi_S \\ \beta = \beta_n + \Delta\beta\ \Delta\xi_S \end{array}.$$

Inserting these expressions into Equation (19.30.7) results in

$$\begin{array}{l} f(\Delta\xi_S) = \Delta\beta(1-\xi_S^n)\Delta\xi_S^2 + (R_f^{AS} - \tilde{F}_{AS}^n + (\beta_n - \Delta F^{\text{trial}})(1-\xi_S^n))\Delta\xi_S + \\ \qquad (1-\xi_S^n)(\tilde{F}_{AS}^n - F_{\text{n}}^{\text{trial}}) = 0 \end{array} \tag{19.30.8}$$

and

$$\begin{array}{l} f(\Delta\xi_S) = \Delta\beta\xi_S^n\Delta\xi_S^2 + (\tilde{F}_{SA}^n - R_f^{SA} + (\beta_n - \Delta F^{\text{trial}})\xi_S^n)\Delta\xi_S + \\ \qquad \xi_S^n(\tilde{F}_{SA}^n - F_{\text{n}}^{\text{trial}}) = 0 \end{array}, \tag{19.30.9}$$

respectively, where we have for simplicity set

$$\begin{array}{l} \tilde{F}_{AS}^n = \min(\max(F^n, R_s^{AS}), R_f^{AS}) \\ \tilde{F}_{SA}^n = \min(\max(F^n, R_f^{SA}), R_s^{SA}) \end{array}.$$

The solutions to these equations are approximated with two Newton iterations starting in the point $\Delta\xi_S = 0$. Now set $\xi_S^{n+1} = \min(1, \max(0, \xi_S^n + \Delta\xi_S))$ and compute $\boldsymbol{\sigma}^{n+1}$ and F^{n+1} according to Equations (19.30.1), (19.30.2), (19.30.3), (19.30.4), (19.30.6) and $\xi_S = \xi_S^{n+1}$.

19.30.3 Tangent Stiffness Matrix

An algorithmic tangent stiffness matrix relating a change in true strain to a corresponding change in Kirchhoff stress is derived in the following. Taking the variation of Equation (19.30.2) results in

$$\delta p = K(\delta\theta - 3\alpha\delta\xi_S\varepsilon_L) + \delta K(\theta - 3\alpha\xi_S\varepsilon_L) \qquad (19.30.10)$$

$$\delta\mathbf{t} = 2G(\delta\mathbf{e} - \delta\xi_S\varepsilon_L\mathbf{n} - \xi_S\varepsilon_L\delta\mathbf{n}) + 2\delta G(\mathbf{e} - \xi_S\varepsilon_L\mathbf{n}).$$

The variation of the unit vector in Equation (19.30.3) can be written

$$\delta\mathbf{n} = \frac{1}{\|\mathbf{e}\| + 10^{-12}}(\mathbf{I} - \mathbf{n}\otimes\mathbf{n})\delta\mathbf{e},$$

where $\mathbf{I}$ is the fourth order identity tensor. For the variation of martensite fraction we introduce the indicator parameters H^{AS} and H^{SA} that should give information of the probability of phase transformation occurring in the next stress update. Set initially $H^{AS} = H^{SA} = 0$ and change them according to

$$\left.\begin{aligned} F^{\text{trial}} - R_s^{AS} &> 0 \\ F^{\text{trial}} - F^n &> 0 \\ \xi_S^n + \Delta\xi_S &\le 1 \end{aligned}\right\} \quad \Rightarrow \quad H^{AS} = 1$$

$$\left.\begin{aligned} F^{\text{trial}} - R_s^{SA} &< 0 \\ F^{\text{trial}} - F^n &< 0 \\ \xi_S^n + \Delta\xi_S &\ge 0 \end{aligned}\right\} \quad \Rightarrow \quad H^{SA} = 1$$

using the quantities computed in the previous stress update. For the variation of the martensite fraction we take variations of Equations (19.30.8) and (19.30.9) with

$$\delta F_n^{\text{trial}} = 2G\mathbf{n} : \delta\mathbf{e} + 3\alpha K\delta\theta$$

which results in

$$\delta\xi_S = \gamma(2G\mathbf{n} : \delta\mathbf{e} + 3\alpha K\delta\theta)$$

where

$$\gamma = \frac{(1-\xi_S^n)H^{AS}}{R_f^{AS} - \tilde{F}_{AS}^n + (\beta_n - \Delta F_n^{\text{trial}})(1-\xi_S^n)} + \frac{\xi_S^n H^{SA}}{\tilde{F}_{SA}^n - R_f^{SA} + (\beta_n - \Delta F_n^{\text{trial}})\xi_S^n}.$$

As can be seen, we use the value of γ obtained in the previous stress update since this is easier to implement and will probably give a good indication of the current value of this parameter.

The variation of the material parameters K and G results in

$$\delta K = (K_S - K_A)\delta\xi_S$$
$$\delta G = (G_S - G_A)\delta\xi_S$$

and, finally, using the identities

$$\mathbf{n} : \delta\mathbf{e} = \mathbf{n} : \delta\boldsymbol{\varepsilon}$$

$$\delta\theta = \mathbf{i} : \delta\boldsymbol{\varepsilon}$$

$$\delta\boldsymbol{\tau} = \mathbf{i}\delta p + \delta\mathbf{t}$$

results in

$$\delta\boldsymbol{\tau} = \left\{ 2G\left(1 - \frac{\xi_S\varepsilon_L}{\|\mathbf{e}\| + 10^{-12}}\right)\mathbf{I}^{\text{dev}} + K\left(1 - 9\alpha^2 K\gamma\varepsilon_L + 3\alpha\gamma(K_S - K_A)(\theta - 3\alpha\xi_S\varepsilon_L)\right)\mathbf{i}\otimes\mathbf{i} + \right.$$
$$2\gamma G(K_S - K_A)(\theta - 3\alpha\xi_S\varepsilon_L)\mathbf{i}\otimes\mathbf{n} + 6\gamma\alpha K(G_S - G_A)(\|e\| - \xi_S\varepsilon_L)\mathbf{n}\otimes\mathbf{i} +$$
$$\left. 2G\left(\frac{\xi_S\varepsilon_L}{\|\mathbf{e}\| + 10^{-12}} - 2G\gamma\varepsilon_L + 2\gamma(G_S - G_A)(\|e\| - \xi_S\varepsilon_L)\right)\mathbf{n}\otimes\mathbf{n} - 6KG\alpha\gamma\varepsilon_L(\mathbf{i}\otimes\mathbf{n} + \mathbf{n}\otimes\mathbf{i})\right\}\delta\boldsymbol{\varepsilon}$$

where $\mathbf{I}^{\text{dev}}$ is the fourth order deviatoric identity tensor. In general this tangent is not symmetric because of the terms on the second line in the expression above. We simply use a symmetrization of the tangent stiffness above in the implementation. Furthermore, we transform the tangent to a tangent closer related to the one that should be used in the LS-DYNA implementation,

$$\mathbf{C} = J^{-1}\left\{ 2G\left(1 - \frac{\xi_S\varepsilon_L}{\|\mathbf{e}\| + 10^{-12}}\right)\mathbf{I}^{\text{dev}} + K\left(1 - 9\alpha^2 K\gamma\varepsilon_L + 3\alpha\gamma(K_S - K_A)(\theta - 3\alpha\xi_S\varepsilon_L)\right)\mathbf{i}\otimes\mathbf{i} + \right.$$
$$\left(\gamma G(K_S - K_A)(\theta - 3\alpha\xi_S\varepsilon_L) + 3\gamma\alpha K(G_S - G_A)(\|e\| - \xi_S\varepsilon_L) - 6KG\alpha\gamma\varepsilon_L\right)(\mathbf{i}\otimes\mathbf{n} + \mathbf{n}\otimes\mathbf{i}) +$$
$$\left. 2G\left(\frac{\xi_S\varepsilon_L}{\|\mathbf{e}\| + 10^{-12}} - 2G\gamma\varepsilon_L + 2\gamma(G_S - G_A)(\|e\| - \xi_S\varepsilon_L)\right)\mathbf{n}\otimes\mathbf{n}\right\}\delta\boldsymbol{\varepsilon}$$

Material Model 31: Slightly Compressible Rubber Model

This model implements a modified form of the hyperelastic constitutive law first described in [Kenchington 1988].

The strain energy functional, U, is defined in terms of the input constants as:

$$U = C_{100}\, I_1 + C_{200}\, I_1^{\,2} + C_{300}\, I_1^{\,3} + C_{400}\, I_1^{\,4} + C_{110} I_1 I_2 + C_{210} I_1^{\,2} I_2 + C_{010}\, I_2 + C_{020} I_2^{\,2} + f(J) \tag{19.31.1}$$

where the strain invariants can be expressed in terms of the deformation gradient matrix, F_{ij}, and the Green-St. Venant strain tensor, E_{ij}:

$$\begin{aligned} J &= \left|F_{ij}\right| \\ I_1 &= E_{ii} \\ I_2 &= \frac{1}{2!}\delta^{ij}_{pq} E_{pi} E_{qj} \end{aligned} \tag{19.31.2}$$

The derivative of U with respect to a component of strain gives the corresponding component of stress

$$S_{ij} = \frac{\partial U}{\partial E_{ij}} \tag{19.31.3}$$

where, S_{ij}, is the second Piola-Kirchhoff stress tensor which is transformed into the Cauchy stress tensor:

$$\sigma_{ij} = \frac{\rho}{\rho_0}\frac{\partial x_i}{\partial X_k}\frac{\partial x_j}{\partial X_l} S_{kl} \tag{19.31.4}$$

where ρ_0 and ρ are the initial and current density, respectively.

Material Model 32: Laminated Glass Model

This model is available for modeling safety glass. Safety glass is a layered material of glass bonded to a polymer material which can undergo large strains.

The glass layers are modeled by isotropic hardening plasticity with failure based on exceeding a specified level of plastic strain. Glass is quite brittle and cannot withstand large strains before failing. Plastic strain was chosen for failure since it increases monotonically and, therefore, is insensitive to spurious numerical noise in the solution.

The material to which the glass is bonded is assumed to stretch plastically without failure. The user defined integration rule option must be used with this material. The user defined rule specifies the thickness of the layers making up the safety glass. Each integration point is flagged with a zero if the layer is glass and with a one if the layer is polymer.

An iterative plane stress plasticity algorithm is used to enforce the plane stress condition.

Material Model 33: Barlat's Anisotropic Plasticity Model

This model was developed by Barlat, Lege, and Brem [1991] for modeling material behavior in forming processes. The finite element implementation of this model is described in detail by Chung and Shah [1992] and is used here.

The yield function Φ is defined as

$$\Phi = |S_1 - S_2|^m + |S_2 - S_3|^m + |S_3 - S_1|^m = 2^m \tag{19.33.1}$$

where $\bar{\sigma}$ is the effective stress, and S_i *for* $i = 1,2,3$ are the principal values of the symmetric matrix $S_{\alpha\beta}$,

$$\begin{aligned} S_{xx} &= [c(\sigma_{xx} - \sigma_{yy}) - b(\sigma_{zz} - \sigma_{xx})]/3 \\ S_{yy} &= [a(\sigma_{yy} - \sigma_{zz}) - c(\sigma_{xx} - \sigma_{yy})]/3 \\ S_{zz} &= [b(\sigma_{zz} - \sigma_{xx}) - a(\sigma_{yy} - \sigma_{zz})]/3 \\ S_{yz} &= f\sigma_{yz} \\ S_{zx} &= g\sigma_{zx} \\ S_{xy} &= h\sigma_{xy} \end{aligned} \tag{19.33.2}$$

The material constants a, b, c, f, g and h represent anisotropic properties. When $a = b = c = f = g = h = 1$, the material is isotropic and the yield surface reduces to the Tresca yield surface for m=1 and von Mises yield surface for m= 2 or 4. For face-centered-cubic (FCC) materials m=8 is recommended and for body-centered-cubic (BCC) materials m=6 is used.

The yield strength of the material is

$$\sigma_y = k(1 + \varepsilon_0)^n \tag{19.33.3}$$

Material Model 34: Fabric

The fabric model is a variation on the Layered Orthotropic Composite material model (Material 22) and is valid for only 3 and 4 node membrane elements. This material model is *strongly* recommended for modeling airbags and seatbelts. In addition to being a constitutive model, this model also invokes a special membrane element formulation that is better suited to the large deformations experienced by fabrics. For thin fabrics, buckling (wrinkling) can occur with the associated inability of the structure to support compressive stresses; a material parameter flag is included for this option. A linear elastic liner is also included which can be used to reduce the tendency for these material/elements to be crushed when the no-compression option is invoked.

If the airbag material is to be approximated as an isotropic elastic material, then only one Young's modulus and Poisson's ratio should be defined. The elastic approximation is very efficient because the local transformations to the material coordinate system may be skipped. If orthotropic constants are defined, it is very important to consider the orientation of the local material system and use great care in setting up the finite element mesh.

If the reference configuration of the airbag is taken as the folded configuration, the geometrical accuracy of the deployed bag will be affected by both the stretching and the compression of elements during the folding process. Such element distortions are very difficult to avoid in a folded bag. By reading in a reference configuration such as the final unstretched configuration of a deployed bag, any distortions in the initial geometry of the folded bag will have no effect on the final geometry of the inflated bag. This is because the stresses depend only on the deformation gradient matrix:

$$F_{ij} = \frac{\partial x_i}{\partial X_j} \tag{19.34.1}$$

where the choice of X_j may coincide with the folded or unfold configurations. It is this unfolded configuration which may be specified here. When the reference geometry is used then the no-compression option should be active. With the reference geometry it is possible to shrink the airbag and then perform the inflation. Although the elements in the shrunken bag are very small, the time step can be based on the reference geometry so a very reasonable time step size is obtained. The reference geometry based time step size is optional in the input.

The parameters fabric leakage coefficient, FLC, fabric area coefficient, FAC, and effective leakage area, ELA, for the fabric in contact with the structure are optional for the Wang-Nefske and hybrid inflation models. It is possible for the airbag to be constructed of multiple fabrics having different values of porosity and permeability. The gas leakage through the airbag fabric then requires an accurate determination of the areas by part ID available for leakage. The leakage area may change over time due to stretching of the airbag fabric or blockage when the outer surface of the bag is in contact with the structure. LS-DYNA can check the interaction of the bag with the structure and split the areas into regions that are blocked and unblocked depending on whether the regions are in contact or not, respectively. Typically, the parameters, FLC and FAC, must be determined experimentally and their variation with time and pressure are optional inputs that allow for maximum modeling flexibility.

Material Model 35: Kinematic/Isotropic Plastic Green-Naghdi Rate

The reader interested in an detailed discussion of the relative merits of various stress rates, especially Jaumann [1911] and Green-Naghdi [1965], is urged to read the work of Johnson and Bammann [1984]. A mathematical description of these two stress rates, and how they are implemented in LS-DYNA, is given in the section entitled *Stress Update Overview* in this manual.

In the cited work by Johnson and Bammann, they conclude that the Green-Naghdi stress rate is to be preferred over all other proposed stress rates, including the most widely used Jaumann rate, because the Green-Naghdi stress rate is based on the notions of invariance under superimposed rigid-body motions. However, implementation of the Green-Naghdi stress rate comes at a significant computational cost compared to the Jaumann stress rate, e.g., see the discussion in this manual in the section entitled *Green-Naghdi Stress Rate*.

Perhaps more importantly, in practical applications there is little if any noted difference in the results generated using either Jaumann or Green-Naghdi stress rate. This is in part due to

the fact that the Jaumann stress rate only produces erroneous results[1] when linear kinematic hardening is used; the results for isotropic hardening are not affected by the choice of either of these two stress rates. Also in practical applications, the shear strains are rather small, compared to the extensional strains, and if they are not small it is usually the case that the material description, i.e., constitutive model, is not valid for such large shear strains.

Material Model 36: Barlat's 3-Parameter Plasticity Model

Material model 36 in LS-DYNA aims at modeling sheets with anisotropy under plane stress conditions. It allows the use of the Lankford parameters for the definition of the anisotropic yield surface. The yield condition can be written

$$f(\boldsymbol{\sigma}, \varepsilon_p) = \sigma_{\text{eff}}(\sigma_{11}, \sigma_{22}, \sigma_{12}) - \sigma_{\text{Y}}(\varepsilon_p) \le 0 \qquad (19.36.1)$$

where

$$\sigma_{\text{eff}}(\sigma_{11}, \sigma_{22}, \sigma_{12}) = \left(\frac{a}{2}\left|K_1 + K_2\right|^m + \frac{a}{2}\left|K_1 - K_2\right|^m + \frac{c}{2}\left|2K_2\right|^m \right)^{1/m}$$

$$K_1 = K_1(\sigma_{11}, \sigma_{22}, \sigma_{12}) = \frac{\sigma_{11} + h\sigma_{22}}{2}$$

$$K_2 = K_2(\sigma_{11}, \sigma_{22}, \sigma_{12}) = \sqrt{\left(\frac{\sigma_{11} - h\sigma_{22}}{2}\right)^2 + p^2\sigma_{12}^2}$$

and the hardening of the yield surface is either linear, exponential or determined by a load curve. In the above, the stress components σ_{11}, σ_{22} and σ_{12} are with respect to the material coordinate system and ε_p denotes the effective plastic strain. The material parameters a, c, h and p can be determined from the Lankford parameters as described in the LS-DYNA Keyword User's Manual [Hallquist 2003]. The gradient of the yield surface is denoted

$$\frac{\partial f}{\partial \boldsymbol{\sigma}}(\boldsymbol{\sigma}) = \begin{pmatrix} \dfrac{\partial f}{\partial \sigma_{11}}(\boldsymbol{\sigma}) \\ \dfrac{\partial f}{\partial \sigma_{22}}(\boldsymbol{\sigma}) \\ \dfrac{\partial f}{\partial \sigma_{12}}(\boldsymbol{\sigma}) \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \dfrac{\partial f}{\partial \sigma_{11}}(\sigma_{11}, \sigma_{22}, \sigma_{12}) \\ \dfrac{\partial f}{\partial \sigma_{22}}(\sigma_{11}, \sigma_{22}, \sigma_{12}) \\ \dfrac{\partial f}{\partial \sigma_{12}}(\sigma_{11}, \sigma_{22}, \sigma_{12}) \\ 0 \\ 0 \end{pmatrix}$$

where

[1] The results of a simple shear simulation, monotonically increasing shear deformation, produce sinusoidal stress response.

$$\frac{\partial f}{\partial \sigma_{11}}(\sigma_{11},\sigma_{22},\sigma_{12}) = \frac{\partial \sigma_{\text{eff}}}{\partial \sigma_{11}}(\sigma_{11},\sigma_{22},\sigma_{12}) = \frac{\sigma_{\text{eff}}(\sigma_{11},\sigma_{22},\sigma_{12})^{1-m}}{2}\cdot$$
$$\left\{a(K_1-K_2)\left|K_1-K_2\right|^{m-2}\left(\frac{1}{2}-\frac{\sigma_{11}-h\sigma_{22}}{4K_2}\right)+\right.$$
$$a(K_1+K_2)\left|K_1+K_2\right|^{m-2}\left(\frac{1}{2}+\frac{\sigma_{11}-h\sigma_{22}}{4K_2}\right)+$$
$$\left.c2^m K_2^{m-1}\frac{\sigma_{11}-h\sigma_{22}}{4K_2}\right\}$$

$$\frac{\partial f}{\partial \sigma_{22}}(\sigma_{11},\sigma_{22},\sigma_{12}) = \frac{\partial \sigma_{\text{eff}}}{\partial \sigma_{22}}(\sigma_{11},\sigma_{22},\sigma_{12}) = \frac{\sigma_{\text{eff}}(\sigma_{11},\sigma_{22},\sigma_{12})^{1-m}}{2}h\cdot$$
$$\left\{a(K_1-K_2)\left|K_1-K_2\right|^{m-2}\left(\frac{1}{2}+\frac{\sigma_{11}-h\sigma_{22}}{4K_2}\right)+\right.$$
$$a(K_1+K_2)\left|K_1+K_2\right|^{m-2}\left(\frac{1}{2}-\frac{\sigma_{11}-h\sigma_{22}}{4K_2}\right)-$$
$$\left.c2^m K_2^{m-1}\frac{\sigma_{11}-h\sigma_{22}}{4K_2}\right\}$$

and

$$\frac{\partial f}{\partial \sigma_{12}}(\sigma_{11},\sigma_{22},\sigma_{12}) = \frac{\partial \sigma_{\text{eff}}}{\partial \sigma_{12}}(\sigma_{11},\sigma_{22},\sigma_{12}) = \frac{\sigma_{\text{eff}}(\sigma_{11},\sigma_{22},\sigma_{12})^{1-m}}{2}\frac{p^2\sigma_{12}}{K_2}\cdot$$
$$\left\{-a(K_1-K_2)\left|K_1-K_2\right|^{m-2}+a(K_1+K_2)\left|K_1+K_2\right|^{m-2}+c2^m K_2^{m-1}\right\}$$

For one of the implementations described in the next section, a three-dimensional formulation is used. The yield condition is generalized to include the normal stress as follows

$$\tilde{f}(\boldsymbol{\sigma},\varepsilon_p) = \sigma_{\text{eff}}(\sigma_{11}-\sigma_{33},\sigma_{22}-\sigma_{33},\sigma_{12}) - \sigma_{\text{Y}}(\varepsilon_p) \le 0\,. \qquad (19.36.2)$$

This generalization keeps the convexity of the yield surface and it reduces to Equation (19.36.1) for a plane state of stress. The gradient of the yield surface is for this effective stress determined from

$$\frac{\partial \tilde{f}}{\partial \boldsymbol{\sigma}}(\boldsymbol{\sigma}) = \begin{pmatrix} \frac{\partial \tilde{f}}{\partial \sigma_{11}}(\boldsymbol{\sigma}) \\ \frac{\partial \tilde{f}}{\partial \sigma_{22}}(\boldsymbol{\sigma}) \\ \frac{\partial \tilde{f}}{\partial \sigma_{33}}(\boldsymbol{\sigma}) \\ \frac{\partial \tilde{f}}{\partial \sigma_{12}}(\boldsymbol{\sigma}) \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{\partial f}{\partial \sigma_{11}}(\sigma_{11}-\sigma_{33}, \sigma_{22}-\sigma_{33}, \sigma_{12}) \\ \frac{\partial f}{\partial \sigma_{22}}(\sigma_{11}-\sigma_{33}, \sigma_{22}-\sigma_{33}, \sigma_{12}) \\ -\frac{\partial f}{\partial \sigma_{11}}(\sigma_{11}-\sigma_{33}, \sigma_{22}-\sigma_{33}, \sigma_{12}) - \frac{\partial f}{\partial \sigma_{22}}(\sigma_{11}-\sigma_{33}, \sigma_{22}-\sigma_{33}, \sigma_{12}) \\ \frac{\partial f}{\partial \sigma_{12}}(\sigma_{11}-\sigma_{33}, \sigma_{22}-\sigma_{33}, \sigma_{12}) \\ 0 \\ 0 \end{pmatrix}$$

19.36.1 Material Tangent Stiffness

Since the plastic model is associative, the general expression for tangent relating the total strain rate to total stress rate can be found in standard textbooks. Since this situation is rather special we derive it here for the plane stress model presented in the previous section. The elastic stress-strain relation can be written

$$\dot{\boldsymbol{\sigma}} = \begin{pmatrix} \dot{\sigma}_{11} \\ \dot{\sigma}_{22} \\ \dot{\sigma}_{12} \\ \dot{\sigma}_{23} \\ \dot{\sigma}_{13} \end{pmatrix} = \frac{E}{1-\nu^2} \begin{pmatrix} 1 & \nu & & & \\ \nu & 1 & & & \\ & & \frac{1-\nu}{2} & & \\ & & & \frac{1-\nu}{2} & \\ & & & & \frac{1-\nu}{2} \end{pmatrix} \begin{pmatrix} \dot{\varepsilon}_{11} - \dot{\varepsilon}_{11}^p \\ \dot{\varepsilon}_{22} - \dot{\varepsilon}_{22}^p \\ 2(\dot{\varepsilon}_{12} - \dot{\varepsilon}_{12}^p) \\ 2(\dot{\varepsilon}_{23} - \dot{\varepsilon}_{23}^p) \\ 2(\dot{\varepsilon}_{13} - \dot{\varepsilon}_{13}^p) \end{pmatrix} = \mathbf{C}^{\text{ps}}(\dot{\boldsymbol{\varepsilon}} - \dot{\boldsymbol{\varepsilon}}^p) \quad (19.36.3)$$

where E is the Young's modulus, ν is the Poisson's ratio and $\mathbf{C}^{\text{ps}}$ denotes the plane stress elastic tangential stiffness matrix. The associative flow rule for the plastic strain can be written

$$\dot{\varepsilon}^p = \dot{\lambda} \frac{\partial f}{\partial \sigma} \quad (19.36.4)$$

and the consistency condition results in

$$\frac{\partial f}{\partial \sigma}^T \dot{\sigma} + \frac{\partial f}{\partial \varepsilon_{\text{p}}} \dot{\varepsilon}_p = 0 . \quad (19.36.5)$$

For algorithmic consistency, the effective plastic strain rate is defined as $\dot{\varepsilon}_p = \dot{\lambda}$. Multiplying Equation (19.36.3) with $\frac{\partial f}{\partial \sigma}$ and using Equation (19.36.4) and Equation (19.36.5) gives

$$\dot{\lambda} = \frac{\left\{\frac{\partial f}{\partial \sigma}\right\}^T \mathrm{C}^{\mathrm{ps}} \dot{\varepsilon}}{\left\{\frac{\partial f}{\partial \sigma}\right\}^T \mathrm{C}^{\mathrm{ps}} \left\{\frac{\partial f}{\partial \sigma}\right\} - \frac{\partial f}{\partial \varepsilon_p}}.$$

Inserting

$$\dot{\varepsilon}^p = \frac{\left\{\frac{\partial f}{\partial \sigma}\right\}^T \mathrm{C}^{\mathrm{ps}} \dot{\varepsilon}}{\left\{\frac{\partial f}{\partial \sigma}\right\}^T \mathrm{C}^{\mathrm{ps}} \left\{\frac{\partial f}{\partial \sigma}\right\} - \frac{\partial f}{\partial \varepsilon_p}} \left\{\frac{\partial f}{\partial \sigma}\right\}$$

into Equation (19.36.3) results in

$$\dot{\sigma} = \left(\mathrm{C}^{\mathrm{ps}} - \frac{\left\{\mathrm{C}^{\mathrm{ps}} \left\{\frac{\partial f}{\partial \sigma}\right\}\right\} \left\{\mathrm{C}^{\mathrm{ps}} \left\{\frac{\partial f}{\partial \sigma}\right\}\right\}^T}{\left\{\frac{\partial f}{\partial \sigma}\right\}^T \mathrm{C}^{\mathrm{ps}} \left\{\frac{\partial f}{\partial \sigma}\right\} - \frac{\partial f}{\partial \varepsilon_p}} \right) \dot{\varepsilon}.$$

To get the elastic-plastic tangent stiffness tensor in the element coordinate system it needs to be transformed back. Since the elastic tangential stiffness tensor is isotropic with respect to the axis of rotation, the plastic tangent stiffness tensor can be written

$$\mathrm{C}^{\mathrm{ps}}_{\mathrm{plastic}} = \left(\mathrm{C}^{\mathrm{ps}} - \frac{\left\{\mathrm{QC}^{\mathrm{ps}} \left\{\frac{\partial f}{\partial \sigma}\right\}\right\} \left\{\mathrm{QC}^{\mathrm{ps}} \left\{\frac{\partial f}{\partial \sigma}\right\}\right\}}{\left\{\frac{\partial f}{\partial \sigma}\right\}^T \mathrm{C}^{\mathrm{ps}} \left\{\frac{\partial f}{\partial \sigma}\right\} - \frac{\partial f}{\partial \varepsilon_p}} \right)$$

where Q is the rotation matrix in Voigt form.

Material Type 37: Transversely Anisotropic Elastic-Plastic

This fully iterative plasticity model is available only for shell elements. The input parameters for this model are: Young's modulus E; Poisson's ratio υ; the yield stress; the tangent modulus E_t; and the anisotropic hardening parameter R.

Consider Cartesian reference axes which are parallel to the three symmetry planes of anisotropic behavior. Then the yield function suggested by Hill [1948] can be written

$$\begin{aligned} &F\left(\sigma_{22}-\sigma_{33}\right)^2 + G\left(\sigma_{33}-\sigma_{11}\right)^2 + H\left(\sigma_{11}-\sigma_{22}\right)^2 \\ &+2L\sigma_{23}^2 + 2M\sigma_{31}^2 + 2N\sigma_{12}^2 - 1 = 0 \end{aligned} \qquad (19.37.1)$$

where σ_{y1}, σ_{y2}, and σ_{y3}, are the tensile yield stresses and σ_{12}, σ_{23}, and σ_{y31} are the shear yield stresses. The constants *F, G, H, L, M,* and *N* are related to the yield stress by

$$\begin{aligned} 2L &= \frac{1}{\sigma_{23}^2} \\ 2M &= \frac{1}{\sigma_{y31}^2} \\ 2N &= \frac{1}{\sigma_{y12}^2} \\ 2F &= \frac{1}{\sigma_{y2}^2} + \frac{1}{\sigma_{y3}^2} - \frac{1}{\sigma_{y1}^2} \\ 2G &= \frac{1}{\sigma_{y3}^2} + \frac{1}{\sigma_{y1}^2} - \frac{1}{\sigma_{y2}^2} \\ 2H &= \frac{1}{\sigma_{y1}^2} + \frac{1}{\sigma_{y2}^2} - \frac{1}{\sigma_{y3}^2}. \end{aligned} \tag{19.37.2}$$

The isotropic case of von Mises plasticity can be recovered by setting

$$F = G = H = \frac{1}{2\sigma_y^2}$$

and

$$L = M = N = \frac{1}{2\sigma_y^2} \ .$$

For the particular case of transverse anisotropy, where properties do not vary in the x_1-x_2 plane, the following relations hold:

$$\begin{aligned} 2F &= 2G = \frac{1}{\sigma_{y3}^2} \\ 2H &= \frac{2}{\sigma_y^2} - \frac{1}{\sigma_{y3}^2} \\ N &= \frac{2}{\sigma_y^2} - \frac{1}{2}\frac{1}{\sigma_{y3}^2} \end{aligned} \tag{19.37.3}$$

where it has been assumed that $\sigma y_1 = \sigma y_2 = \sigma y$.

Letting $K = \frac{\sigma_y}{\sigma_{y3}}$, the yield criterion can be written

$$F(\sigma) = \sigma_e = \sigma_y \ , \tag{19.37.4}$$

where

$$F(\sigma) \equiv \Big[\sigma_{11}^2 + \sigma_{22}^2 + K^2\sigma_{33}^2 - K^2\sigma_{33}(\sigma_{11}+\sigma_{22}) - (2-K^2)\sigma_{11}\sigma_{22} + 2L\sigma_y^2(\sigma_{23}^2+\sigma_{31}^2) + 2\left(2-\frac{1}{2}K^2\right)\sigma_{12}^2 \Big]^{1/2}$$

The rate of plastic strain is assumed to be normal to the yield surface so $\dot{\varepsilon}_{ij}^p$ is found from

$$\dot{\varepsilon}_{ij}^p = \lambda \frac{\partial F}{\partial \sigma_{ij}}. \tag{19.37.5}$$

Now consider the case of plane stress, where $\sigma_{33} = 0$. Also, define the anisotropy input parameter R as the ratio of the in-plane plastic strain rate to the out-of-plane plastic strain rate:

$$R = \frac{\dot{\varepsilon}_{22}^p}{\dot{\varepsilon}_{33}^p}. \tag{19.37.6}$$

It then follows that

$$R = \frac{2}{K^2} - 1. \tag{19.37.7}$$

Using the plane stress assumption and the definition of R, the yield function may now be written

$$F(\sigma) = \left[\sigma_{11}^2 + \sigma_{22}^2 - \frac{2R}{R+1}\sigma_{11}\sigma_{22} + 2\frac{2R+1}{R+1}\sigma_{12}^2 \right]^{1/2} \tag{19.37.8}$$

Material Type 38: Blatz-Ko Compressible Foam

$$W(I_1, I_2, I_3) = \frac{\mu}{2}\left(\frac{I_2}{I_3} + 2\sqrt{I_3} - 5 \right) \tag{19.38.1}$$

where μ is the shear modulus and I_1, I_2, and I_3 are the strain invariants. Blatz and Ko [1962] suggested this form for a 47 percent volume polyurethane foam rubber with a Poisson's ratio of 0.25. The second Piola-Kirchhoff stresses are given as

$$S^{ij} = \mu\left[\left(I\delta_{ij} - G_{ij}\right)\frac{1}{I_3} + \left(\sqrt{I_3} - \frac{I_2}{I_3}\right)G^{ij}\right] \qquad (19.38.2)$$

where

$$G_{ij} = \frac{\partial x_k}{\partial X_i}\frac{\partial x_k}{\partial X_j}$$

$$G^{ij} = \frac{\partial X_i}{\partial x_k}\frac{\partial X_j}{\partial x_k}$$

after determining S_{ij}, it is transformed into the Cauchy stress tensor:

$$\sigma_{ij} = \frac{\rho}{\rho_0}\frac{\partial x_i}{\partial X_k}\frac{\partial x_j}{\partial X_l}S_{kl}$$

where ρ_0 and ρ are the initial and current density, respectively.

Material Model 39: Transversely Anisotropic Elastic-Plastic With FLD

See Material Model 37 for the similar model theoretical basis. The first history variable is the maximum strain ratio defined by:

$$\frac{\varepsilon_{major_{workpiece}}}{\varepsilon_{major_{fld}}} \qquad (19.39.1)$$

corresponding to $\varepsilon_{minor_{workpiece}}$. This history variable replaces the effective plastic strain in the output. Plastic strains can still be obtained but one additional history variable must be written into the D3PLOT database.

The strains on which these calculations are based are integrated in time from the strain rates:

$$\varepsilon_{ij}^{n+1} = \varepsilon_{ij}^{n} + \varepsilon_{ij}^{\nabla n+1/2}\Delta t^{n+1/2} \qquad (19.39.2)$$

and are stored as history variables. The resulting strain measure is logarithmic.

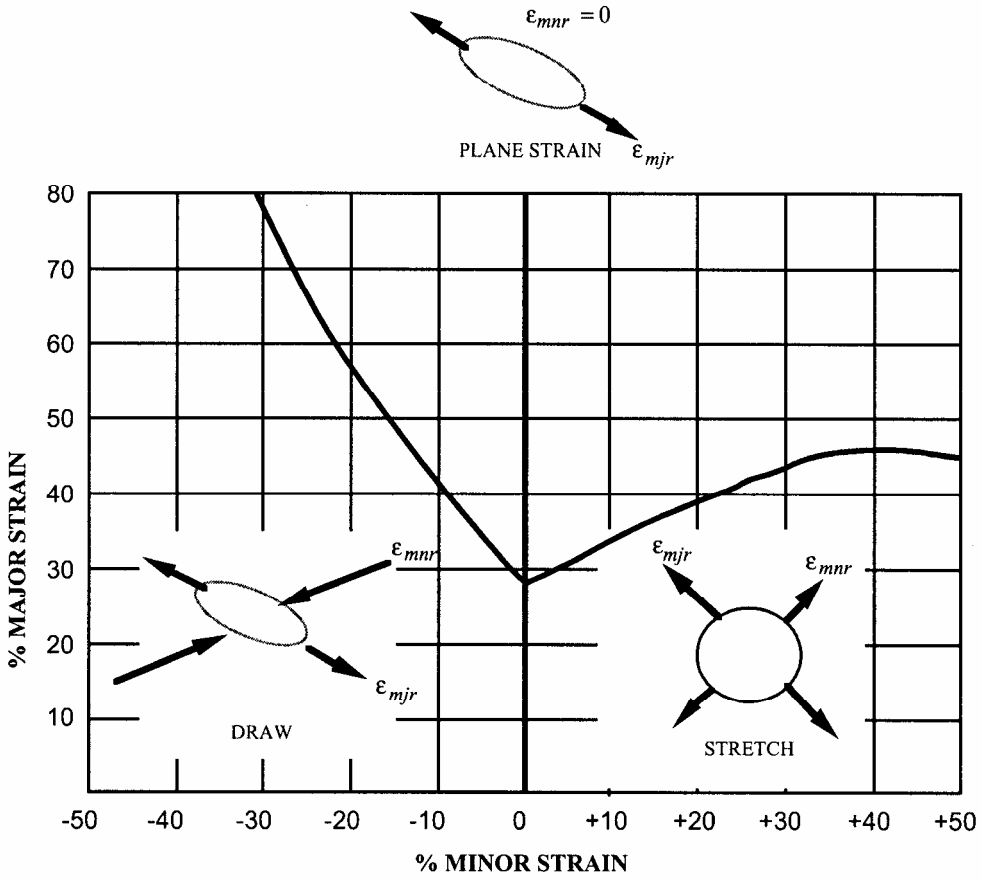


Figure 19.39.1. Flow limit diagram.

Material Model 42: Planar Anisotropic Plasticity Model

This model is built into LS-DYNA as a user material model for modeling plane stress anisotropic plasticity in shells. Please note that only three cards are input here. The orthotropic angles must be defined later as for all materials of this type. This model is currently not vectorized.

This is an implementation of an anisotropic plasticity model for plane stress where the flow rule, see Material Type 37, simplifies to:

$$F\left(\sigma_{22}\right)^2 + G\left(\sigma_{11}\right)^2 + H\left(\sigma_{11} - \sigma_{22}\right)^2 + 2N\sigma_{12}^2 - 1 = 0 \ . \qquad (19.42.1)$$

The anisotropic parameters R00, R45, and R90 are defined in terms of F, G, H, and N as [Hill, 1989]:

$$2R00 = \frac{H}{G} \qquad (19.42.2)$$

$$2R45 = \frac{2N}{(F+G)} - 1.$$
$$2R90 = \frac{H}{F} \tag{19.42.3}$$

The yield function for this model is given as:

$$\sigma_y = A\varepsilon^m \dot{\varepsilon}^n \tag{19.42.4}$$

To avoid numerical problems the minimum strain rate, $\dot{\varepsilon}_{\min}$ must be defined and the initial yield stress σ_0 is calculated as

$$\sigma_0 = A\varepsilon_0^m \dot{\varepsilon}_{\min}^n = E\varepsilon_0 \tag{19.42.5}$$

$$\varepsilon_0 = \left(\frac{E}{A\dot{\varepsilon}_{\min}^n} \right)^{\frac{1}{m-1}} . \tag{19.42.6}$$

Material Model 51: Temperature and Rate Dependent Plasticity

The kinematics associated with the model are discussed in references [Hill 1948, Bammann and Aifantis 1987, Bammann 1989]. The description below is taken nearly verbatim from Bammann [Hill 1948].

With the assumption of linear elasticity we can write,

$$\overset{o}{\sigma} = \lambda \operatorname{tr}\left(D^e\right)1 + 2\mu D^e \tag{19.51.1}$$

where, the Cauchy stress σ is convected with the elastic spin W^e as,

$$\overset{o}{\sigma} = \dot{\sigma} - W^e\sigma + \sigma W^e \tag{19.51.2}$$

This is equivalent to writing the constitutive model with respect to a set of directors whose direction is defined by the plastic deformation [Bammann and Aifantis 1987, Bammann and Johnson 1987]. Decomposing both the skew symmetric and symmetric parts of the velocity gradient into elastic and plastic parts we write for the elastic stretching D^e and the elastic spin W^e,

$$D^e = D - D^p - D^{th}, W^e = W = W^p. \tag{19.51.3}$$

Within this structure it is now necessary to prescribe an equation for the plastic spin W^p in addition to the normally prescribed flow rule for D^p and the stretching due to the thermal expansion D^{th}. As proposed, we assume a flow rule of the form,

$$D^p = f(T)\sinh\left[\frac{|\xi| - \kappa - Y(T)}{V(T)}\right]\frac{\xi'}{|\xi'|}. \qquad (19.51.4)$$

Where T is the temperate, κ is the scalar hardening variable, ξ' is the difference between the deviatoric Cauchy stress σ' and the tensor variable α',

$$\xi' = \sigma' - \alpha' \qquad (19.51.5)$$

and $f(T), Y(T), V(T)$ are scalar functions whose specific dependence upon the temperature is given below. Assuming isotropic thermal expansion, and introducing the expansion coefficient $\dot{A}$, the thermal stretching can be written,

$$D^{th} = \dot{A}\dot{T}1. \qquad (19.51.6)$$

The evolution of the internal variables α and κ are prescribed in a hardening minus recovery format as,

$$\overset{o}{\alpha} = h(T)D^p - \left[r_d(T)\left|D^p\right| + r_s(T)\right]|\alpha|\alpha, \qquad (19.51.7)$$

$$\dot{\kappa} = H(T)D^p - \left[R_d(T)\left|D^p\right| - R_s(T)\right]\kappa^2 \qquad (19.51.8)$$

where h and H are the hardening moduli, $r_s(T)$ and $R_s(T)$ are scalar functions describing the diffusion controlled 'static' or 'thermal' recovery, and $r_d(T)$ and $R_d(T)$ are the functions describing dynamic recovery.

If we assume that $W^p = 0$, we recover the Jaumann stress rate which results in the prediction of an oscillatory shear stress response in simple shear when coupled with a Prager kinematic hardening assumption [Johnson and Bammann 1984]. Alternatively we can choose,

$$W^p = R^T\dot{U}U^{-1}R, \qquad (19.51.9)$$

which recovers the Green-Naghdi rate of Cauchy stress and has been shown to be equivalent to Mandel's isoclinic state [Bammann and Aifantis 1987]. The model employing this rate allows a reasonable prediction of directional softening for some materials but in general under-predicts the softening and does not accurately predict the axial stresses which occur in the torsion of the thin walled tube.

The final equation necessary to complete our description of high strain rate deformation is one which allows us to compute the temperature change during the deformation. In the absence of a coupled thermomechanical finite element code we assume adiabatic temperature change and follow the empirical assumption that 90 - 95% of the plastic work is dissipated as heat. Hence,

$$\dot{T} = \frac{.9}{\rho C_v}\left(\sigma \cdot D^p\right), \tag{19.51.10}$$

where ρ is the density of the material and C_v the specific heat.

In terms of the input parameters the functions defined above become:

V(T) = C1 exp(-C2/T)	h(T) = C9 exp(C10/T)
Y(T) = C3 exp(C4/T)	rs(T) = C11exp(-C12/T)
f(T) = C5 exp(-C6/T)	RD(T) = C13exp(-C14/T)
rd(T) = C7 exp(-C8/T)	H(T) = C15exp(C16/T)
	RS(T) = C17exp(-C18/T)

and the heat generation coefficient is

$$HC = \frac{.9}{\rho C_V}. \tag{19.51.11}$$

Material Model 52: Sandia's Damage Model

The evolution of the damage parameter, ϕ, is defined by [Bammann, et. al., 1990]

$$\dot{\phi} = \beta\left[\frac{1}{(1-\phi)^N} - (1-\phi)\right]^{|D^p|} \tag{19.52.1}$$

in which

$$\beta = \sinh\left[\frac{2(2N-1)\,p}{(2N-1)\,\bar{\sigma}}\right] \tag{19.52.2}$$

where p is the pressure and $\bar{\sigma}$ is the effective stress.

Material Model 53: Low Density Closed Cell Polyurethane Foam

A rigid, low density, closed cell, polyurethane foam model developed at Sandia Laboratories [Neilsen et al., 1987] has been recently implemented for modeling impact limiters in automotive applications. A number of such foams were tested at Sandia and reasonable fits to the experimental data were obtained.

In some respects this model is similar to the crushable honeycomb model type 26 in that the components of the stress tensor are uncoupled until full volumetric compaction is achieved.

However, unlike the honeycomb model this material possesses no directionality but includes the effects of confined air pressure in its overall response characteristics.

$$\sigma_{ij} = \sigma_{ij}^{sk} - \delta_{ij}\sigma^{air} \tag{19.53.1}$$

where σ_{ij}^{sk} is the skeletal stress and σ^{air} is the air pressure computed from the equation:

$$\sigma^{air} = -\frac{p_0\gamma}{1+\gamma-\phi} \tag{19.53.2}$$

where p_0 is the initial foam pressure usually taken as the atmospheric pressure and γ defines the volumetric strain

$$\gamma = V - 1 + \gamma_0 \tag{19.53.3}$$

where V is the relative volume and γ_0 is the initial volumetric strain which is typically zero. The yield condition is applied to the principal skeletal stresses which are updated independently of the air pressure. We first obtain the skeletal stresses:

$$\sigma_{ij}^{sk} = \sigma_{ij} + \sigma_{ij}\,\sigma^{air} \tag{19.53.4}$$

and compute the trial stress, σ_i^{skt}

$$\sigma_{ij}^{skt} = \sigma_{ij}^{sk} + E\,\dot{\varepsilon}_{ij}\,\Delta t \tag{19.53.5}$$

where E is Young's modulus. Since Poisson's ratio is zero, the update of each stress component is uncoupled and $2G = E$ where G is the shear modulus. The yield condition is applied to the principal skeletal stresses such that if the magnitude of a principal trial stress component, σ_i^{skt}, exceeds the yield stress, σ_y, then

$$\sigma_i^{sk} = \min(\sigma_y, \left|\sigma_i^{skt}\right|)\frac{\sigma_i^{skt}}{\left|\sigma_i^{skt}\right|} \quad . \tag{19.53.6}$$

The yield stress is defined by

$$\sigma_y = a + b(1 + c\gamma) \tag{19.53.7}$$

where a, b, and c are user defined input constants. After scaling the principal stresses they are transformed back into the global system and the final stress state is computed

$$\sigma_{ij} = \sigma_{ij}^{sk} - \delta_{ij}\sigma^{air} \, . \tag{19.53.8}$$

Material Models 54 and 55: Enhanced Composite Damage Model

These models are very close in their formulations. Material 54 uses the Chang matrix failure criterion (as Material 22), and material 55 uses the Tsay-Wu criterion for matrix failure.

Arbitrary orthothropic materials, e.g., unidirectional layers in composite shell structures can be defined. Optionally, various types of failure can be specified following either the suggestions of [Chang and Chang, 1984] or [Tsai and Wu, 1981]. In addition special measures are taken for failure under compression. See [Matzenmiller and Schweizerhof, 1990]. This model is only valid for thin shell elements.

The Chang/Chang criteria is given as follows:
for the tensile fiber mode,

$$\sigma_{aa} > 0 \quad then \quad e_f^2 = \left(\frac{\sigma_{aa}}{X_t}\right)^2 + \beta\left(\frac{\sigma_{ab}}{S_c}\right) - 1 \begin{cases} \geq 0 \ failed \\ < 0 \ elastic \end{cases}, \tag{19.54.1}$$

$$E_a = E_b = G_{ab} = \nu_{ba} = \nu_{ab} = 0,$$

for the compressive fiber mode,

$$\sigma_{aa} < 0 \quad then \quad e_c^2 = \left(\frac{\sigma_{aa}}{X_c}\right)^2 - 1 \begin{cases} \geq 0 \ failed \\ < 0 \ elastic \end{cases}, \tag{19.54.2}$$

$$E_a = \nu_{ba} = \nu_{ab} = 0.$$

for the tensile matrix mode,

$$\sigma_{bb} > 0 \quad then \quad e_m^2 = \left(\frac{\sigma_{bb}}{Y_t}\right)^2 + \left(\frac{\sigma_{ab}}{S_c}\right)^2 - 1 \begin{cases} \geq 0 \ failed \\ < 0 \ elastic \end{cases}, \tag{19.54.3}$$

$$E_b = \nu_{ba} = 0. \quad \rightarrow G_{ab} = 0,$$

and for the compressive matrix mode,

$$\sigma_{bb} < 0 \quad then \quad e_d^2 = \left(\frac{\sigma_{bb}}{2S_c}\right)^2 + \left[\left(\frac{Y_c}{2S_c}\right)^2 - 1\right]\frac{\sigma_{bb}}{Y_c} + \left(\frac{\sigma_{ab}}{S_c}\right)^2 - 1 \begin{cases} \geq 0 \ failed \\ < 0 \ elastic \end{cases} \tag{19.54.4}$$

$$_b = \nu_{ba} = \nu_{ab} = 0. \quad \rightarrow G_{ab} = 0$$
$$X_c = 2Y_c \quad for\ 50\%\ fiber\ volume.$$

In the Tsay/Wu criteria the tensile and compressive fiber modes are treated as in the Chang/Chang criteria. The failure criterion for the tensile and compressive matrix mode is given as:

$$e_{md}^2 = \frac{\sigma_{bb}^2}{Y_c Y_t} + \left(\frac{\sigma_{ab}}{S_c}\right)^2 + \frac{(Y_c - Y_t)\sigma_{bb}}{Y_c Y_t} - 1 \begin{cases} \geq 0 \ failed \\ < 0 \ elastic \end{cases} \qquad (19.54.5)$$

For $\beta = 1$ we get the original criterion of Hashin [1980] in the tensile fiber mode.

For $\beta = 0$, we get the maximum stress criterion which is found to compare better to experiments.

Failure can occur in any of four different ways:

1. If DFAILT is zero, failure occurs if the Chang/Chang failure criterion is satisfied in the tensile fiber mode.
2. If DFAILT is greater than zero, failure occurs if the tensile fiber strain is greater than DFAILT or less than DFAILC.
3. If EFS is greater than zero, failure occurs if the effective strain is greater than EFS.
4. If TFAIL is greater than zero, failure occurs according to the element time step as described in the definition of TFAIL above.

When failure has occurred in all the composite layers (through-thickness integration points), the element is deleted. Elements which share nodes with the deleted element become "crashfront" elements and can have their strengths reduced by using the SOFT parameter with TFAIL greater than zero.

Information about the status in each layer (integration point) and element can be plotted using additional integration point variables. The number of additional integration point variables for shells written to the LS-DYNA database is input by the *DATABASE_BINARY definition as variable NEIPS. For Models 54 and 55 these additional variables are tabulated below (*i* = shell integration point):

History Variable	Description	Value	LS-PREPOST History Variable
1. *ef(i)*	*tensile fiber mode*	*1 – elastic* *0 – failed*	*1*
2. *ec(i)*	*compressive fiber mode*		*2*
3. *em(i)*	*tensile matrix mode*		*3*
4. *ed(i)*	*compressive matrix mode*		*4*
5. *efail*	*max[ef(ip)]*		*5*
6. *dam*	*damage parameter*	*-1 - element intact* *10^{-8} - element in crashfront* *+1 - element failed*	*6*

The following components, defined by the sum of failure indicators over all through-thickness integration points, are stored as element component 7 instead of the effective plastic strain.:

Description	Integration point
$\frac{1}{nip}\sum_{i=1}^{nip} ef(i)$	*1*
$\frac{1}{nip}\sum_{i=1}^{nip} ec(i)$	*2*
$\frac{1}{nip}\sum_{i=1}^{nip} cm(i)$	*3*

Material Model 57: Low Density Urethane Foam

The urethane foam model is available to model highly compressible foams such as those used in seat cushions and as padding on the Side Impact Dummy (SID). The compressive behavior is illustrated in Figure 19.57.1 where hysteresis on unloading is shown. This behavior under uniaxial loading is assumed not to significantly couple in the transverse directions. In tension the material behaves in a linear fashion until tearing occurs. Although our implementation may be somewhat unusual, it was first motivated by Shkolnikov [1991] and a paper by Storakers [1986]. The recent additions necessary to model hysteretic unloading and rate effects are due to Chang, et al., [1994]. These latter additions have greatly expanded the usefulness of this model.

The model uses tabulated input data for the loading curve where the nominal stresses are defined as a function of the elongations, ε_i, which are defined in terms of the principal stretches, λ_i, as:

$$\varepsilon_i = \lambda_i - 1 \tag{19.57.1}$$

The stretch ratios are found by solving for the eigenvalues of the left stretch tensor, V_{ij}, which is obtained via a polar decomposition of the deformation gradient matrix, F_{ij}:

$$F_{ij} = R_{ik}U_{kj} = V_{ik}R_{kj} \tag{19.57.2}$$

The update of V_{ij} follows the numerically stable approach of Taylor and Flanagan [1989]. After solving for the principal stretches, the elongations are computed and, if the elongations are compressive, the corresponding values of the nominal stresses, $\tau_{i.}$ are interpolated. If the elongations are tensile, the nominal stresses are given by

$$\tau_i = E\varepsilon_i \tag{19.57.3}$$

The Cauchy stresses in the principal system become

$$\sigma_i = \frac{\tau_i}{\lambda_j \lambda_k} \quad . \tag{19.57.4}$$

The stresses are then transformed back into the global system for the nodal force calculations.

When hysteretic unloading is used, the reloading will follow the unloading curve if the decay constant, β, is set to zero. If β is nonzero the decay to the original loading curve is governed by the expression:

$$1 - e^{-\beta t} \tag{19.57.5}$$

The bulk viscosity, which generates a rate dependent pressure, may cause an unexpected volumetric response and, consequently, it is optional with this model.

Rate effects are accounted for through linear viscoelasticity by a convolution integral of the form

$$\sigma_{ij}^r = \int_0^t g_{ijkl}(t-\tau) \frac{\partial \varepsilon_{kl}}{\partial \tau} d\tau \tag{19.57.6}$$

where $g_{ijkl}(t-\tau)$ is the relaxation function. The stress tensor, σ_{ij}^r, augments the stresses determined from the foam, σ_{ij}^f; consequently, the final stress, σ_{ij}, is taken as the summation of the two contributions:

$$\sigma_{ij} = \sigma_{ij}^f + \sigma_{ij}^r . \tag{19.57.7}$$

Since we wish to include only simple rate effects, the relaxation function is represented by one term from the Prony series:

$$g(t) = \alpha_0 + \sum_{m=1}^{N} \alpha_m e^{-\beta t} \tag{19.57.8}$$

given by,

$$g(t) = E_d e^{-\beta_1 t} \tag{19.57.9}$$

This model is effectively a Maxwell fluid which consists of a damper and spring in series. We characterize this in the input by a Young's modulus, E_d, and decay constant, β_1. The formulation is performed in the local system of principal stretches where only the principal values of stress are computed and triaxial coupling is avoided. Consequently, the one-dimensional nature of this foam material is unaffected by this addition of rate effects. The addition of rate effects necessitates twelve additional history variables per integration point. The cost and memory overhead of this model comes primarily from the need to “remember” the local system of principal stretches.

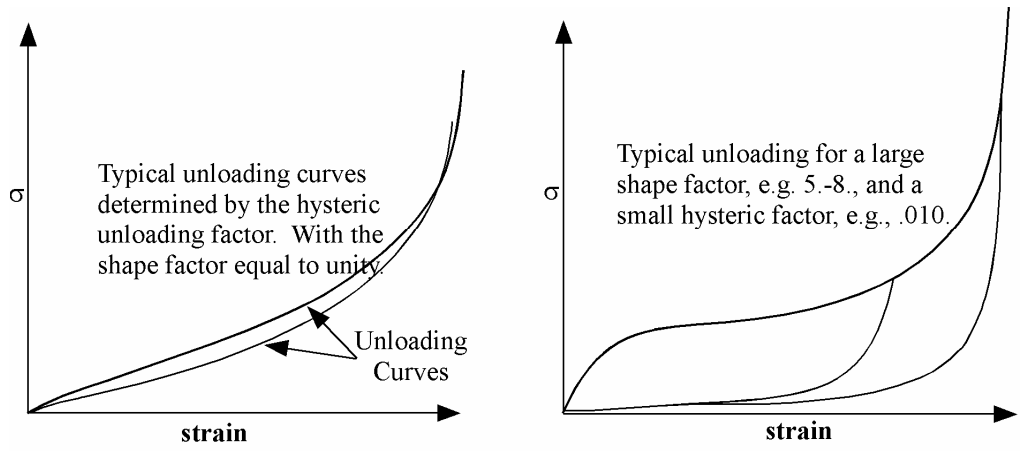


Figure 19.57.1. Behavior of the low-density urethane foam model.

Material Type 58: Laminated Composite Fabric

Parameters to control failure of an element layer are: ERODS, the maximum effective strain, i.e., maximum 1 = 100 % straining. The layer in the element is completely removed after the maximum effective strain (compression/tension including shear) is reached.

The stress limits are factors used to limit the stress in the softening part to a given value,

$$\sigma_{\min} = SLIMxx \cdot strength,$$

thus, the damage value is slightly modified such that elastoplastic like behavior is achieved with the threshold stress. As a factor for *SLIMxx* a number between 0.0 and 1.0 is possible. With a factor of 1.0, the stress remains at a maximum value identical to the strength, which is similar to ideal elastoplastic behavior. For tensile failure a small value for *SLIMTx* is often reasonable; however, for compression *SLIMCx = 1.0* is preferred. This is also valid for the corresponding shear value. If *SLIMxx* is smaller than 1.0 then localization can be observed depending on the total behavior of the lay-up. If the user is intentionally using *SLIMxx < 1.0,* it is generally recommended to avoid a drop to zero and set the value to something in between 0.05 and 0.10. Then elastoplastic behavior is achieved in the limit which often leads to less numerical problems. Defaults for SLIMXX = 1.0E-8.

The crashfront-algorithm is started if and only if a value for TSIZE (time step size, with element elimination after the actual time step becomes smaller than TSIZE) is input .

The damage parameters can be written to the postprocessing database for each integration point as the first three additional element variables and can be visualized.

Material models with FS=1 or FS=-1 are favorable for complete laminates and fabrics, as all directions are treated in a similar fashion.

For material model FS=1 an interaction between normal stresses and shear stresses is assumed for the evolution of damage in the a- and b- directions. For the shear damage is always the maximum value of the damage from the criterion in a- or b- direction is taken.

For material model FS=-1 it is assumed that the damage evolution is independent of any of the other stresses. A coupling is present only via the elastic material parameters and the complete structure.

In tensile and compression directions and in a- as well as in b- direction, different failure surfaces can be assumed. The damage values, however, increase only when the loading direction changes.

Special control of shear behavior of fabrics

For fabric materials a nonlinear stress strain curve for the shear part of failure surface FS=-1 can be assumed as given below. This is not possible for other values of FS.
The curve, shown in Figure 19.58.1, is defined by three points:

a) the origin (0,0) is assumed,
b) the limit of the first slightly nonlinear part (must be input), stress (TAU1) and strain (GAMMA1), see below.
c) the shear strength at failure and shear strain at failure.

In addition a stress limiter can be used to keep the stress constant via the *SLIMS* parameter. This value must be less than or equal to 1.0 and positive, which leads to an elastoplastic behavior for the shear part. The default is 1.0E-08, assuming almost brittle failure once the strength limit SC is reached.

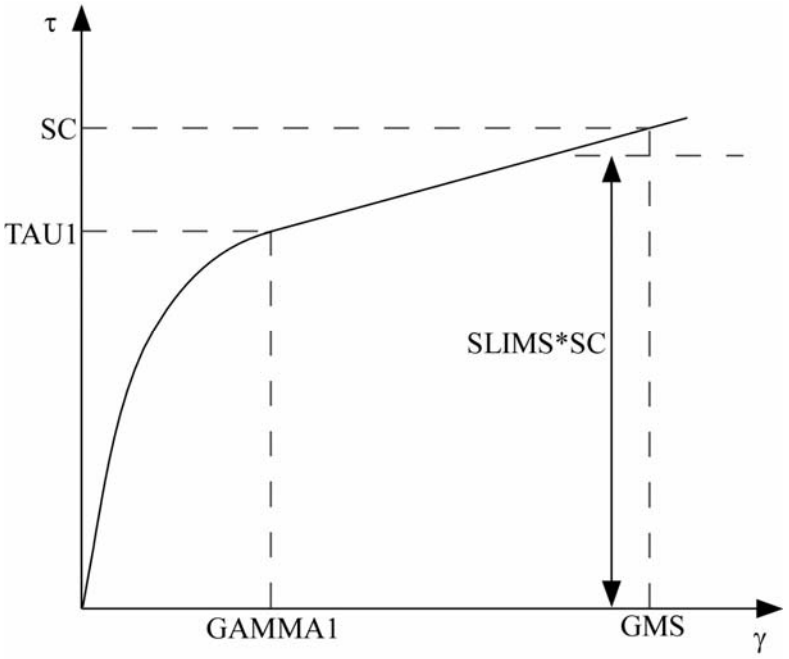


Figure 19.58.1. Stress-strain diagram for shear.

Material Type 60: Elastic With Viscosity

This material model was developed to simulate the forming of glass products (e.g., car windshields) at high temperatures. Deformation is by viscous flow but elastic deformations can also be large. The material model, in which the viscosity may vary with temperature, is suitable for treating a wide range of viscous flow problems and is implemented for brick and shell elements.

Volumetric behavior is treated as linear elastic. The deviatoric strain rate is considered to be the sum of elastic and viscous strain rates:

$$\dot{\underset{\sim}{\varepsilon}}'_{total} = \dot{\underset{\sim}{\varepsilon}}'_{elastic} + \dot{\underset{\sim}{\varepsilon}}'_{viscous} = \frac{\dot{\underset{\sim}{\sigma}}'}{2G} + \frac{\dot{\underset{\sim}{\sigma}}'}{2\nu} \tag{19.60.1}$$

where G is the elastic shear modulus, ν is the viscosity coefficient, and ~ indicates a tensor. The stress increment over one time step dt is

$$d\underset{\sim}{\sigma}' = 2G\dot{\underset{\sim}{\varepsilon}}'_{total}dt - \frac{G}{\upsilon}dt\underset{\sim}{\sigma}' \tag{19.60.2}$$

The stress before the update is used for $\underset{\sim}{\sigma}'$. For shell elements, the through-thickness strain rate is calculated as follows

$$d\sigma_{33} = 0 = K\left(\dot{\varepsilon}_{11} + \dot{\varepsilon}_{22} + \dot{\varepsilon}_{33}\right)dt + 2G\dot{\varepsilon}'_{33}\,dt - \frac{G}{\upsilon}dt\sigma'_{33} \tag{19.60.3}$$

where the subscript $ij=33$ denotes the through-thickness direction and K is the elastic bulk modulus. This leads to:

$$\dot{\varepsilon}_{33} = -a\left(\dot{\varepsilon}_{11} + \dot{\varepsilon}_{22}\right) + bp \tag{19.60.4}$$

$$a = \frac{\left(K - \frac{2}{3}G\right)}{\left(K + \frac{4}{3}G\right)} \tag{19.60.5}$$

$$b = \frac{Gdt}{\upsilon\left(K + \frac{4}{3}G\right)} \tag{19.60.6}$$

in which p is the pressure defined as the negative of the hydrostatic stress.

Material Model 61: Maxwell/Kelvin Viscoelastic with Maximum Strain

The shear relaxation behavior is described for the Maxwell model by:

$$G(t) = G_\infty + (G_0 - G_\infty)e^{-\beta t}. \tag{19.61.1}$$

A Jaumann rate formulation is used

$$\overset{\nabla}{s}'_{ij} = 2\int_0^t G\left(t-\tau\right)\dot{\varepsilon}'_{ij}\left(\tau\right)d\tau \tag{19.61.2}$$

where the prime denotes the deviatoric part of the stress rate, $\overset{\nabla}{s}'_{ij}$, and $\dot{\varepsilon}'_{ij}$ is the deviatoric strain rate.

For the Kelvin model the stress evolution equation is defined as:

$$\dot{s}_{ij} + \frac{1}{\tau} s_{ij} = \left(1+\delta_{ij}\right) G_0 \dot{\varepsilon}'_{ij} + \left(1+\delta_{ij}\right) \frac{G_\infty}{\tau} \varepsilon'_{ij} \qquad (19.61.3)$$

where δ_{ij} is the Kronecker delta, G_0 is the instantaneous shear modulus, G_∞ is the long term shear modulus, and τ is the decay constant.

The pressure is determined from the bulk modulus and the volumetric strain:

$$p = -K\varepsilon_v \qquad (19.61.4)$$

where

$$\varepsilon_v = \ln\left(\frac{V}{V_0}\right) \qquad (19.61.5)$$

defines the logarithmic volumetric strain from the relative volume.

Bandak's [1991] calculation of the total strain tensor, ε_{ij}, for output uses an incremental update based on Jaumann rate:

$$\varepsilon_{ij}^{n+1} = \varepsilon_{ij}^{n} + r_{ij}^{n} + \varepsilon_{ij}^{\nabla n+½} \Delta t^{n+½} \qquad (19.61.6)$$

where

$$\Delta\varepsilon_{ij}^{n+½} = \dot{\varepsilon}_{ij}^{n+½} \Delta t^{n+½} \qquad (19.61.7)$$

and r_{ij}^{n} gives the rotation of the stain tensor at time tt^{n} to the configuration at t^{n+1}

$$r_{ij}^{n} = \left(\varepsilon_{ip}^{n} \omega_{pj}^{n+½} + \varepsilon_{jp}^{n} \omega_{pi}^{n+½}\right) \Delta t^{n+½} \qquad (19.61.8)$$

Material Type 62: Viscous Foam

This model was written to represent the energy absorbing foam found on certain crash dummies, i.e., the 'Confor Foam' covering the ribs of the Eurosid dummy.

The model consists of a nonlinear elastic stiffness in parallel with a viscous damper. A schematic is shown in Figure 19.62.1. The elastic stiffness is intended to limit total crush while the viscous damper absorbs energy. The stiffness E_2 prevents timestep problems.

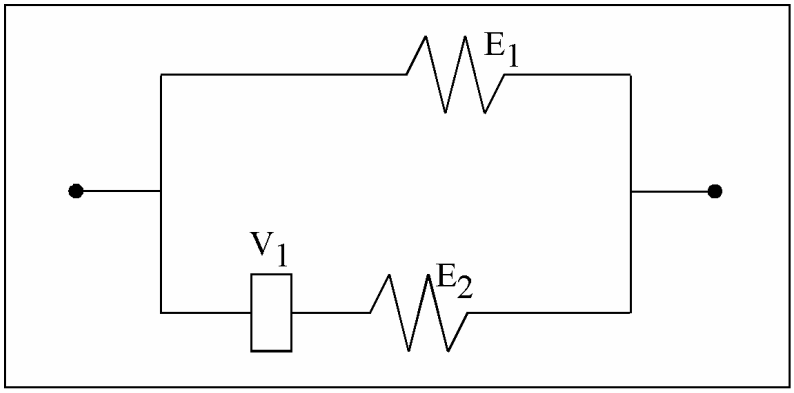


Figure 19.62.1. Schematic of Material Model 62.

Both E_1 and V_2 are nonlinear with crush as follows:

$$E_1^t = E_1\left(V^{-n_1}\right)$$
$$V_2^t = V_2\left(abs\left(1-V\right)\right)^{n_2}$$

where V is the relative volume defined by the ratio of the current to initial volume. Typical values are (units of N, mm, s)

$$E_1 = 0.0036$$
$$n_1 = 4.0$$
$$V_2 = 0.0015$$
$$E_2 = 100.0$$
$$n_2 = 0.2$$
$$\nu = 0.05$$

Material Type 63: Crushable Foam

The intent of this model is model crushable foams in side impact and other applications where cyclic behavior is unimportant.

This isotropic foam model crushes one-dimensionally with a Poisson's ratio that is essentially zero. The stress versus strain behavior is depicted in Figure 19.63.1 where an example of unloading from point **a** to the tension cutoff stress at **b** then unloading to point **c** and finally reloading to point **d** is shown. At point the reloading will continue along the loading curve. It is important to use nonzero values for the tension cutoff to prevent the disintegration of the material under small tensile loads. For high values of tension cutoff the behavior of the material will be similar in tension and compression.

In the implementation we assume that Young's modulus is constant and update the stress assuming elastic behavior.

$$\sigma_{ij}^{trial} = \sigma_{ij}^{n} + E\dot{\varepsilon}_{ij}^{n+\frac{1}{2}}\Delta t^{n+\frac{1}{2}} \qquad (19.63.1)$$

The magnitudes of the principal values, $\sigma_i^{trial}, i=1,3$ are then checked to see if the yield stress, σ_y, is exceeded and if so they are scaled back to the yield surface:

$$if\ \sigma_y < \left|\sigma_i^{trial}\right| \quad then \quad \sigma_i^{n+1} = \sigma_y \frac{\sigma_i^{trial}}{\left|\sigma_i^{trial}\right|} \tag{19.63.2}$$

After the principal values are scaled, the stress tensor is transformed back into the global system. As seen in Figure 19.63.1, the yield stress is a function of the natural logarithm of the relative volume, V, i.e., the volumetric strain.

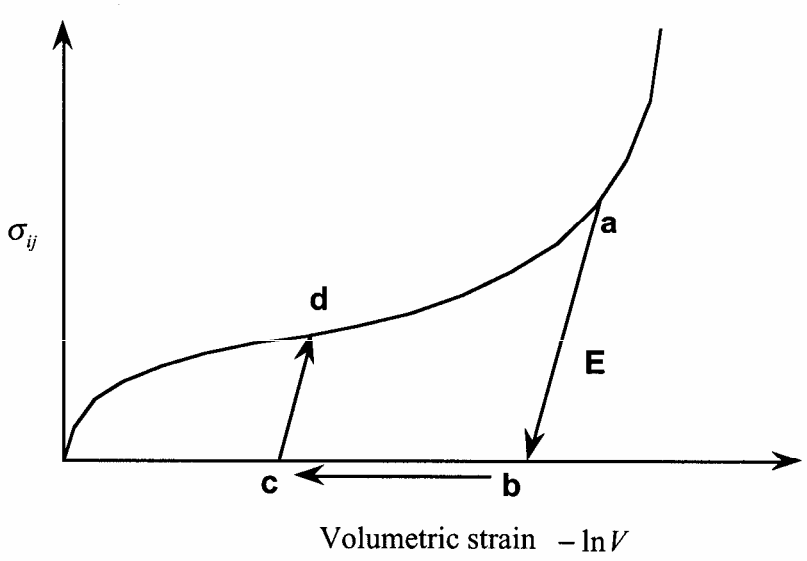


Figure 18.63.1. Yield stress versus volumetric strain curve for the crushable foam.

Material Model 64: Strain Rate Sensitive Power-Law Plasticity

This material model follows a constitutive relationship of the form:

$$\sigma = k\varepsilon^m \dot{\varepsilon}^n \tag{19.64.1}$$

where σ is the yield stress, ε is the effective plastic strain, $\dot{\varepsilon}$ is the effective plastic strain rate, and the constants $k, m,$ and n can be expressed as functions of effective plastic strain or can be constant with respect to the plastic strain. The case of no strain hardening can be obtained by setting the exponent of the plastic strain equal to a very small positive value, i.e., 0.0001.

This model can be combined with the superplastic forming input to control the magnitude of the pressure in the pressure boundary conditions in order to limit the effective plastic strain rate so that it does not exceed a maximum value at any integration point within the model.

A fully viscoplastic formulation is optional. An additional cost is incurred but the improvement in results can be dramatic.

Material Model 65: Modified Zerilli/Armstrong

The Armstrong-Zerilli Material Model expresses the flow stress as follows. For fcc metals,

$$\sigma = C_1 + \left\{ C_2 \left(\varepsilon^p\right)^{1/2} \left[e^{\left(-C_3 + C_4 \ln\left(\dot{\varepsilon}^*\right)\right)T} \right] + C_5 \right\} \left(\frac{\mu(T)}{\mu(293)} \right) \tag{19.65.1}$$

ε^p = effective plastic strain

$\dot{\varepsilon}^* = \dfrac{\dot{\varepsilon}}{\dot{\varepsilon}_0}$ effective plastic strain rate where $\dot{\varepsilon}_0$ = 1, 1e-3, 1e-6 for time units of seconds, milliseconds, and microseconds, respectively.

For bcc metals,

$$\sigma = C_1 + C_2 e^{\left(-C_3 + C_4 \ln\left(\dot{\varepsilon}^*\right)\right)T} + \left[C_5 \left(\varepsilon^p\right)^n + C_6 \right] \left(\frac{\mu(T)}{\mu(293)} \right) \tag{19.65.2}$$

where

$$\left(\frac{\mu(T)}{\mu(293)} \right) = B_1 + B_2 T + B_3 T^2 . \tag{19.65.3}$$

The relationship between heat capacity (specific heat) and temperature may be characterized by a cubic polynomial equation as follows:

$$C_p = G_1 + G_2 T + G_3 T^2 + G_4 T^3 \tag{19.65.4}$$

A fully viscoplastic formulation is optional. An additional cost is incurred but the improvement in results can be dramatic.

Material Model 66: Linear Stiffness/Linear Viscous 3D Discrete Beam

The formulation of the discrete beam (Type 6) assumes that the beam is of zero length and requires no orientation node. A small distance between the nodes joined by the beam is permitted. The local coordinate system which determines *(r,s,t)* is given by the coordinate ID in the cross sectional input where the global system is the default. The local coordinate system axes rotate with the average of the rotations of the two nodes that define the beam.

For null stiffness coefficients, no forces corresponding to these null values will develop. The viscous damping coefficients are optional.

Material Model 67: Nonlinear Stiffness/Viscous 3D Discrete Beam

The formulation of the discrete beam (Type 6) assumes that the beam is of zero length and requires no orientation node. A small distance between the nodes joined by the beam is permitted. The local coordinate system which determines *(r,s,t)* is given by the coordinate ID in the cross sectional input where the global system is the default. The local coordinate system axes rotate with the average of the rotations of the two nodes that define the beam.

For null load curve ID's, no forces are computed. The force resultants are found from load curves (See Figure 19.67.1) that are defined in terms of the force resultant versus the relative displacement in the local coordinate system for the discrete beam.

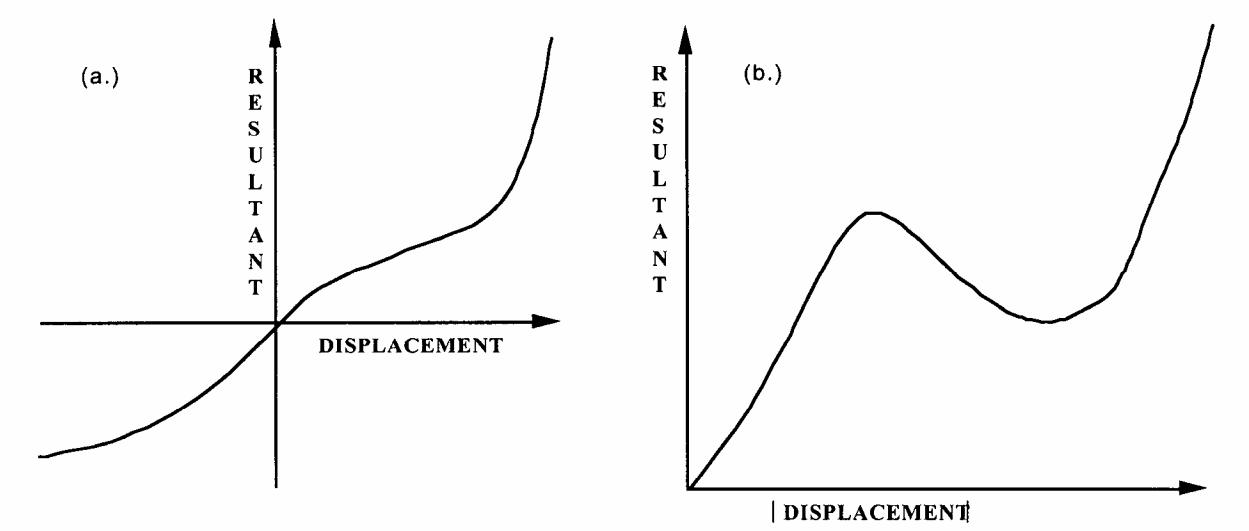


Figure 19.67.1. The resultant forces and moments are determined by a table lookup. If the origin of the load curve is at [0,0] as in (b.) and tension and compression responses are symmetric.

Material Model 68: Nonlinear Plastic/Linear Viscous 3D Discrete Beam

The formulation of the discrete beam (Type 6) assumes that the beam is of zero length and requires no orientation node. A small distance between the nodes joined by the beam is permitted. The local coordinate system which determines *(r,s,t)* is given by the coordinate ID in the cross sectional input where the global system is the default. The local coordinate system axes rotate with the average of the rotations of the two nodes that define the beam. Each force resultant in the local system can have a limiting value defined as a function of plastic displacement by using a load curve (See Figure 19.68.1). For the degrees of freedom where elastic behavior is desired, the load curve ID is simply set to zero.

Catastrophic failure, based on force resultants, occurs if the following inequality is satisfied:

$$\left(\frac{F_r}{F_r^{fail}}\right)^2+\left(\frac{F_s}{F_s^{fail}}\right)^2+\left(\frac{F_t}{F_t^{fail}}\right)^2+\left(\frac{M_r}{M_r^{fail}}\right)^2+\left(\frac{M_s}{M_s^{fail}}\right)^2+\left(\frac{M_t}{M_t^{fail}}\right)^2-1.\geq 0. \quad (19.68.1)$$

Likewise, catastrophic failure based on displacement resultants occurs if:

$$\left(\frac{u_r}{u_r^{fail}}\right)^2+\left(\frac{u_s}{u_s^{fail}}\right)^2+\left(\frac{u_t}{u_t^{fail}}\right)^2+\left(\frac{\theta_r}{\theta_r^{fail}}\right)^2+\left(\frac{\theta_s}{\theta_s^{fail}}\right)^2+\left(\frac{\theta_t}{\theta_t^{fail}}\right)^2-1.\geq 0. \qquad (19.68.2)$$

After failure, the discrete element is deleted. If failure is included, either one or both of the criteria may be used.

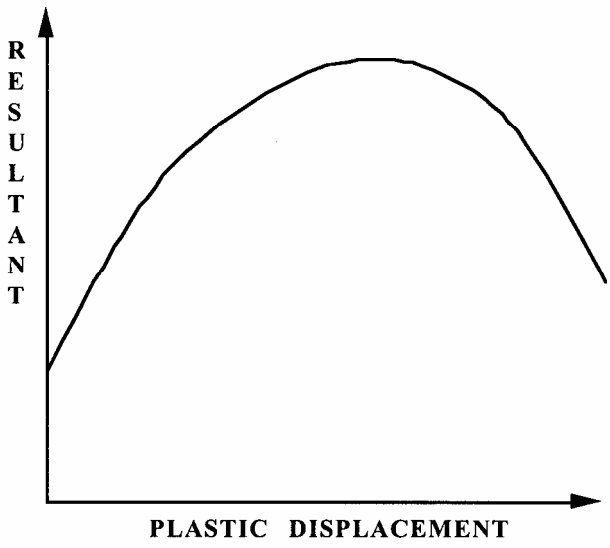


Figure 19.68.1. The resultant forces and moments are limited by the yield definition. The initial yield point corresponds to a plastic displacement of zero.

Material Model 69: Side Impact Dummy Damper (SID Damper)

The side impact dummy uses a damper that is not adequately treated by nonlinear force versus relative velocity curves, since the force characteristics are also dependent on the displacement of the piston. As the damper moves, the fluid flows through the open orifices to provide the necessary damping resistance. While moving as shown in Figure 19.69.1, the piston gradually blocks off and effectively closes the orifices. The number of orifices and the size of their openings control the damper resistance and performance. The damping force is computed from the equation:

$$F=SF\left\{KA_pV_p\left\{\frac{C_1}{A_0^t}+C_2\left|V_p\right|\rho_{fluid}\left[\left(\frac{A_p}{CA_0^t}\right)^2-1\right]\right\}-f\left(s+s_0\right)+V_pg(s+s_0)\right\} \qquad (19.69.1)$$

where K is a user defined constant or a tabulated function of the absolute value of the relative velocity, V_p is the piston's relative velocity, C is the discharge coefficient, A_p is the piston area,

A_0^t is the total open areas of orifices at time t, ρ_{fluid} is the fluid density, C_1 is the coefficient for the linear term, and C_2 is the coefficient for the quadratic term.

In the implementation, the orifices are assumed to be circular with partial covering by the orifice controller. As the piston closes, the closure of the orifice is gradual. This gradual closure is taken into account to insure a smooth response. If the piston stroke is exceeded, the stiffness value, k, limits further movement, i.e., if the damper bottoms out in tension or compression, the damper forces are calculated by replacing the damper by a bottoming out spring and damper, k and c, respectively. The piston stroke must exceed the initial length of the beam element. The time step calculation is based in part on the stiffness value of the bottoming out spring. A typical force versus displacement curve at constant relative velocity is shown in Figure 19.15. The factor, SF, which scales the force defaults to 1.0 and is analogous to the adjusting ring on the damper.

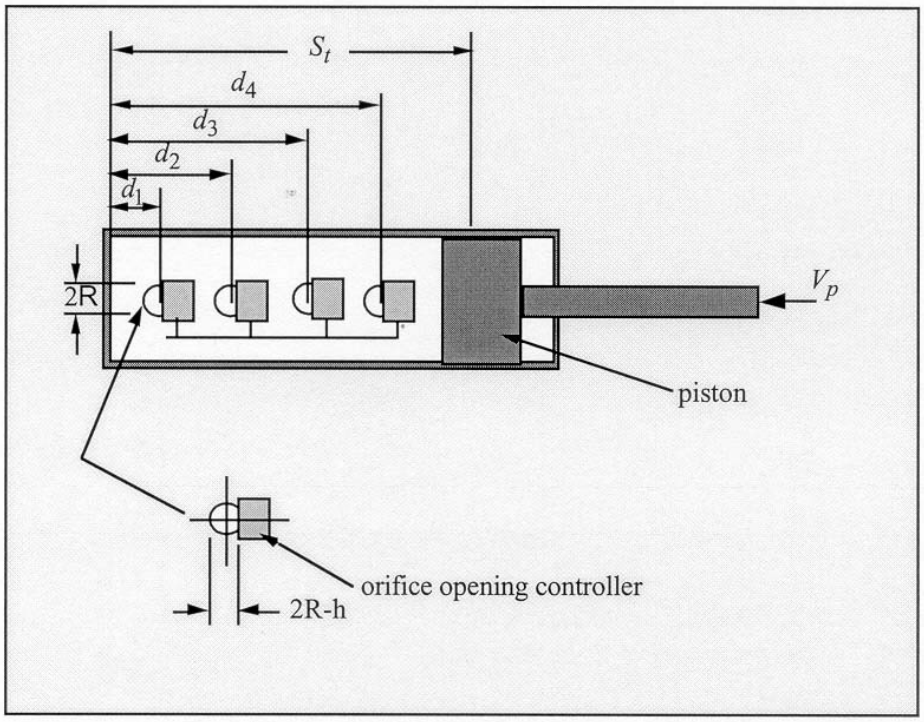


Figure 19.69.1. Mathematical model for the Side Impact Dummy damper.

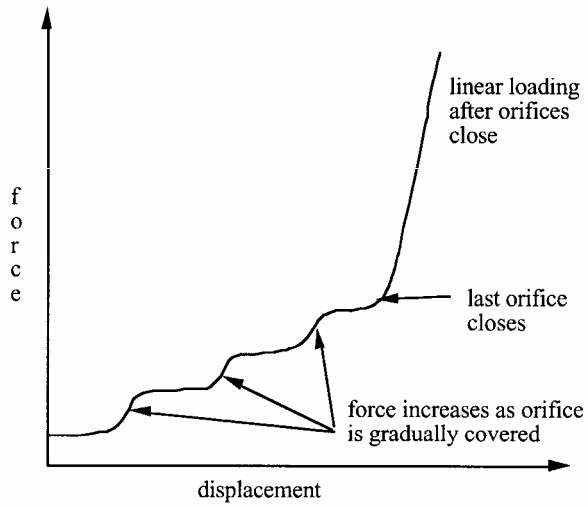


Figure 19.69.2. Force versus displacement as orifices are covered at a constant relative velocity. Only the linear velocity term is active.

Material Model 70: Hydraulic/Gas Damper

This special purpose element represents a combined hydraulic and gas-filled damper which has a variable orifice coefficient. A schematic of the damper is shown in Figure 19.70.1. Dampers of this type are sometimes used on buffers at the end of railroad tracks and as aircraft undercarriage shock absorbers. This material can be used only as a discrete beam element.

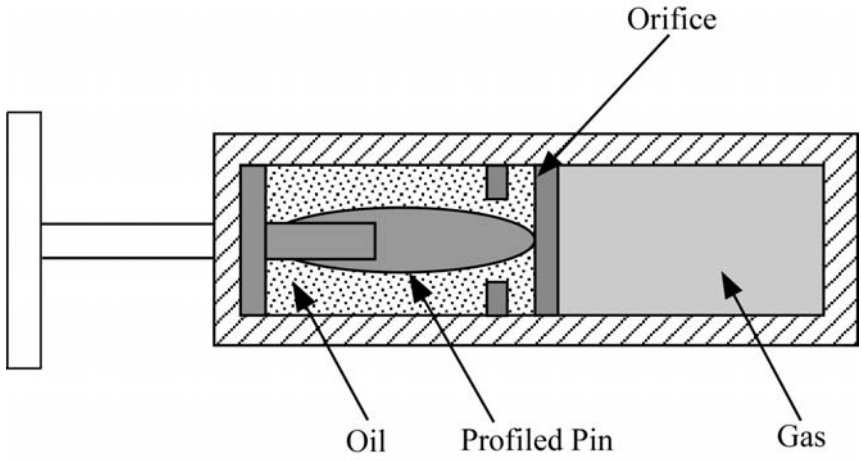


Figure 19.70.1. Schematic of Hydraulic/Gas damper.

As the damper is compressed two actions contribute to the force that develops. First, the gas is adiabatically compressed into a smaller volume. Secondly, oil is forced through an orifice. A profiled pin may occupy some of the cross-sectional area of the orifice; thus, the orifice area available for the oil varies with the stroke. The force is assumed proportional to the square of the velocity and inversely proportional to the available area. The equation for this element is:

$$F = SCLF \cdot \left\{ K_h \left(\frac{V}{a_0} \right)^2 + \left[P_0 \left(\frac{C_0}{C_0 - S} \right)^n - P_a \right] \cdot A_p \right\} \tag{19.70.1}$$

where S is the element deflection and V is the relative velocity across the element.

Material Model 71: Cable

This material can be used only as a discrete beam element. The force, F, generated by the cable is nonzero only if the cable is in tension. The force is given by:

$$F = K \cdot \max(\Delta L, 0.) \tag{19.71.1}$$

where ΔL is the change in length

$$\Delta L = current\ length - (initial\ length - offset) \tag{19.71.2}$$

and the stiffness is defined as:

$$K = \frac{E \cdot area}{(initial\ length - offset)} \tag{19.71.3}$$

The area and offset are defined on either the cross section or element cards in the LS-DYNA input. For a slack cable the offset should be input as a negative length. For an initial tensile force the offset should be positive. If a load curve is specified, the Young's modulus will be ignored and the load curve will be used instead. The points on the load curve are defined as engineering stress versus engineering strain, i.e., the change in length over the initial length. The unloading behavior follows the loading.

Material Model 73: Low Density Viscoelastic Foam

This viscoelastic foam model is available to model highly compressible viscous foams. The hyperelastic formulation of this model follows that of material 57.

Rate effects are accounted for through linear viscoelasticity by a convolution integral of the form

$$\sigma_{ij}^r = \int_0^t g_{ijkl}(t-\tau) \frac{\partial \varepsilon_{kl}}{\partial \tau} d\tau \tag{19.73.1}$$

where $g_{ijkl}(t-\tau)$ is the relaxation function. The stress tensor, σ_{ij}^r, augments the stresses determined from the foam, σ_{ij}^f; consequently, the final stress, σ_{ij}, is taken as the summation of the two contributions:

$$\sigma_{ij} = \sigma_{ij}^f + \sigma_{ij}^r . \tag{19.73.2}$$

Since we wish to include only simple rate effects, the relaxation function is represented by up to six terms of the Prony series:

$$g(t) = \alpha_0 + \sum_{m=1}^{N} \alpha_m e^{-\beta t} \qquad (19.73.3)$$

This model is effectively a Maxwell fluid which consists of a dampers and springs in series. The formulation is performed in the local system of principal stretches where only the principal values of stress are computed and triaxial coupling is avoided. Consequently, the one-dimensional nature of this foam material is unaffected by this addition of rate effects. The addition of rate effects necessitates 42 additional history variables per integration point. The cost and memory overhead of this model comes primarily from the need to "remember" the local system of principal stretches and the evaluation of the viscous stress components.

Material Model 74: Elastic Spring for the Discrete Beam

This model permits elastic springs with damping to be combined and represented with a discrete beam element type 6. Linear stiffness and damping coefficients can be defined, and, for nonlinear behavior, a force versus deflection and force versus rate curves can be used. Displacement based failure and an initial force are optional

If the linear spring stiffness is used, the force, *F*, is given by:

$$F = F_0 + K\Delta L + D\Delta\dot{L} \qquad (19.74.1)$$

where *K* is the stiffness constant, and D is the viscous damping coefficient.

If the load curve ID for $f(\Delta L)$ is specified, nonlinear behavior is activated. For this case the force is given by:

$$F = F_0 + K\, f(\Delta L)\left[1 + C1\cdot\Delta\dot{L} + C2\cdot\operatorname{sgn}(\Delta\dot{L})\ln\left(\max\left\{1., \frac{|\Delta\dot{L}|}{DLE}\right\}\right)\right] + D\Delta\dot{L} + g(\Delta L)h(\Delta\dot{L}) \qquad (19.74.2)$$

where *C1* and *C2* are damping coefficients for nonlinear behavior, *DLE* is a factor to scale time units, and $g(\Delta L)$ is an optional load curve defining a scale factor versus deflection for load curve ID, $h(d\Delta L/dt)$.

In these equations, ΔL is the change in length

$$\Delta L = currentlength - initiallength$$

Failure can occur in either compression or tension based on displacement values of CDF and TDF, respectively. After failure no forces are carried. Compressive failure does not apply if the spring is initially zero length.

The cross sectional area is defined on the section card for the discrete beam elements, in *SECTION_BEAM. The square root of this area is used as the contact thickness offset if these elements are included in the contact treatment.

Material Model 75: Bilkhu/Dubois Foam Model

This model uses uniaxial and triaxial test data to provide a more realistic treatment of crushable foam. The Poisson's ratio is set to zero for the elastic response. The volumetric strain is defined in terms of the relative volume, V , as:

$$\gamma = -\ln(V) \tag{19.75.1}$$

In defining the curves, the stress and strain pairs should be positive values starting with a volumetric strain value of zero.

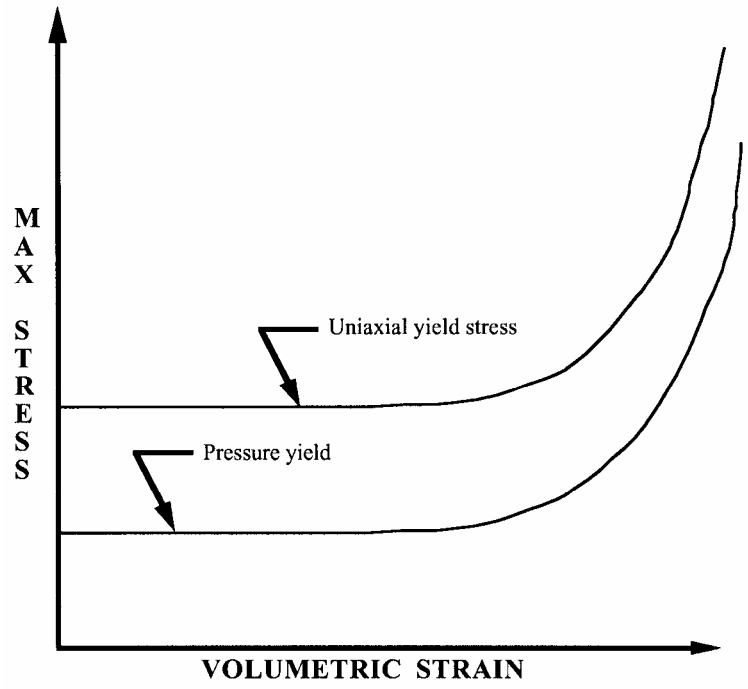


Figure 19.75.1. Behavior of crushable foam. Unloading is elastic.

Material Model 76: General Viscoelastic

Rate effects are taken into account through linear viscoelasticity by a convolution integral of the form:

$$\sigma_{ij} = \int_0^t g_{ijkl}(t-\tau)\frac{\partial \varepsilon_{kl}}{\partial \tau} d\tau \qquad (19.76.1)$$

where $g_{ijkl}(t-\tau)$ is the relaxation function.

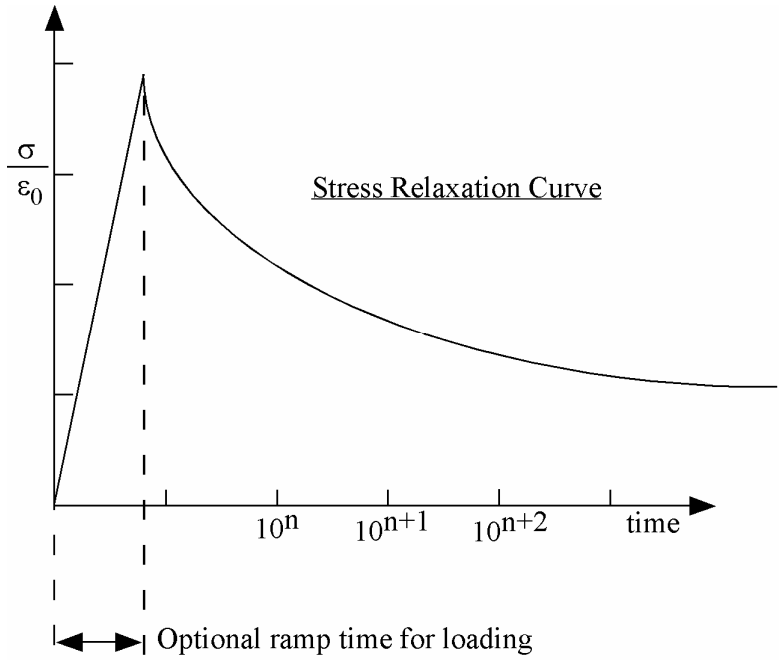


Figure 19.76.1. Relaxation curve. This curve defines stress versus time where time is defined on a logarithmic scale. For best results, the points defined in the load curve should be equally spaced on the logarithmic scale. Furthermore, the load curve should be smooth and defined in the positive quadrant. If nonphysical values are determined by least squares fit, LS-DYNA will terminate with an error message after the initialization phase is completed. If the ramp time for loading is included, then the relaxation which occurs during the loading phase is taken into account. This effect may or may not be important.

If we wish to include only simple rate effects for the deviatoric stresses, the relaxation function is represented by six terms from the Prony series:

$$g(t) = \sum_{m=1}^{N} G_m e^{-\beta_m t} \qquad (19.76.2)$$

We characterize this in the input by shear modulii, G_i, and decay constants, β_i. An arbitrary number of terms, up to 6, may be used when applying the viscoelastic model.

For volumetric relaxation, the relaxation function is also represented by the Prony series in terms of bulk modulii:

$$k(t) = \sum_{m=1}^{N} K_m e^{-\beta_{k_m} t} \tag{19.76.3}$$

Material Model 77: Hyperviscoelastic Rubber

Material type 77 in LS-DYNA consists of two hyperelastic rubber models, a general hyperelastic rubber model and an Ogden rubber model, that can be combined optionally with a viscoelastic stress contribution. As for the rate independent part, the constitutive law is determined by a strain energy function which in this case advantageously can be expressed in terms of the principal stretches, i.e., $W = W(\lambda_1, \lambda_2, \lambda_3)$. To obtain the Cauchy stress σ_{ij}, as well as the constitutive tensor of interest, D_{ijkl}^{TC}, they are first calculated in the principal basis after which they are transformed back to the "base frame", or standard basis. The complete set of formulas is given by Crisfield [1997] and is for the sake of completeness recapitulated here.

The principal Kirchoff stress components are given by

$$\tau_{ii}^{E} = \lambda_i \frac{\partial W}{\partial \lambda_i} \text{ (no sum)}$$

that are transformed to the standard basis using the standard formula

$$\tau_{ij} = q_{ik} q_{jl} \tau_{kl}^{E}.$$

The q_{ij} are the components of the orthogonal tensor containing the eigenvectors of the principal basis. The Cauchy stress is then given by

$$\sigma_{ij} = J^{-1} \tau_{ij},$$

where $J = \lambda_1 \lambda_2 \lambda_3$ is the relative volume change.

The constitutive tensor that relates the rate of deformation to the Truesdell (convected) rate of Kirchoff stress can in the principal basis be expressed as

$$D_{iijj}^{TKE} = \lambda_j \frac{\partial \tau_{ii}^{E}}{\partial \lambda_j} - 2\tau_{ii}^{E} \delta_{ij}$$

$$D_{ijij}^{TKE} = \frac{\lambda_j^2 \tau_{ii}^{E} - \lambda_i^2 \tau_{jj}^{E}}{\lambda_i^2 - \lambda_j^2}, \quad i \neq j, \lambda_i \neq \lambda_j \quad \text{(no sum)}$$

$$D_{ijij}^{TKE} = \frac{\lambda_i}{2}\left(\frac{\partial \tau_{ii}^{E}}{\partial \lambda_i} - \frac{\partial \tau_{ii}^{E}}{\partial \lambda_j}\right), \quad i \neq j, \lambda_i = \lambda_j$$

These components are transformed to the standard basis according to

$$D_{ijkl}^{TK} = q_{ip} q_{jq} q_{kr} q_{ls} D_{pqrs}^{TKE},$$

and finally the constitutive tensor relating the rate of deformation to the Truesdell rate of Cauchy stress is obtained through

$$D_{ijkl}^{TC} = J^{-1} D_{ijkl}^{TK}.$$

When dealing with shell elements, the tangent moduli in the corotational coordinates is of interest. This matrix is given by

$$\hat{D}_{ijkl}^{TC} = R_{pi} R_{qj} R_{rk} R_{sl} D_{pqrs}^{TC} = J^{-1} R_{pi} R_{qj} R_{rk} R_{sl} D_{pqrs}^{TK} = J^{-1} \hat{q}_{ip} \hat{q}_{jq} \hat{q}_{kr} \hat{q}_{ls} D_{pqrs}^{TKE} \qquad (19.77.1)$$

where R_{ij} is the matrix containing the unit basis vectors of the corotational system and $\hat{q}_{ij} = R_{ki} q_{kj}$. The latter matrix can be determined as the eigenvectors of the co-rotated left Cauchy-Green tensor (or the left stretch tensor). In LS-DYNA, the tangent stiffness matrix is after assembly transformed back to the standard basis according to standard transformation formulae.

19.77.1 General Hyperelastic Rubber Model

The strain energy function for the general hyperelastic rubber model is given by

$$W = \sum_{p,q=0}^{n} C_{pq} W_1^{\,p} W_2^{\,q} + \frac{1}{2} K (J-1)^2$$

where K is the bulk modulus,

$$W_1 = I_1 I_3^{-1/3} - 3$$
$$W_2 = I_2 I_3^{-2/3} - 3$$

and

$$I_1 = \lambda_1^2 + \lambda_2^2 + \lambda_3^2$$
$$I_2 = \lambda_1^2 \lambda_2^2 + \lambda_2^2 \lambda_3^2 + \lambda_1^2 \lambda_3^2$$
$$I_3 = \lambda_1^2 \lambda_2^2 \lambda_3^2$$

are the invariants in terms of the principal stretches. To apply the formulas in the previous section, we require

$$\tau_{ii}^E = \lambda_i \frac{\partial W}{\partial \lambda_i} = \sum_{p,q=0}^{n} C_{pq}(pW_1^{p-1}W_{1i}^{'}W_2^q + qW_1^pW_2^{q-1}W_{2i}^{'}) + KJ(J-1) \qquad (19.77.2)$$

where

$$W_{1i}^{'} := \lambda_i \frac{\partial W_1}{\partial \lambda_i} = (2\lambda_i^2 - \frac{2}{3}I_1)I_3^{-1/3}$$

$$W_{2i}^{'} := \lambda_i \frac{\partial W_2}{\partial \lambda_i} = (2\lambda_i^2(I_1 - \lambda_i^2) - \frac{4}{3}I_2)I_3^{-2/3}$$

If C_{pq} is nonzero only for $pq = 01,10,11,20,02,30$, then Equation (19.77.2) can be written as

$$\tau_{ii}^E = (C_{10} + C_{11}W_2 + 2C_{20}W_1 + 3C_{30}W_1^2)W_{1i}^{'} + (C_{01} + C_{11}W_1 + 2C_{02}W_2)W_{2i}^{'} + KJ(J-1)$$

Proceeding with the constitutive tensor, we have

$$\begin{aligned}\lambda_j \frac{\partial \tau_{ii}^E}{\partial \lambda_j} = \sum_{p,q=0}^{n} C_{pq}\big(&p(p-1)W_1^{p-2}W_{1i}^{'}W_{1j}^{'}W_2^q + pW_1^{p-1}W_{1ij}^{''}W_2^q + pqW_1^{p-1}W_{1i}^{'}W_2^{q-1}W_{2j}^{'} \\ &+ qpW_1^{p-1}W_{1j}^{'}W_2^{q-1}W_{2i}^{'} + q(q-1)W_1^pW_2^{q-2}W_{2i}^{'}W_{2j}^{'} + qW_1^pW_2^{q-1}W_{2ij}^{''}\big) \\ &+ KJ(2J-1)\end{aligned} \qquad (19.77.3)$$

where

$$W_{1ij}^{''} := \lambda_j \frac{\partial W_{1i}^{'}}{\partial \lambda_j} = (4\lambda_i^2\delta_{ij} - \frac{4}{3}(\lambda_i^2 + \lambda_j^2) + \frac{4}{9}I_1)I_3^{-1/3}$$

$$W_{2ij}^{''} := \lambda_j \frac{\partial W_{2i}^{'}}{\partial \lambda_j} = ((4\lambda_i^2 I_1 - 8\lambda_i^4)\delta_{ij} + 4\lambda_i^2\lambda_j^2 - \frac{8}{3}(\lambda_i^2(I_1 - \lambda_i^2) + \lambda_j^2(I_1 - \lambda_j^2)) + \frac{16}{9}I_2)I_3^{-2/3}$$

Again, using only the nonzero coefficients mentioned above, Equation (19.77.3) is reduced to

$$\begin{aligned}\lambda_j \frac{\partial \tau_{ii}^E}{\partial \lambda_j} = &C_{11}(W_{1i}^{'}W_{2j}^{'} + W_{1j}^{'}W_{2i}^{'}) + 2(C_{20} + 3C_{30}W_1)W_{1j}^{'}W_{1i}^{'} + 2C_{02}W_{2i}^{'}W_{2j}^{'} + \\ &(C_{10} + C_{11}W_2 + 2C_{20}W_1 + 3C_{30}W_1^2)W_{1ij}^{''} + (C_{01} + C_{11}W_1 + 2C_{02}W_2)W_{2ij}^{''} + \\ &KJ(2J-1)\end{aligned}$$

19.77.2 Ogden Rubber Model

The strain energy function for the Ogden rubber model is given by

$$W = \sum_{m=1}^{n} \frac{\mu_m}{\alpha_m}(\tilde{\lambda}_1^{\alpha_m} + \tilde{\lambda}_2^{\alpha_m} + \tilde{\lambda}_3^{\alpha_m} - 3) + \frac{1}{2}K(J-1)^2$$

where

$$\tilde{\lambda}_i = \frac{\lambda_i}{J^{1/3}}$$

are the volumetric independent principal stretches, and μ_m and α_m are material parameters. To apply the formulas in the previous section, we require

$$\tau_{ii}^E = \lambda_i \frac{\partial W}{\partial \lambda_i} = \sum_{m=1}^{n} \mu_m (\tilde{\lambda}_i^{\alpha_m} - \frac{1}{3} a_m) + KJ(J-1)$$

where

$$a_m = \tilde{\lambda}_1^{\alpha_m} + \tilde{\lambda}_2^{\alpha_m} + \tilde{\lambda}_3^{\alpha_m}$$

Proceeding with the constitutive tensor, we have

$$\lambda_j \frac{\partial \tau_{ii}^E}{\partial \lambda_j} = \sum_{m=1}^{n} \frac{\mu_m \alpha_m}{3} (\frac{1}{3} a_m + 3\tilde{\lambda}_i^{\alpha_m} \delta_{ij} - \tilde{\lambda}_i^{\alpha_m} - \tilde{\lambda}_j^{\alpha_m}) + KJ(2J-1).$$

19.77.3 The Viscoelastic Contribution

As mentioned above, this material model is accompanied with a viscoelastic stress contribution. The rate form of this constitutive law can in co-rotational coordinates be written

$$\dot{\hat{\sigma}}_{ij}^{ve} = \sum_{m=1}^{n} 2G_m \hat{D}_{ij}^{\text{dev}} - \sum_{m=1}^{n} 2\beta_m G_m \int_0^t e^{-\beta_m (t-\tau)} \hat{D}_{ij}^{\text{dev}}(\tau) d\tau .$$

Here n is a number less than or equal to 6, $\hat{\sigma}_{ij}^{ve}$ is the co-rotated viscoelastic stress, $\hat{D}_{ij}^{\text{dev}}$ is the deviatoric co-rotated rate-of-deformation and G_m and β_m are material parameters. The parameters G_m can be thought of as shear moduli and β_m as decay coefficients determining the relaxation properties of the material. This rate form can be integrated in time to form the corotated viscoelastic stress

$$\hat{\sigma}_{ij}^{ve} = \sum_{m=1}^{6} 2G_m \int_0^t e^{-\beta_m (t-\tau)} \hat{D}_{ij}^{\text{dev}}(\tau) d\tau . \qquad (19.77.4)$$

For the constitutive matrix, we refer to Borrvall [2002] and here simply state that it is equal to

$$\hat{D}_{ijkl}^{TCve} = \sum_{m=1}^{n} 2G_m (\frac{1}{2}(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) - \frac{1}{3}\delta_{ij}\delta_{kl}) .$$

19.77.4 Stress Update for Shell Elements

In principal, the stress update for material 77 and shell elements follows closely the one that is implemented for material 27. The stress is evaluated in corotational coordinates after which it is transformed back to the standard basis according to

$$\sigma_{ij} = R_{ik} R_{jl} \hat{\sigma}_{kl} ,$$

or equivalently the internal force is assembled in the corotational system and then transformed back to the standard basis according to standard transformation formulae. Here R_{ij} is the rotation matrix containing the corotational basis vectors. The so-called corotated stress $\hat{\sigma}_{ij}$ is evaluated as the sum of the stresses given in Sections 19.77.1 and 19.77.4.

The viscoelastic stress contribution is incrementally updated with aid of the corotated rate of deformation. To be somewhat more precise, the values of the 12 integrals in Equation (19.77.4) are kept as history variables that are updated in each time step. Each integral is discretized in time and the mean value theorem is used in each time step to determine their values.

For the hyperelastic stress contribution, the principal stretches are needed and here taken as the square root of the eigenvalues of the co-rotated left Cauchy-Green tensor $\hat{b}_{ij}$. The corotated left Cauchy-Green tensor is incrementally updated with the aid of a time increment Δt, the corotated velocity gradient $\hat{L}_{ij}$, and the angular velocity $\hat{\Omega}_{ij}$ with which the embedded coordinate system is rotating,

$$\hat{b}_{ij} = \hat{b}_{ij} + \Delta t(\hat{L}_{ik} - \hat{\Omega}_{ik})\hat{b}_{kj} + \Delta t\hat{b}_{ik}(\hat{L}_{ik} - \hat{\Omega}_{ik}) . \qquad (19.77.5)$$

The primary reason for taking a corotational approach is to facilitate the maintenance of a vanishing normal stress through the thickness of the shell, something that is achieved by adjusting the corresponding component of the corotated velocity gradient $\hat{L}_{33}$ accordingly. The problem can be stated as to determine $\hat{L}_{33}$ such that when updating the left Cauchy-Green tensor through Equation (19.77.5) and subsequently the stress through formulae in Sections 19.77.1 and 19.77.4, $\hat{\sigma}_{33} = 0$. To this end, it is assumed that

$$\hat{L}_{33} = \alpha(\hat{L}_{11} + \hat{L}_{22}) ,$$

for some parameter α that is determined in the following three step procedure. In the first two steps, $\alpha = 0$ and $\alpha = -1$, respectively, resulting in two trial normal stresses $\hat{\sigma}_{33}^{(0)}$ and $\hat{\sigma}_{33}^{(-1)}$. Then it is assumed that the actual normal stress depends linearly on α, meaning that the latter can be determined from

$$0 = \sigma_{33}^{(\alpha)} = \sigma_{33}^{(0)} + \alpha(\sigma_{33}^{(0)} - \sigma_{33}^{(-1)}) .$$

In the current implementation, α is given by

$$\alpha = \begin{cases} \dfrac{\hat{\sigma}_{33}^{(0)}}{\hat{\sigma}_{33}^{(-1)} - \hat{\sigma}_{33}^{(0)}}, & \text{if } \left|\hat{\sigma}_{33}^{(-1)} - \hat{\sigma}_{33}^{(0)}\right| \geq 10^{-4} \\ -1, & \text{otherwise} \end{cases},$$

and the stresses are determined from this value of α. Finally, to make sure that the normal stress through the thickness vanishes, it is set to 0 (zero) before exiting the stress update routine.

Material Model 78: Soil/Concrete

Concrete pressure is positive in compression. Volumetric strain is defined as the natural log of the relative volume and is *positive* in compression where the relative volume, V, is the ratio of the current volume to the initial volume. The tabulated data should be given in order of increasing compression. If the pressure drops below the cutoff value specified, it is reset to that value and the deviatoric stress state is eliminated.

If the load curve ID is provided as a positive number, the deviatoric perfectly plastic pressure dependent yield function ϕ, is described in terms of the second invariant, J_2, the pressure, p, and the tabulated load curve, $F(p)$, as

$$\phi = \sqrt{3J_2} - F(p) = \sigma_y - F(p)$$

where J_2 is defined in terms of the deviatoric stress tensor as:

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

assuming that if the ID is given as negative, then the yield function becomes:

$$\phi = J_2 - F(p)$$

being the deviatoric stress tensor.

If cracking is invoked, the yield stress is multiplied by a factor f which reduces with plastic stain according to a trilinear law as shown in Figure 19.78.1.

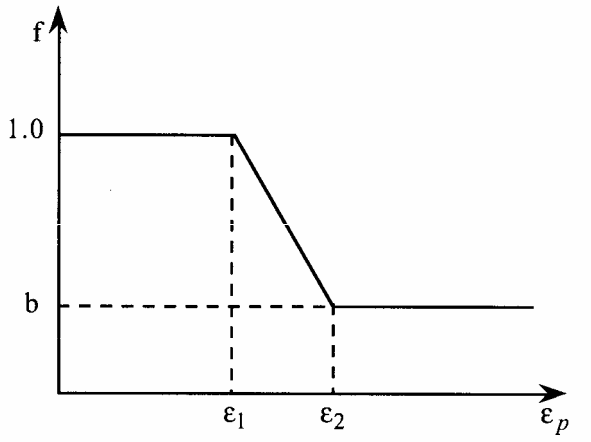


Figure 19.78.1. Strength reduction factor.

b	=	residual strength factor
ε_1	=	plastic stain at which cracking begins.
ε_2	=	plastic stain at which residual strength is reached.

ε_1 and ε_2 are tabulated functions of pressure that are defined by load curves (see Figure 19.78.2). The values on the curves are pressure versus strain and should be entered in order of increasing pressure. The strain values should always increase monotonically with pressure.

By properly defining the load curves, it is possible to obtain the desired strength and ductility over a range of pressures. See Figure 19.78.1.

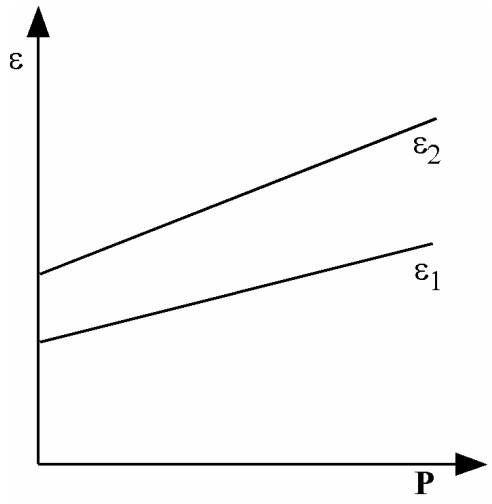


Figure 19.78.2. Cracking strain versus pressure.

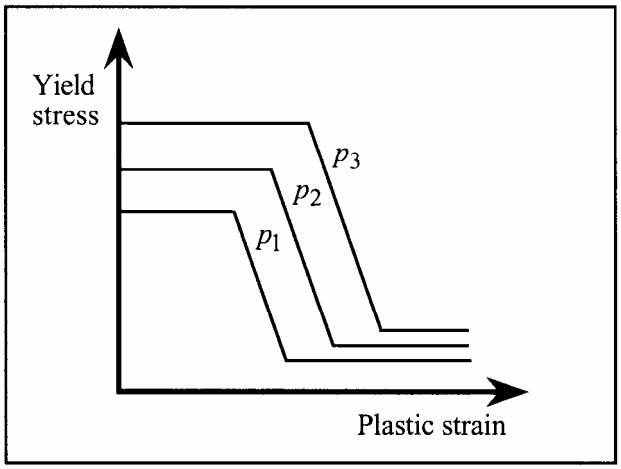


Figure 19.78.3.

Material Model 79: Hysteretic Soil

This model is a nested surface model with five superposed "layers" of elasto-perfectly plastic material, each with its own elastic modulii and yield values. Nested surface models give hysteretic behavior, as the different "layers" yield at different stresses.

The constants a_0, a_1, a_2 govern the pressure sensitivity of the yield stress. Only the ratios between these values are important - the absolute stress values are taken from the stress-strain curve.

The stress strain pairs $(\gamma 1, \tau 1)$, ... $(\gamma 5, \tau 5)$ define a shear stress versus shear strain curve. The first point on the curve is assumed by default to be (0,0) and does not need to be entered. The slope of the curve must decrease with increasing γ. Not all five points need be to be defined. This curve applies at the reference pressure; at other pressures the curve varies according to a_0, a_1, and a_2 as in the soil and crushable foam model, Material 5.

The elastic moduli G and K are pressure sensitive.

$$G = G_0 \left(p - p_0 \right)^b$$
$$K = K_0 \left(p - p_0 \right)^b$$

where G_0 and K_0 are the input values, p is the current pressure, p_0 the cut-off or reference pressure (must be zero or negative). If p attempts to fall below p_0 (i.e., more tensile) the shear stresses are set to zero and the pressure is set to p_0. Thus, the material has no stiffness or strength in tension. The pressure in compression is calculated as follows:

$$p = \left[-K_0 \ln \left(V \right) \right]^{1/1-b}$$

where V is the relative volume, i.e., the ratio between the original and current volume.

Material Model 80: Ramberg-Osgood Plasticity

The Ramberg-Osgood equation is an empirical constitutive relation to represent the one-dimensional elastic-plastic behavior of many materials, including soils. This model allows a simple rate independent representation of the hysteretic energy dissipation observed in soils subjected to cyclic shear deformation. For monotonic loading, the stress-strain relationship is given by:

$$\begin{aligned} \frac{\gamma}{\gamma_y} &= \frac{\tau}{\tau_y} + \alpha \left| \frac{\tau}{\tau_y} \right|^r \quad if \quad \gamma \geq 0 \\ \frac{\gamma}{\gamma_y} &= \frac{\tau}{\tau_y} - \alpha \left| \frac{\tau}{\tau_y} \right|^r \quad if \quad \gamma < 0 \end{aligned} \tag{19.80.1}$$

where γ is the shear and τ is the stress. The model approaches perfect plasticity as the stress exponent $r \to \infty$. These equations must be augmented to correctly model unloading and reloading material behavior. The first load reversal is detected by $\gamma\dot{\gamma} < 0$. After the first reversal, the stress-strain relationship is modified to

$$\begin{aligned} \frac{(\gamma - \gamma_0)}{2\gamma_y} &= \frac{(\tau - \tau_0)}{2\tau_y} + \alpha \left| \frac{(\tau - \tau_0)}{2\tau_y} \right|^r \quad if \quad \gamma \geq 0 \\ \frac{(\gamma - \gamma_0)}{2\gamma_y} &= \frac{(\tau - \tau_0)}{2\tau_y} - \alpha \left| \frac{(\tau - \tau_0)}{2\tau_y} \right|^r \quad if \quad \gamma < 0 \end{aligned} \tag{19.80.2}$$

where γ_0 and τ_0 represent the values of strain and stress at the point of load reversal. Subsequent load reversals are detected by $(\gamma - \gamma_0)\dot{\gamma} < 0$.

The Ramberg-Osgood equations are inherently one-dimensional and are assumed to apply to shear components. To generalize this theory to the multidimensional case, it is assumed that each component of the deviatoric stress and deviatoric tensorial strain is independently related by the one-dimensional stress-strain equations. A projection is used to map the result back into deviatoric stress space if required. The volumetric behavior is elastic, and, therefore, the pressure p is found by

$$p = -K\varepsilon_v \tag{19.80.3}$$

where ε_v is the volumetric strain.

Material Model 81 and 82: Plasticity with Damage and Orthotropic Option

With this model an elasto-viscoplastic material with an arbitrary stress versus strain curve and arbitrary strain rate dependency can be defined. Damage is considered before rupture occurs. Also, failure based on a plastic strain or a minimum time step size can be defined.

An option in the keyword input, ORTHO, is available, which invokes an orthotropic damage model. This option, which is implemented only for shell elements with multiple through thickness integration points, is an extension to include orthotropic damage as a means of treating failure in aluminum panels. 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.

The stress versus strain behavior may be treated by a bilinear stress strain curve by defining the tangent modulus, ETAN. Alternately, a curve similar to that shown in Figure 18.81.1 is expected to be defined by (EPS1,ES1) - (EPS8,ES8); however, an effective stress versus effective plastic strain curve (LCSS) may be input instead if eight points are insufficient. The cost is roughly the same for either approach. The most general approach is to use the table definition (LCSS) discussed below.

Two options to account for strain rate effects are possible. Strain rate may be accounted for using the Cowper-Symonds model which scales the yield stress with the factor,

$$1+\left(\frac{\dot{\varepsilon}}{C}\right)^{1/p} \tag{19.81.1}$$

where $\dot{\varepsilon}$ is the strain rate, $\dot{\varepsilon}=\sqrt{\dot{\varepsilon}_{ij}\dot{\varepsilon}_{ij}}$. If the viscoplastic option is active, VP=1.0, and if SIGY is > 0 then the dynamic yield stress is computed from the sum of the static stress, $\sigma_y^s\left(\varepsilon_{eff}^p\right)$, which is typically given by a load curve ID, and the initial yield stress, SIGY, multiplied by the Cowper-Symonds rate term as follows:

$$\sigma_y\left(\varepsilon_{eff}^p,\dot{\varepsilon}_{eff}^p\right)=\sigma_y^s\left(\varepsilon_{eff}^p\right)+SIGY\cdot\left(\frac{\dot{\varepsilon}_{eff}^p}{C}\right)^{1/p} \tag{19.81.2}$$

where the plastic strain rate is used. With this latter approach similar results can be obtained between this model and material model: *MAT_ANISOTROPIC_VISCOPLASTIC. If SIGY=0, the following equation is used instead where the static stress, $\sigma_y^s\left(\varepsilon_{eff}^p\right)$, must be defined by a load curve:

$$\sigma_y\left(\varepsilon_{eff}^p,\dot{\varepsilon}_{eff}^p\right)=\sigma_y^s\left(\varepsilon_{eff}^p\right)\left[1+\left(\frac{\dot{\varepsilon}_{eff}^p}{C}\right)^{1/p}\right] \tag{19.81.3}$$

This latter equation is always used if the viscoplastic option is off.

For complete generality a load curve (LCSR) to scale the yield stress may be input instead. In this curve the scale factor versus strain rate is defined.

The constitutive properties for the damaged material are obtained from the undamaged material properties. The amount of damage evolved is represented by the constant, ω, which varies from zero if no damage has occurred to unity for complete rupture. For uniaxial loading, the nominal stress in the damaged material is given by

$$\sigma_{nominal} = \frac{P}{A} \tag{19.81.4}$$

where P is the applied load and A is the surface area. The true stress is given by:

$$\sigma_{true} = \frac{P}{A - A_{loss}} \tag{19.81.5}$$

where A_{loss} is the void area. The damage variable can then be defined:

$$\omega = \frac{A_{loss}}{A} \qquad 0 \le \omega \le 1 \tag{19.81.6}$$

In this model damage is defined in terms of plastic strain after the failure strain is exceeded:

$$\omega = \frac{\varepsilon^p_{eff} - \varepsilon^p_{failure}}{\varepsilon^p_{rupture} - \varepsilon^p_{failure}} \quad if \quad \varepsilon^p_{failure} \le \varepsilon^p_{eff} \le \varepsilon^p_{rupture} \tag{19.81.7}$$

After exceeding the failure strain softening begins and continues until the rupture strain is reached.

By default, deletion of the element occurs when all integration points in the shell have failed. A parameter is available, NUMINT, that defines the number of through thickness integration points for element deletion. The default of all integration points is not recommended since elements undergoing large strain are often not deleted due to nodal fiber rotations which limit strains at active integration points after most points have failed. Better results are obtained if NUMINT is set to 1 or a number less than one half of the number of through thickness points. For example, if four through thickness points are used, NUMINT should not exceed 2, even for fully integrated shells which have 16 integration points.

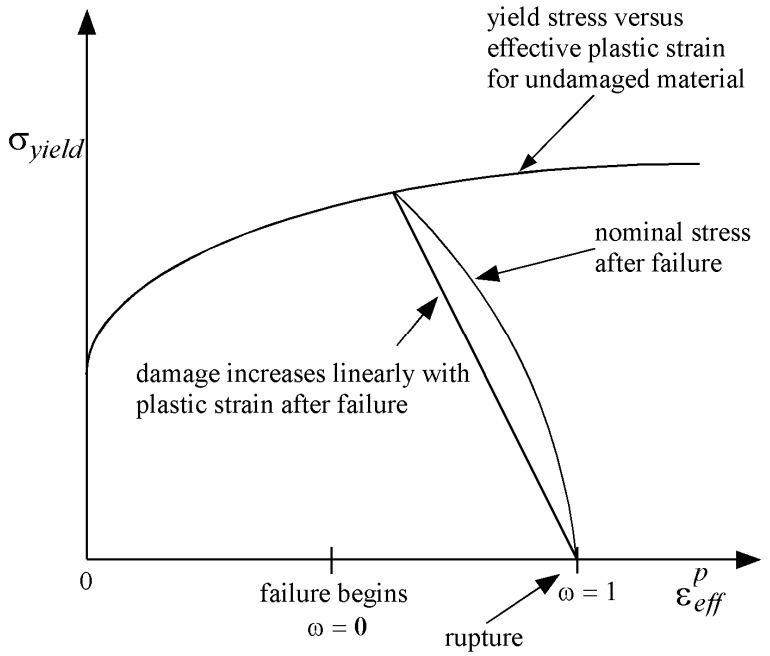


Figure 19.81.1. Stress strain behavior when damage is included.

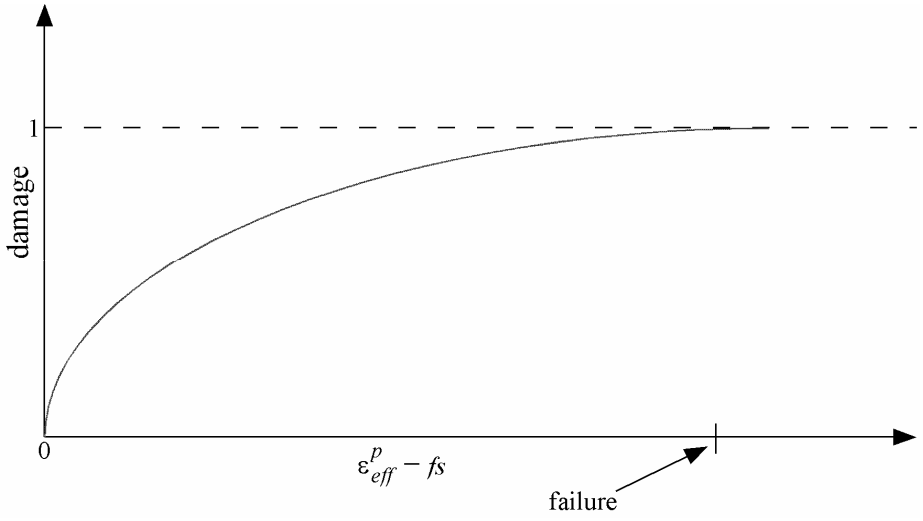


Figure 19.81.2. A nonlinear damage curve is optional. Note that the origin of the curve is at (0,0). It is permissible to input the failure strain, *fs*, as zero for this option. The nonlinear damage curve is useful for controlling the softening behavior after the failure strain is reached.

Material Model 82: Isotropic Elastic-Plastic with Anisotropic Damage

Material 82 is an isotropic elastic-plastic material model with anisotropic damage. The model is available for shell elements only. The stress update is performed as follows. For a given stress state σ_{ij}^t and damage parameters D_i^t, $i=1,2$ at time t, the local stress is obtained as

$$\sigma_{ij}^l = q_{ki} q_{lj} \sigma_{kl}^t , \tag{19.82.1}$$

where q_{ij} is an orthogonal matrix determining the direction of the damage. The directions are determined as follows. The first direction is taken as the one in which the plastic strain first reaches the plastic strain at impending failure, see below. The other direction is orthogonal to the first and in the plane of the shell.

For this local stress, the undamaged stress is computed as

$$\begin{aligned}
\sigma_{11}^u &= \frac{\sigma_{11}^l}{1-D_1^t} \\
\sigma_{22}^u &= \frac{\sigma_{22}^l}{1-D_2^t} \\
\sigma_{12}^u &= \frac{2\sigma_{12}^l}{2-D_1^t-D_2^t} \\
\sigma_{23}^u &= \frac{\sigma_{23}^l}{1-D_2^t} \\
\sigma_{13}^u &= \frac{\sigma_{13}^l}{1-D_1^t}
\end{aligned} \tag{19.82.2}$$

A new undamaged stress σ_{ij}^{u+} is then computed following a standard elastic-plastic stress update. The damage at the next time step is computed according to

$$D_i^{t+} = \max(D_i^t, \frac{\varepsilon_{ii}^p - \varepsilon_f}{\varepsilon_r - \varepsilon_f}) \quad i=1,2 \tag{19.82.3}$$

where ε_f is the plastic strain at impending failure, ε_r is the plastic strain at rupture and ε_{ii}^p is the current plastic strain in the local i direction. There is also an option of defining a nonlinear damage curve, with this option the new damage is computed as

$$D_i^{t+} = \max(D_i^t, f(\varepsilon_{ii}^p - \varepsilon_f)) \;\; i=1,2 \tag{19.82.4}$$

for a user-defined load curve f.

The new local (damaged) stress is given by

$$\begin{aligned}
\sigma_{11}^{l+} &= \sigma_{11}^{u+}(1-D_1^{t+}) \\
\sigma_{22}^{l+} &= \sigma_{22}^{u+}(1-D_2^{t+}) \\
\sigma_{12}^{l+} &= \sigma_{12}^{u+}\frac{2-D_1^{t+}-D_2^{t+}}{2} \\
\sigma_{23}^{l+} &= \sigma_{23}^{u+}(1-D_2^{t+}) \\
\sigma_{13}^{l+} &= \sigma_{13}^{u+}(1-D_1^{t+})
\end{aligned} \tag{19.82.5}$$

which is transformed back to the local system to obtain the new global damaged stress as

$$\sigma_{ij}^{t+} = q_{ik}q_{jl}\sigma_{kl}^{l+} \tag{19.82.6}$$

An integration point is completely failed, i.e., it is removed from the calculations, when $\max(D_1, D_2) > 0.999$. The element is removed from the model when a user specified number of integration points in that element have failed.

There are options of using visco-plasticity in the current model. The details of this part of the stress update is omitted here.

The Rc-Dc Damage Model

The Rc-Dc model is defined as the following, see the report on the Fundamental Study of Crack Initiation and Propagation [2003]. The damage D is updated as

$$D^{t+} = D^t + \omega_1\omega_2\Delta\varepsilon_p \tag{19.82.7}$$

where $\Delta\varepsilon_p$ is the plastic strain increment and

$$\begin{aligned}
\omega_1 &= (1+\gamma p)^{-\alpha} \\
\omega_2 &= (2-A_D)^{\beta}
\end{aligned} \tag{19.82.8}$$

Here p is the pressure, α, β and γ are material parameters and

$$A_D = \begin{cases} 1.9999 & \text{if } \max(S_1, S_2) \le 0 \\ \min\left(\left|\dfrac{S_1}{S_2}\right|, \left|\dfrac{S_2}{S_1}\right|\right) & \text{otherwise} \end{cases} \tag{19.82.9}$$

where S_1 and S_2 are the in-plane principal stress values. Fracture is initiated when the accumulation of damage is greater than a critical damage D_c given by

$$D_c = D_0(1+b\|\nabla D\|^{\lambda}). \tag{19.82.10}$$

Here D_0, b and λ are material parameters and ∇D is the spatial gradient of damage. We have added an option to use a non-local formulation with D as the non-local variable and a characteristic length l. More information on this can be found in the LS-DYNA Keyword User's Manual [Hallquist 2003]. With this option we compute D_c as

$$D_c = D_0, \tag{19.82.11}$$

hence the parameters b and λ are not used. A fracture fraction given by

$$F = \frac{D - D_c}{D_s} \tag{19.82.12}$$

defines the degradations of the material by the Rc-Dc model. Here D_s is yet another parameter determined by the user. The stress update of material 82 is modified accordingly.

Upon entry the stress is divided by the factor $1 - F^t$ to account for the Rc-Dc damage. Before exiting the routine, the stress is multiplied by the new Rc-Fc (reversed) fracture fraction $1 - F^{t+}$. An integration point is considered failed when $\min(1 - D_1, 1 - D_2)(1 - F) < 0.001$.

Material Model 83: Fu-Chang's Foam With Rate Effects

This model allows rate effects to be modeled in low and medium density foams, see Figure 19.83.1. Hysteretic unloading behavior in this model is a function of the rate sensitivity with the most rate sensitive foams providing the largest hysteresis and visa versa. The unified constitutive equations for foam materials by Fu-Chang [1995] provide the basis for this model. This implementation incorporates the coding in the reference in modified form to ensure reasonable computational efficiency. The mathematical description given below is excerpted from the reference.

The strain is divided into two parts: a linear part and a non-linear part of the strain

$$E(t) = E^L(t) + E^N(t) \tag{19.83.1}$$

and the strain rate becomes

$$\dot{E}(t) = \dot{E}^L(t) + \dot{E}^N(t) \tag{19.83.2}$$

$\dot{E}^N$ is an expression for the past history of E^N. A postulated constitutive equation may be written as:

$$\sigma(t) = \int_{\tau=0}^{\infty} \left[E_t^N(\tau), S(t) \right] d\tau \tag{19.83.3}$$

where $S(t)$ is the state variable and $\int_{.\tau=0}^{\infty}$ is a functional of all values of τ in $T_\tau : 0 \le \tau \le \infty$ and

$$E_t^N(\tau) = E^N(t-\tau) \tag{19.83.4}$$

where τ is the history parameter:

$$E_t^N(\tau=\infty) \Leftrightarrow \textit{the virgin material} \tag{19.83.5}$$

It is assumed that the material remembers only its immediate past, i.e., a neighborhood about $\tau = 0$. Therefore, an expansion of $E_t^N(\tau)$ in a Taylor series about $\tau = 0$ yields:

$$E_t^N(\tau) = E^N(0) + \frac{\partial E_t^N}{\partial t}(0)\,dt \tag{19.83.6}$$

Hence, the postulated constitutive equation becomes:

$$\sigma(t) = \sigma^*\left(E^N(t), \dot{E}^N(t), S(t)\right) \tag{19.83.7}$$

where we have replaced $\frac{\partial E_t^N}{\partial t}$ by $\dot{E}^N$, and σ^* is a function of its arguments.

For a special case,

$$\sigma(t) = \sigma^*\left(\dot{E}^N(t), S(t)\right) \tag{19.83.8}$$

we may write

$$\dot{E}_t^N = f\left(S(t), s(t)\right) \tag{19.83.9}$$

which states that the nonlinear strain rate is the function of stress and a state variable which represents the history of loading. Therefore, the proposed kinetic equation for foam materials is:

$$\dot{E}^N = \frac{\sigma}{\|\sigma\|} D_0 \exp\left[-c_0\left(\frac{tr(\sigma S)}{(\|\sigma\|)^2}\right)^{2n_0}\right] \tag{19.83.10}$$

where D_0, c_0, and n_0 are material constants, and S is the overall state variable. If either $D_0 = 0$ or $c_0 \to \infty$ then the nonlinear strain rate vanishes.

$$\dot{S}_{ij} = \left[c_1\left(a_{ij}R - c_2 S_{ij}\right)P + c_3 W^{n_1}\left(\|\dot{E}^N\|\right)^{n_2} I_{ij}\right]R \tag{19.83.11}$$

$$R = 1 + c_4\left(\frac{\|\dot{E}^N\|}{c_5} - 1\right)^{n_3} \tag{19.83.12}$$

$$P = tr\left(\sigma \dot{E}^N\right) \tag{19.83.13}$$

$$W = \int tr\left(\sigma\, dE\right) \tag{19.83.14}$$

where $c_1, c_2, c_3, c_4, c_5, n_1, n_2, n_3,$ and a_{ij} are material constants and:

$$\begin{aligned} \|\sigma\| &= \left(\sigma_{ij}\sigma_{ij}\right)^{\frac{1}{2}} \\ \|\dot{E}\| &= \left(\dot{E}_{ij}\dot{E}_{ij}\right)^{\frac{1}{2}} \\ \|\dot{E}^N\| &= \left(\dot{E}^N_{ij}\dot{E}^N_{ij}\right)^{\frac{1}{2}} \end{aligned} \tag{19.83.15}$$

In the implementation by Fu Chang the model was simplified such that the input constants a_{ij} and the state variables S_{ij} are scalars.

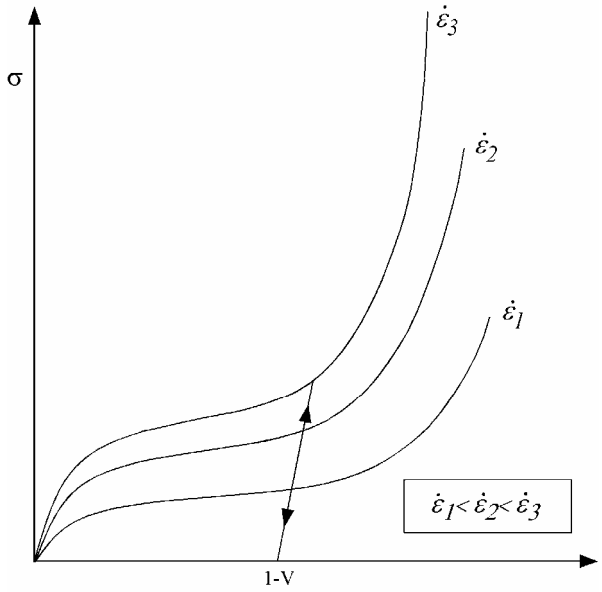


Figure 19.83.1. Rate effects in Fu Chang's foam model.

Material Model 84 and 85: Winfrith Concrete

Pressure is positive in compression; volumetric strain is given by the natural log of the relative volume and is negative in compression. The tabulated data are given in order of increasing compression, with no initial zero point.

If the volume compaction curve is omitted, the following scaled curve is automatically used where p_1 is the pressure at uniaxial compressive failure computed from:

$$p_1 = \frac{\sigma_c}{3}$$

and K is the unloading bulk modulus computed from

$$K = \frac{E_s}{3(1-2\nu)}$$

where E_s is the input tangent modulus for concrete and ν is Poisson's ratio.

Volumetric Strain	**Pressure (MPa)**
$-p_1/K$	1.00xp_1
-0.002	1.50xp_1
-0.004	3.00xp_1
-0.010	4.80xp_1
-0.020	6.00xp_1
-0.030	7.50xp_1
-0.041	9.45xp_1
-0.051	11.55xp_1
-0.062	14.25xp_1
-0.094	25.05xp_1

Table 19.84.1. Default pressure versus volumetric strain curve for concrete if the curve is not defined.

Material Model 87: Cellular Rubber

This material model provides a cellular rubber model combined with linear viscoelasticity as outlined by Christensen [1980].

Rubber is generally considered to be fully incompressible since the bulk modulus greatly exceeds the shear modulus in magnitude. To model the rubber as an unconstrained material a hydrostatic work term, $W_H(J)$, is included in the strain energy functional which is function of the relative volume, J, [Ogden, 1984]:

$$W(J_1, J_2, J) = \sum_{p,q=0}^{n} C_{pq}\left(J_1 - 3\right)^p \left(J_2 - 3\right)^q + W_H\left(J\right)$$
$$J_1 = I_1 I_3^{-1/3} \qquad (19.87.1)$$
$$J_2 = I_2 I_3^{-2/3}$$

In order to prevent volumetric work from contributing to the hydrostatic work the first and second invariants are modified as shown. This procedure is described in more detail by Sussman and Bathe [1987].

The effects of confined air pressure in its overall response characteristics are included by augmenting the stress state within the element by the air pressure.

$$\sigma_{ij} = \sigma_{ij}^{sk} - \delta_{ij}\sigma^{air} \qquad (19.87.2)$$

where σ_{ij}^{sk} is the bulk skeletal stress and σ^{air} is the air pressure computed from the equation:

$$\sigma^{air} = -\frac{p_0 \gamma}{1 + \gamma - \phi} \qquad (19.87.3)$$

where p_0 is the initial foam pressure usually taken as the atmospheric pressure and γ defines the volumetric strain

$$\gamma = V - 1 + \gamma_0 \qquad (19.87.4)$$

where V is the relative volume of the voids and γ_0 is the initial volumetric strain which is typically zero. The rubber skeletal material is assumed to be incompressible.

Rate effects are taken into account through linear viscoelasticity by a convolution integral of the form:

$$\sigma_{ij} = \int_0^t g_{ijkl}\left(t - \tau\right)\frac{\partial \varepsilon_{kl}}{\partial \tau}\, d\tau \qquad (19.87.5)$$

or in terms of the second Piola-Kirchhoff stress, S_{ij}, and Green's strain tensor, E_{ij},

$$S_{ij} = \int_0^t G_{ijkl}\left(t - \tau\right)\frac{\partial E_{kl}}{\partial \tau}\, d\tau \qquad (19.87.6)$$

where $g_{ijkl}(t - \tau)$ and $G_{ijkl}\left(t - \tau\right)$ are the relaxation functions for the different stress measures. This stress is added to the stress tensor determined from the strain energy functional.

Since we wish to include only simple rate effects, the relaxation function is represented by one term from the Prony series:

$$g(t) = \alpha_0 + \sum_{m=1}^{N} \alpha_m e^{-\beta t} \tag{19.87.7}$$

given by,

$$g(t) = E_d e^{-\beta_1 t} \tag{19.87.8}$$

This model is effectively a Maxwell fluid which consists of a damper and spring in series. We characterize this in the input by a shear modulus, G, and decay constant, β_1.

The Mooney-Rivlin rubber model is obtained by specifying *n*=1. In spite of the differences in formulations with Model 27, we find that the results obtained with this model are nearly identical with those of 27 as long as large values of Poisson's ratio are used.

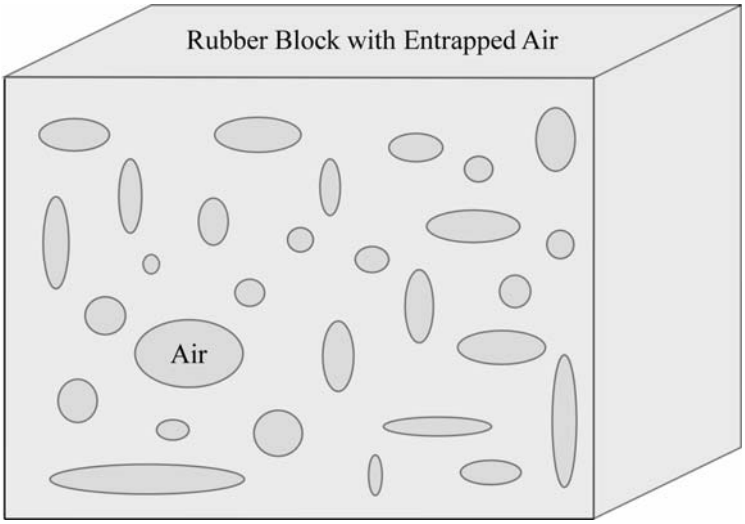


Figure 19.87.1. Cellular rubber with entrapped air. By setting the initial air pressure to zero, an open cell, cellular rubber can be simulated.

Material Model 88: MTS Model

The Mechanical Threshhold Stress (MTS) model is due to Mauldin, Davidson, and Henninger [1990] and is available for applications involving large strains, high pressures and strain rates. As described in the foregoing reference, this model is based on dislocation mechanics and provides a better understanding of the plastic deformation process for ductile materials by using an internal state variable called the mechanical threshold stress. This kinematic quantity tracks the evolution of the material's microstructure along some arbitrary strain, strain rate, and temperature-dependent path using a differential form that balances dislocation generation and recovery processes. Given a value for the mechanical threshold stress, the flow stress is determined using either a thermal-activation-controlled or a drag-controlled kinetics relationship. An equation-of-state is required for solid elements and a bulk modulus must be defined below for shell elements.

The flow stress σ is given by:

$$\sigma = \hat{\sigma}_a + \frac{G}{G_0}\left[s_{th}\hat{\sigma} + s_{th,i}\hat{\sigma}_i + s_{th,s}\hat{\sigma}_s \right] \tag{19.88.1}$$

The first product in the equation for σ contains a micro-structure evolution variable, i.e., $\hat{\sigma}$, called the *Mechanical Threshold Stress* (MTS), that is multiplied by a constant-structure deformation variable $s_{th} : s_{th}$ is a function of absolute temperature T and the plastic strain-rates $\dot{\varepsilon}^P$. The evolution equation for $\hat{\sigma}$ is a differential hardening law representing dislocation-dislocation interactions:

$$\frac{\partial \hat{\sigma}}{\partial \varepsilon^p} \equiv \Theta_o \left[1 - \frac{\tanh\left(\alpha \dfrac{\hat{\sigma}}{\hat{\sigma}_{\varepsilon s}} \right)}{\tanh(\alpha)} \right] \tag{19.88.2}$$

The term, $\dfrac{\partial \hat{\sigma}}{\partial \varepsilon^p}$, represents the hardening due to dislocation generation and the stress ratio, $\dfrac{\hat{\sigma}}{\hat{\sigma}_{\varepsilon s}}$, represents softening due to dislocation recovery. The threshold stress at zero strain-hardening $\hat{\sigma}_{\varepsilon s}$ is called the saturation threshold stress. Relationships for Θ_o, $\hat{\sigma}_{\varepsilon s}$ are:

$$\Theta_o = a_o + a_1 \ln\left(\frac{\dot{\varepsilon}^p}{\varepsilon_0} \right) + a_2 \sqrt{\frac{\dot{\varepsilon}^p}{\varepsilon_0}} \tag{19.88.3}$$

which contains the material constants a_o, a_1, and a_2. The constant, $\hat{\sigma}_{\varepsilon s}$, is given as:

$$\hat{\sigma}_{\varepsilon s} = \hat{\sigma}_{\varepsilon so} \left(\frac{\dot{\varepsilon}^p}{\dot{\varepsilon}_{\varepsilon so}} \right)^{kT/Gb^3 A} \tag{19.88.4}$$

which contains the input constants: $\hat{\sigma}_{\varepsilon so}$, $\dot{\varepsilon}_{\varepsilon so}$, b, A, and k. The shear modulus G appearing in these equations is assumed to be a function of temperature and is given by the correlation.

$$G = G_0 - b_1 / (e^{b_2/T} - 1) \tag{19.88.5}$$

which contains the constants: G_0, b_1, and b_2. For thermal-activation controlled deformation s_{th} is evaluated via an Arrhenius rate equation of the form:

$$s_{th} = \left[1 - \left(\frac{kT \ln\left(\frac{\dot{\varepsilon}_0}{\dot{\varepsilon}^p} \right)}{Gb^3 g_0} \right)^{\frac{1}{q}} \right]^{\frac{1}{p}} \tag{19.88.6}$$

The absolute temperature is given as:

$$T = T_{ref} + \rho c_p E \tag{19.88.7}$$

where E in the internal energy density per unit initial volume.

Material Model 89: Plasticity Polymer

Unlike other LS-DYNA material models, both the input stress-strain curve and the strain to failure are defined as total true strain, not plastic strain. The input can be defined from uniaxial tensile tests; nominal stress and nominal strain from the tests must be converted to true stress and true strain. The elastic component of strain must not be subtracted out.

The stress-strain curve is permitted to have sections steeper (i.e. stiffer) than the elastic modulus. When these are encountered the elastic modulus is increased to prevent spurious energy generation.

Material Model 90: Acoustic

This model is appropriate for tracking low-pressure stress waves in an acoustic media such as air or water and can be used only with the acoustic pressure element formulation. The acoustic pressure element requires only one unknown per node. This element is very cost effective.

Material Model 91: Soft Tissue

The overall strain energy W is "uncoupled" and includes two isotropic deviatoric matrix terms, a fiber term F, and a bulk term:

$$W = C_1\left(\tilde{I}_1 - 3 \right) + C_2\left(\tilde{I}_2 - 3 \right) + F\left(\lambda \right) + \tfrac{1}{2} K\left[\ln\left(J \right) \right]^2 \tag{19.91.1}$$

Here, $\tilde{I}_1$ and $\tilde{I}_2$ are the deviatoric invariants of the right Cauchy deformation tensor, λ is the deviatoric part of the stretch along the current fiber direction, and $J = \det \mathbf{F}$ is the volume ratio. The material coefficients C_1 and C_2 are the Mooney-Rivlin coefficients, while K is the effective bulk modulus of the material (input parameter XK).

The derivatives of the fiber term F are defined to capture the behavior of crimped collagen. The fibers are assumed to be unable to resist compressive loading - thus the model is isotropic when $\lambda < 1$. An exponential function describes the straightening of the fibers, while a linear function describes the behavior of the fibers once they are straightened past a critical fiber stretch level $\lambda \geq \lambda^*$ (input parameter XLAM):

$$\frac{\partial F}{\partial \lambda} = \begin{Bmatrix} 0 & \lambda < 1 \\ \frac{C_3}{\lambda}\left[\exp\left(C_4(\lambda-1)\right)-1\right] & \lambda < \lambda^* \\ \frac{1}{\lambda}\left(C_5\lambda + C_6\right) & \lambda \geq \lambda^* \end{Bmatrix} \qquad (19.91.2)$$

Coefficients C_3, C_4, and C_5 must be defined by the user. C_6 is determined by LS-DYNA to ensure stress continuity at $\lambda = \lambda^*$. Sample values for the material coefficients $C_1 - C_5$ and λ^* for ligament tissue can be found in Quapp and Weiss [1998]. The bulk modulus K should be at least 3 orders of magnitude larger than C_1 to ensure near-incompressible material behavior.

Viscoelasticity is included via a convolution integral representation for the time-dependent second Piola-Kirchoff stress $\mathbf{S}(\mathbf{C},t)$:

$$\mathbf{S}(\mathbf{C},t) = \mathbf{S}^e(\mathbf{C}) + \int_0^t 2G(t-s)\frac{\partial W}{\partial \mathbf{C}(s)}ds \qquad (19.91.3)$$

Here, $\mathbf{S}^e$ is the elastic part of the second PK stress as derived from the strain energy, and $G(t-s)$ is the reduced relaxation function, represented by a Prony series:

$$G(t) = \sum_{i=1}^{6} S_i \exp\left(\frac{t}{T_i}\right) \qquad (19.91.4)$$

Puso and Weiss [1998] describe a graphical method to fit the Prony series coefficients to relaxation data that approximates the behavior of the continuous relaxation function proposed by Y-C. Fung, as quasilinear viscoelasticity.

Material Model 94: Inelastic Spring Discrete Beam

The yield force is taken from the load curve:

$$F^Y = F_y\left(\Delta L^{plastic}\right) \qquad (19.94.1)$$

where $L^{plastic}$ is the plastic deflection. A trial force is computed as:

$$F^T = F^n + K \cdot \Delta\dot{L} \cdot \Delta t \qquad (19.94.2)$$

and is checked against the yield force to determine F:

$$F = \begin{cases} F^Y \ if \ F^T > F^Y \\ F^T \ if \ F^T \leq F^Y \end{cases} \qquad (19.94.3)$$

The final force, which includes rate effects and damping, is given by:

$$F^{n+1} = F \cdot \left[1 + C1 \cdot \Delta\dot{L} + C2 \cdot \mathrm{sgn}\left(\Delta\dot{L}\right) \ln\left(\max\left\{ 1., \frac{\left|\Delta\dot{L}\right|}{DLE} \right\} \right) \right] + D\Delta\dot{L} + g\left(\Delta L\right) h\left(\Delta\dot{L}\right)$$

where $C1$, $C2$ are damping coefficients, *DLE* is a factor to scale time units.

Unless the origin of the curve starts at (0,0), the negative part of the curve is used when the spring force is negative where the negative of the plastic displacement is used to interpolate, F_y. The positive part of the curve is used whenever the force is positive. In these equations, ΔL is the change in length

$$\Delta L = current\ length - initial\ length$$

Material Model 96: Brittle Damage Model

A full description of the tensile and shear damage parts of this material model is given in Govindjee, Kay and Simo [1994,1995]. It is an anisotropic brittle damage model designed primarily for concrete, though it can be applied to a wide variety of brittle materials. It admits progressive degradation of tensile and shear strengths across smeared cracks that are initiated under tensile loadings. Compressive failure is governed by a simplistic J2 flow correction that can be disabled if not desired. Damage is handled by treating the rank 4 elastic stiffness tensor as an evolving internal variable for the material. Softening induced mesh dependencies are handled by a characteristic length method [Oliver 1989].

Description of properties:

1. E is the Young's modulus of the undamaged material also known as the virgin modulus.

2. υ is the Poisson's ratio of the undamaged material also known as the virgin Poisson's ratio.

3. f_n is the initial principal tensile strength (stress) of the material. Once this stress has been reached at a point in the body a smeared crack is initiated there with a normal that is co-linear with the 1st principal direction. Once initiated, the crack is fixed at that location, though it will convect with the motion of the body. As the loading progresses the allowed tensile traction normal to the crack plane is progressively degraded to a small machine dependent constant.

   The degradation is implemented by reducing the material's modulus normal to the smeared crack plane according to a maximum dissipation law that incorporates exponential softening. The restriction on the normal tractions is given by

$$\phi_t = \left(\mathbf{n} \otimes \mathbf{n}\right) : \sigma - f_n + \left(1 - \varepsilon\right) f_n \left(1 - \exp\left[-H\alpha\right]\right) \le 0$$

   where **n** is the smeared crack normal, ε is the small constant, H is the softening modulus, and α is an internal variable. H is set automatically by the program; see g_c

below. α measures the crack field intensity and is output in the equivalent plastic strain field, $\bar{\varepsilon}^p$, in a normalized fashion.

The evolution of alpha is governed by a maximum dissipation argument. When the normalized value reaches unity it means that the material's strength has been reduced to 2% of its original value in the normal and parallel directions to the smeared crack. Note that for plotting purposes, it is never output greater than 5.

4. f_s is the initial shear traction that may be transmitted across a smeared crack plane. The shear traction is limited to be less than or equal to $f_s(1-\beta)\left(1-\exp\left[-H\alpha\right]\right)$, through the use of two orthogonal shear damage surfaces. Note that the shear degradation is coupled to the tensile degradation through the internal variable alpha which measures the intensity of the crack field. β is the shear retention factor defined below. The shear degradation is taken care of by reducing the material's shear stiffness parallel to the smeared crack plane.

5. g_c is the fracture toughness of the material. It should be entered as fracture energy per unit area crack advance. Once entered the softening modulus is automatically calculated based on element and crack geometries.

6. β is the shear retention factor. As the damage progresses the shear tractions allowed across the smeared crack plane asymptote to the product βf_s.

7. η represents the viscosity of the material. Viscous behavior is implemented as a simple Perzyna regularization method. This allows for the inclusion of first order rate effects. The use of some viscosity is recommend as it serves as regularizing parameter that increases the stability of calculations.

8. σ_y is a uniaxial compressive yield stress. A check on compressive stresses is made using the $J2$ yield function $\mathbf{s}:\mathbf{s}-\sqrt{\frac{2}{3}}\sigma_y \le 0$, where $\mathbf{s}$ is the stress deviator. If violated, a $J2$ return mapping correction is executed. This check is executed when (1) no damage has taken place at an integration point yet, (2) when damage has taken place at a point but the crack is currently closed, and (3) during active damage after the damage integration (ie. as an operator split). Note that if the crack is open, the plasticity correction is done in the plane-stress subspace of the crack plane.

Remark: A variety of experimental data has been replicated using this model from quasi-static to explosive situations. Reasonable properties for a standard grade concrete would be E=3.15x10^6 psi, f_n=450 psi, f_s=2100 psi, ν = 0.2, g_c = 0.8 lbs/in, β =0.03, η = 0.0 psi-sec, σ_y = 4200 psi. For stability, values of η between 104 to 106 psi/sec are recommended. Our limited experience thus far has shown that many problems require nonzero values of η to run to avoid error terminations. Various other internal variables such as crack orientations and degraded stiffness tensors are internally calculated but currently not available for output.

Material Model 97: General Joint Discrete Beam

For explicit calculations, the additional stiffness due to this joint may require additional mass and inertia for stability. Mass and rotary inertia for this beam element is based on the defined mass density, the volume, and the mass moment of inertia defined in the *SECTION_ BEAM input.

The penalty stiffness applies to explicit calculations. For implicit calculations, constraint equations are generated and imposed on the system equations; therefore, these constants, RPST and RPSR, are not used.

Material Model 98: Simplified Johnson Cook

Johnson and Cook express the flow stress as

$$\sigma_y = \left(A + B\,\bar{\varepsilon}^{\,p^n} \right)\left(1 + c\ \ln \dot{\varepsilon}^* \right)$$

where

A, B, C and n are input constants

$\bar{\varepsilon}^p$ effective plastic strain

$\dot{\varepsilon}^* = \dfrac{\dot{\bar{\varepsilon}}}{\dot{\varepsilon}_0}$ effective strain rate for $\dot{\varepsilon}_0 = 1\,s^{-1}$

The maximum stress is limited by *sigmax* and *sigsat* by:

$$\sigma_y = \min\left\{ \min\left[A + B\bar{\varepsilon}^{\,p^n}, sigmax \right]\left(1 + c \ln \dot{\varepsilon}^* \right), sigsat \right\}$$

Failure occurs when the effective plastic strain exceeds *psfail*.

If the viscoplastic option is active, VP=1.0, the parameters SIGMAX and SIGSAT are ignored since these parameters make convergence of the viscoplastic strain iteration loop difficult to achieve. The viscoplastic option replaces the plastic strain in the forgoing equations by the viscoplastic strain and the strain rate by the viscoplastic strain rate. Numerical noise is substantially reduced by the viscoplastic formulation.

Material Model 100: Spot Weld

This material model applies to beam element type 9 for spot welds. These beam elements may be placed between any two deformable shell surfaces, see Figure 19.100.1, and tied with type 7 constraint contact which eliminates the need to have adjacent nodes at spot weld locations. Beam spot welds may be placed between rigid bodies and rigid/deformable bodies by making the node on one end of the spot weld a rigid body node which can be an extra node for the rigid body. In the same way, rigid bodies may also be tied together with this spot weld option.

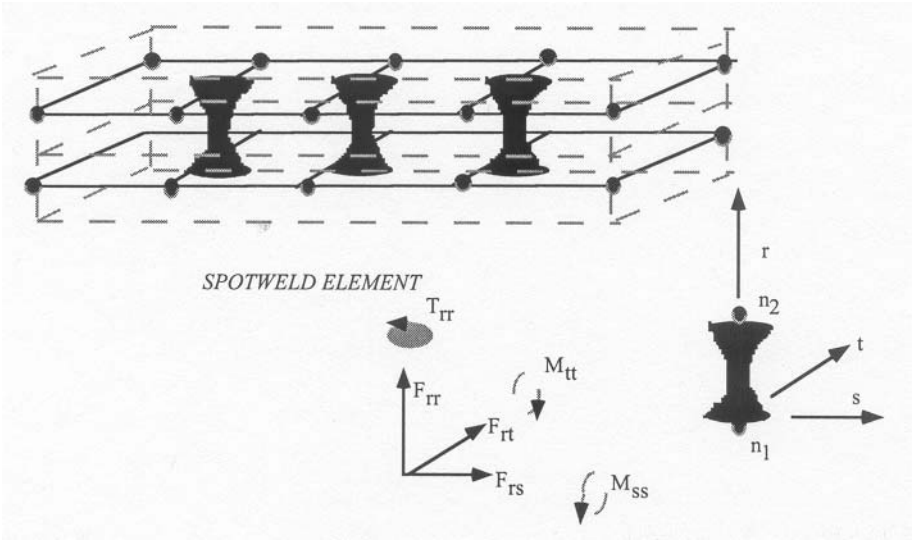


Figure 19.100.1. Deformable spotwelds can be arbitrarily placed within the structure.

It is advisable to include all spot welds, which provide the slave nodes, and spot welded materials, which define the master segments, within a single type 7 tied interface. As a constraint method, multiple type 7 interfaces are treated independently which can lead to significant problems if such interfaces share common nodal points. The offset option, "o 7", should not be used with spot welds.

The DAMAGE-FAILURE option causes one additional line to be read with the damage parameter and a flag that determines how failure is computed from the resultants. On this line the parameter, DMG, if nonzero, invokes damage mechanics combined with the plasticity model to achieve a smooth drop off of the resultant forces prior to the removal of the spot weld. The parameter FOPT determines the method used in computing resultant based failure, which is unrelated to damage.

The weld material is modeled with isotropic hardening plasticity coupled to two failure models. The first model specifies a failure strain which fails each integration point in the spot weld independently. The second model fails the entire weld if the resultants are outside of the failure surface defined by:

$$\left(\frac{N_{rr}}{N_{rr_F}}\right)^2+\left(\frac{N_{rs}}{N_{rs_F}}\right)^2+\left(\frac{N_{rt}}{N_{rt_F}}\right)^2+\left(\frac{M_{rr}}{M_{rr_F}}\right)^2+\left(\frac{M_{ss}}{M_{ss_F}}\right)^2+\left(\frac{T_{rr}}{T_{rr_F}}\right)^2-1=0 \quad (19.100.1)$$

where the *numerators* in the equation are the resultants calculated in the local coordinates of the cross section, and the **denominators** are the values specified in the input. If the user defined parameter, NF, which the number of force vectors stored for filtering, is nonzero the resultants are filtered before failure is checked. The default value is set to zero which is generally recommended unless oscillatory resultant forces are observed in the time history databases. Even though these welds should not oscillate significantly, this option was added for consistency with the other spot weld options. NF affects the storage since it is necessary to store the resultant forces as history variables.

If the failure strain is set to zero, the failure strain model is not used. In a similar manner, when the value of a resultant at failure is set to zero, the corresponding term in the failure surface is ignored. For example, if only N_{rr_F} is nonzero, the failure surface is reduced to $|N_{rr}| = N_{rrF}$. None, either, or both of the failure models may be active depending on the specified input values.

The inertias of the spot welds are scaled during the first time step so that their stable time step size is Δt. A strong compressive load on the spot weld at a later time may reduce the length of the spot weld so that stable time step size drops below Δt. If the value of Δt is zero, mass scaling is not performed, and the spot welds will probably limit the time step size. Under most circumstances, the inertias of the spot welds are small enough that scaling them will have a negligible effect on the structural response and the use of this option is encouraged.

Spotweld force history data is written into the SWFORC ASCII file. In this database the resultant moments are not available, but they are in the binary time history database.

The constitutive properties for the damaged material are obtained from the undamaged material properties. The amount of damage evolved is represented by the constant, ω, which varies from zero if no damage has occurred to unity for complete rupture. For uniaxial loading, the nominal stress in the damaged material is given by

$$\sigma_{nominal} = \frac{P}{A} \tag{19.100.2}$$

where P is the applied load and A is the surface area. The true stress is given by:

$$\sigma_{true} = \frac{P}{A - A_{loss}} \tag{19.100.3}$$

where A_{loss} is the void area. The damage variable can then be defined:

$$\omega = \frac{A_{loss}}{A} \qquad 0 \le \omega \le 1 \tag{19.100.4}$$

In this model damage is defined in terms of plastic strain after the failure strain is exceeded:

$$\omega = \frac{\varepsilon^p_{eff} - \varepsilon^p_{failure}}{\varepsilon^p_{rupture} - \varepsilon^p_{failure}} \quad if \quad \varepsilon^p_{failure} \le \varepsilon^p_{eff} \le \varepsilon^p_{rupture} \tag{19.100.5}$$

After exceeding the failure strain softening begins and continues until the rupture strain is reached.

Material Model 101: GE Thermoplastics

The constitutive model for this approach is:

$$\dot{\varepsilon}_p = \dot{\varepsilon}_0 \exp(A\{\sigma - S(\varepsilon_p)\}) \times \exp(-p\alpha A)$$

where $\dot{\varepsilon}_0$ and A are rate dependent yield stress parameters, $S(\varepsilon_p)$ internal resistance (strain hardening) and α is a pressure dependence parameter.

In this material the yield stress may vary throughout the finite element model as a function of strain rate and hydrostatic stress. Post yield stress behavior is captured in material softening and hardening values. Finally, ductile or brittle failure measured by plastic strain or maximum principal stress respectively is accounted for by automatic element deletion.

Although this may be applied to a variety of engineering thermoplastics, GE Plastics have constants available for use in a wide range of commercially available grades of their engineering thermoplastics.

Material Model 102: Hyperbolic Sine

Resistance to deformation is both temperature and strain rate dependent. The flow stress equation is:

$$\sigma = \frac{1}{\alpha}\sinh^{-1}\left(\left[\frac{Z}{A}\right]^{\frac{1}{N}}\right) \qquad (19.102.1)$$

where Z, the Zener-Holloman temperature compensated strain rate, is:

$$Z = \dot{\varepsilon}\exp\left(\frac{Q}{GT}\right) \qquad (19.102.2)$$

The units of the material constitutive constants are as follows: A (1/sec), N (dimensionless), α (1/MPa), the activation energy for flow, Q(J/mol), and the universal gas constant, G [J/$(\text{mol}\cdot\text{K})$]. The value of G will only vary with the unit system chosen. Typically it will be either 8.3145 J/$(\text{mol}\cdot\text{K})$, or 40.8825 $\text{lb}\cdot\text{in/mol}\cdot\text{R}$.

The final equation necessary to complete the description of high strain rate deformation herein is one that allows computation of the temperature change during the deformation. In the absence of a coupled thermo-mechanical finite element code we assume adiabatic temperature change and follow the empirical assumption that 90-95% of the plastic work is dissipated as heat. Thus the heat generation coefficient is

$$HC \approx \frac{0.9}{\rho C_v}$$

where ρ is the material density and C_v is the specific heat.

Material Model 103: Anisotropic Viscoplastic

The uniaxial stress-strain curve is given on the following form

$$\begin{aligned}\sigma(p,\dot{p}) = \sigma_0 &+ Q_{r1}(1-\exp(-C_{r1}p)) + Q_{r2}(1-\exp(-C_{r2}p)) \\ &+ Q_{\chi 1}(1-\exp(-C_{\chi 1}p)) + Q_{\chi 2}(1-\exp(-C_{\chi 2}p)) \\ &+ V_k \dot{p}^{V_m}\end{aligned} \tag{19.103.1}$$

For bricks the following yield criteria is used

$$\begin{aligned}F(\sigma_{22}-\sigma_{33})^2 + G(\sigma_{33}-\sigma_{11})^2 + H(\sigma_{11}-\sigma_{22})^2 \\ + 2L\sigma_{23}^2 + 2M\sigma_{31}^2 + 2N\sigma_{12}^2 = \sigma(\varepsilon^p, \dot{\varepsilon}^p)\end{aligned} \tag{19.103.2}$$

where ε^p is the effective plastic strain and $\dot{\varepsilon}^p$ is the effective plastic strain rate. For shells the anisotropic behavior is given by R_{00}, R_{45} and R_{90}. When $V_k = 0$ the material will behave elasto-plastically. Default values are given by:

$$F = G = H = \frac{1}{2} \tag{19.103.3a}$$

$$L = M = N = \frac{3}{2} \tag{19.103.3b}$$

$$R_{00} = R_{45} = R_{90} = 1 \tag{19.103.3c}$$

Strain rate is accounted for using the Cowper-Symonds model which, e.g., model 3, scales the yield stress with the factor:

$$1 + \left(\frac{\dot{\varepsilon}}{C}\right)^{1/p} \tag{19.103.4}$$

To convert these constants set the viscoelastic constants, V_k and V_m, to the following values:

$$\begin{aligned}V_k &= \sigma\left(\frac{1}{C}\right)^{\frac{1}{p}} \\ V_m &= \frac{1}{p}\end{aligned} \tag{19.103.5}$$

This model properly treats rate effects and should provide superior results to models 3 and 24.

Material Model 104: Continuum Damage Mechanics Model

Anisotropic Damage model (FLAG = -1). At each thickness integration points, an anisotorpic damage law acts on the plane stress tensor in the directions of the principal total shell strains, ε_1 and ε_2, as follows:

$$\begin{aligned} \sigma_{11} &= (1-D_1(\varepsilon_1))\sigma_{110} \\ \sigma_{22} &= (1-D_2(\varepsilon_2))\sigma_{220} \\ \sigma_{12} &= (1-(D_1+D_2)/2)\sigma_{120} \end{aligned} \tag{19.104.1}$$

The transverse plate shear stresses in the principal strain directions are assumed to be damaged as follows:

$$\begin{aligned} \sigma_{13} &= (1-D_1/2)\sigma_{130} \\ \sigma_{23} &= (1-D_2/2)\sigma_{230} \end{aligned} \tag{19.104.2}$$

In the anisotropic damage formulation, $D_1(\varepsilon_1)$ and $D_2(\varepsilon_2)$ are anisotropic damage functions for the loading directions 1 and 2, respectively. Stresses $\sigma_{110}, \sigma_{220}, \sigma_{120}, \sigma_{130}$ and σ_{230} are stresses in the principal shell strain directions as calculated from the undamaged elastic-plastic material behavior. The strains ε_1 and ε_2 are the magnitude of the principal strains calculated upon reaching the damage thresholds. Damage can only develop for tensile stresses, and the damage functions $D_1(\varepsilon_1)$ and $D_2(\varepsilon_2)$ are identical to zero for negative strains ε_1 and ε_2. The principal strain directions are fixed within an integration point as soon as either principal strain exceeds the initial threshold strain in tension. A more detailed description of the damage evolution for this material model is given in the description of material 82.

The Continuum Damage Mechanics (CDM) model (FLAG≥0) is based on a CDM model proposed by Lemaitre [1992]. The effective stress $\tilde{\sigma}$, which is the stress calculated over the section that effectively resist the forces and reads.

$$\tilde{\sigma} = \frac{\sigma}{1-D} \tag{19.104.3}$$

where D is the damage variable. The evolution equation for the damage variable is defined as

$$\dot{D} = \begin{cases} \dfrac{Y}{S(1-D)}\dot{r} & \text{for} \quad r>r_D \quad \text{and} \quad \sigma_1 > 0 \\ 0 & \textit{otherwise} \end{cases} \tag{19.104.4}$$

where r_D is the damage threshold, Y is a positive material constant, S is the strain energy release rate, and σ_1 is the maximal principal stress. The strain energy density release rate is

$$Y = \frac{1}{2}\mathbf{e_e} : \mathbf{C} : \mathbf{e_e} = \frac{\sigma_{vm}^2 R_v}{2E(1-D)^2} \tag{19.104.5}$$

where σ_{vm} is the equivalent von Mises stress. The triaxiality function R_v is defined as

$$R_v = \frac{2}{3}(1+\nu) + 3(1-2\nu)\left(\frac{\sigma_H}{\sigma_{vm}}\right)^2 \tag{19.104.6}$$

The uniaxial stress-strain curve is given in the following form

$$\begin{aligned}\sigma(r,\dot{\varepsilon}_{eff}^{p}) = \sigma_0 + Q_1(1-\exp(-C_1 r)) + Q_2(1-\exp(-C_2 r)) \\ +V_k \dot{\varepsilon}_{eff}^{p\ V_m}\end{aligned} \tag{19.104.7}$$

where r is the damage accumulated plastic strain, which can be calculated by

$$\dot{r} = \dot{\varepsilon}_{eff}^{p}(1-D) \tag{19.104.8}$$

For bricks the following yield criteria is used

$$\begin{aligned}F(\tilde{\sigma}_{22}-\tilde{\sigma}_{33})^2 + G(\tilde{\sigma}_{33}-\tilde{\sigma}_{11})^2 + H(\tilde{\sigma}_{11}-\tilde{\sigma}_{22})^2 \\ + 2L\tilde{\sigma}_{23}^2 + 2M\tilde{\sigma}_{31}^2 + 2N\tilde{\sigma}_{12}^2 = \sigma(r,\dot{\varepsilon}_{eff}^{p})\end{aligned} \tag{19.104.9}$$

where r is the damage effective viscoplastic strain and $\dot{\varepsilon}_{eff}^{p}$ is the effective viscoplastic strain rate. For shells the anisotropic behavior is given by the R-values: R_{00}, R_{45}, and R_{90}. When $V_k = 0$ the material will behave as an elastoplastic material without rate effects. Default values for the anisotropic constants are given by:

$$\begin{aligned}F = G = H = \frac{1}{2} \\ L = M = N = \frac{3}{2} \\ R_{00} = R_{45} = R_{90} = 1\end{aligned} \tag{19.104.10}$$

so that isotropic behavior is obtained.

Strain rate is accounted for using the Cowper-Symonds model which scales the yield stress with the factor:

$$1 + \left(\frac{\dot{\varepsilon}}{C}\right)^{1/p} \tag{19.104.11}$$

To convert these constants, set the viscoelastic constants, V_k and V_m, to the following values:

$$V_k = \sigma\left(\frac{1}{C}\right)^{\frac{1}{p}}$$

$$V_m = \frac{1}{p} \tag{19.104.12}$$

Material Model 106: Elastic Viscoplastic Thermal

If LCSS is not given any value the uniaxial stress-strain curve has the form

$$\begin{aligned}\sigma(\varepsilon_{eff}^p) = \sigma_0 + Q_{r1}(1-\exp(-C_{r1}\varepsilon_{eff}^p)) + Q_{r2}(1-\exp(-C_{r2}\varepsilon_{eff}^p)) \\ +Q_{\chi 1}(1-\exp(-C_{\chi 1}\varepsilon_{eff}^p)) + Q_{\chi 2}(1-\exp(-C_{\chi 2}\varepsilon_{eff}^p))\end{aligned} \tag{19.106.1}$$

Viscous effects are accounted for using the Cowper-Symonds model, which, scales the yield stress with the factor:

$$1+\left(\frac{\dot{\varepsilon}_{eff}^p}{C}\right)^{1/p} \tag{19.106.2}$$

Material Model 110: Johnson-Holmquist Ceramic Model

The Johnson-Holmquist plasticity damage model is useful for modeling ceramics, glass and other brittle materials. A more detailed description can be found in a paper by Johnson and Holmquist [1993].

The equivalent stress for a ceramic-type material is given in terms of the damage parameter *D* by

$$\sigma^* = \sigma_i^* - D(\sigma_i^* - \sigma_f^*) \tag{19.110.1}$$

Here,

$$\sigma_i^* = a(p^* + t^*)^n(1 + c\ln\dot{\varepsilon}^*) \tag{19.110.2}$$

represents the intact, undamaged behavior. The superscript, '*', indicates a normalized quantity. The stresses are normalized by the equivalent stress at the Hugoniot elastic limit (see below), the pressures are normalized by the pressure at the Hugoniot elastic limit, and the strain rate by the reference strain rate defined in the input. In this equation *a* is the intact normalized strength parameter, *c* is the strength parameter for strain rate dependence, $\dot{\varepsilon}^*$ is the normalized plastic strain rate, and,

$$t^* = \frac{T}{phel} \tag{19.110.3}$$

$$p^* = \frac{p}{phel} \tag{19.110.4}$$

where *T* is the maximum tensile strength, *phel* is the pressure component at the Hugoniot elastic limit, and *p* is the pressure.

$$D = \sum \Delta \varepsilon^p / \varepsilon_f^p \tag{19.110.5}$$

represents the accumulated damage based upon the increase in plastic strain per computational cycle and the plastic strain to fracture

$$\varepsilon_f^p = d_1 (p^* + t^*)^{d_2} \tag{19.110.6}$$

where d_1 and d_2 are user defined input parameters. The equation:

$$\sigma_f^* = b(p^*)^m (1 + c \ln \dot{\varepsilon}^*) \le sfma \tag{19.110.7}$$

represents the damaged behavior where *b* is an input parameter and, *sfma*, is the maximum normalized fracture strength. The parameter, d_1, controls the rate at which damage accumulates. If it approaches 0, full damage can occur in one time step, i.e., instantaneously. This rate parameter is also the best parameter to vary if one attempts to reproduce results generated by another finite element program.

In undamaged material, the hydrostatic pressure is given by

$$p = k_1 \mu + k_2 \mu^2 + k_3 \mu^3 \tag{19.110.8}$$

where $\mu = \rho / \rho_0 - 1$. When damage starts to occur, there is an increase in pressure. A fraction defined in the input, between 0 and 1, of the elastic energy loss, β, is converted into hydrostatic potential energy, which results in an increase in pressure. The details of this pressure increase are given in the reference.

Given *hel* and the shear modulus, *g*, μ_{hel} can be found iteratively from

$$hel = k_1 \mu_{hel} + k_2 \mu_{hel}^2 + k_3 \mu_{hel}^3 + (4/3) g (\mu_{hel} / (1 + \mu_{hel}) \tag{19.110.9}$$

and, subsequently, for normalization purposes,

$$p_{hel} = k_1 \mu_{hel} + k_2 \mu_{hel}^2 + k_3 \mu_{hel}^3 \tag{19.110.10}$$

and

$$\sigma_{hel} = 1.5(hel - p_{hel}) \tag{19.110.11}$$

These are calculated automatically by LS-DYNA if p_{hel} is zero on input.

Material Model 111: Johnson-Holmquist Concrete Model

This model can be used for concrete subjected to large strains, high strain rates, and high pressures. The equivalent strength is expressed as a function of the pressure, strain rate, and damage. The pressure is expressed as a function of the volumetric strain and includes the effect of permanent crushing. The damage is accumulated as a function of the plastic volumetric strain, equivalent plastic strain and pressure. A more detailed description of this model can be found in the paper by Holmquist, Johnson, and Cook [1993]

The normalized equivalent stress is defined as

$$\sigma^* = \frac{\sigma}{f_c'} \tag{19.111.1}$$

where σ is the actual equivalent stress, and f_c' is the quasi-static uniaxial compressive strength. The yield stress is defined in terms of the input parameters *a*, *b*, *c*, and *n* as:

$$\sigma^* = \left[a(1-D) + bp^{*n}\right]\left[1 - c\ln\left(\dot{\varepsilon}^*\right)\right] \tag{19.111.2}$$

where D is the damage parameter, $p^* = p / f_c'$ is the normalized pressure and $\dot{\varepsilon}^* = \dot{\varepsilon} / \dot{\varepsilon}_0$ is the dimensionless strain rate. The model accumulates damage both from equivalent plastic strain and plastic volumetric strain, and is expressed as

$$D = \sum \frac{\Delta\varepsilon_p + \Delta\mu_p}{D_1(p^* + T^*)^{D_2}} \tag{19.111.3}$$

where $\Delta\varepsilon_p$ and $\Delta\mu_p$ are the equivalent plastic strain and plastic volumetric strain, D_1 and D_2 are material constants and $T^* = T / f_c'$ is the normalized maximum tensile hydrostatic pressure where *T* is the maximum tensile hydrostatic pressure.

The pressure for fully dense material is expressed as:

$$P = K_1\bar{\mu} + K_2\bar{\mu}^2 + K_3\bar{\mu}^3 \tag{19.111.4}$$

where K_1 , K_2 and K_3 are material constants and the modified volumetric strain is defined as

$$\bar{\mu} = \frac{\mu - \mu_{lock}}{1 + \mu_{lock}} \tag{19.111.5}$$

where μ_{lock} is the locking volumetric strain.

Material Model 115: Elastic Creep Model

The effective creep strain, $\bar{\varepsilon}^c$, given as:

$$\bar{\varepsilon}^c = A\bar{\sigma}^n \bar{t}^m \tag{19.115.1}$$

where *A*, *n*, and *m* are constants and $\bar{t}$ is the effective time. The effective stress, $\bar{\sigma}$, is defined as:

$$\bar{\sigma} = \sqrt{\frac{3}{2}\sigma_{ij}\sigma_{ij}} \tag{19.115.2}$$

The creep strain, therefore, is only a function of the deviatoric stresses. The volumetric behavior for this material is assumed to be elastic. By varying the time constant m primary creep (*m<1*), secondary creep (*m=1*), and tertiary creep (*m>1*) can be modeled. This model is described by Whirley and Henshall (1992).

Material Model 116: Composite Layup

This material is for modeling the elastic responses of composite lay-ups that have an arbitrary number of layers through the shell thickness. A pre-integration is used to compute the extensional, bending, and coupling stiffness for use with the Belytschko-Tsay resultant shell formulation. The angles of the local material axes are specified from layer to layer in the *SECTION_SHELL input. This material model must be used with the user defined integration rule for shells, which allows the elastic constants to change from integration point to integration point. Since the stresses are not computed in the resultant formulation, the stress output to the binary databases for the resultant elements are zero.

This material law is based on standard composite lay-up theory. The implementation, [Jones 1975], allows the calculation of the force, N, and moment, M, stress resultants from:

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} \\ A_{21} & A_{22} & A_{26} \\ A_{16} & A_{26} & A_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x^0 \\ \varepsilon_y^0 \\ \varepsilon_z^0 \end{Bmatrix} + \begin{bmatrix} B_{11} & B_{12} & B_{16} \\ B_{21} & B_{22} & B_{26} \\ B_{16} & B_{26} & B_{66} \end{bmatrix} \begin{Bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix}$$

$$\begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \begin{bmatrix} B_{11} & B_{12} & B_{16} \\ B_{21} & B_{22} & B_{26} \\ B_{16} & B_{26} & B_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x^0 \\ \varepsilon_y^0 \\ \varepsilon_z^0 \end{Bmatrix} + \begin{bmatrix} D_{11} & D_{12} & D_{16} \\ D_{21} & D_{22} & D_{26} \\ D_{16} & D_{26} & D_{66} \end{bmatrix} \begin{Bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} \tag{19.116.1}$$

where A_{ij} is the extensional stiffness, D_{ij} is the bending stiffness, and B_{ij} is the coupling stiffness, which is a null matrix for symmetric lay-ups. The mid-surface strains and curvatures are denoted by ε_{ij}^0 and κ_{ij}, respectively. Since these stiffness matrices are symmetric, 18 terms are needed per shell element in addition to the shell resultants, which are integrated in time. This is considerably less storage than would typically be required with through thickness integration which requires a minimum of eight history variables per integration point, e.g., if 100 layers are

used 800 history variables would be stored. Not only is memory much less for this model, but the CPU time required is also considerably reduced.

Material Model 117-118: Composite Matrix

This material is used for modeling the elastic responses of composites where pre-integration, which is done outside of LS-DYNA unlike the lay-up option above, is used to compute the extensional, bending, and coupling stiffness coefficients for use with the Belytschko-Tsay and the assumed strain resultant shell formulations. Since the stresses are not computed in the resultant formulation, the stresses output to the binary databases for the resultant elements are zero.

The calculation of the force, N_{ij}, and moment, M_{ij}, stress resultants is given in terms of the membrane strains, ε_i^0, and shell curvatures, κ_i, as:

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \\ M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} & C_{46} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} & C_{56} \\ C_{61} & C_{62} & C_{63} & C_{64} & C_{65} & C_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x^0 \\ \varepsilon_y^0 \\ \varepsilon_z^0 \\ \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} \tag{19.117.1}$$

where $C_{ij} = C_{ji}$.. In this model this symmetric matrix is transformed into the element local system and the coefficients are stored as element history variables.

In a variation of this model, *MAT_COMPOSITE_DIRECT, the resultants are already assumed to be given in the element local system which reduces the storage since the 21 coefficients are not stored as history variables as part of the element data. The shell thickness is built into the coefficient matrix and, consequently, within the part ID, which references this material ID, the thickness must be uniform.

Material Model 119: General Nonlinear 6DOF Discrete Beam

Catastrophic failure, which is based on displacement resultants, occurs if either of the following inequalities are satisfied:

$$\left(\frac{u_r}{u_r^{tfail}}\right)^2 + \left(\frac{u_s}{u_s^{tfail}}\right)^2 + \left(\frac{u_t}{u_t^{tfail}}\right)^2 + \left(\frac{\theta_r}{\theta_r^{tfail}}\right)^2 + \left(\frac{\theta_s}{\theta_s^{tfail}}\right)^2 + \left(\frac{\theta_t}{\theta_t^{tfail}}\right)^2 - 1. \geq 0$$

$$\left(\frac{u_r}{u_r^{cfail}}\right)^2 + \left(\frac{u_s}{u_s^{cfail}}\right)^2 + \left(\frac{u_t}{u_t^{cfail}}\right)^2 + \left(\frac{\theta_r}{\theta_r^{cfail}}\right)^2 + \left(\frac{\theta_s}{\theta_s^{cfail}}\right)^2 + \left(\frac{\theta_t}{\theta_t^{cfail}}\right)^2 - 1. \geq 0 \tag{19.119.1}$$

After failure the discrete element is deleted. If failure is included either the tension failure or the compression failure or both may be used.

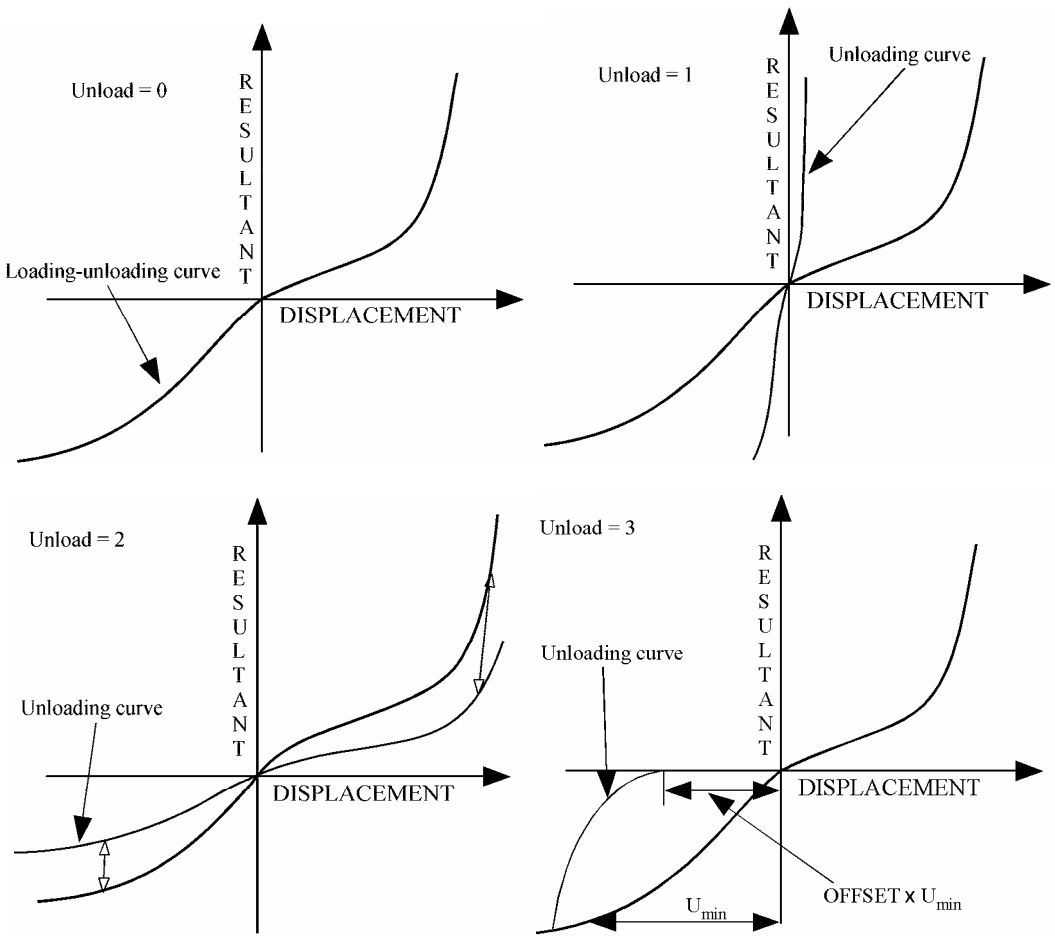


Figure 19.119.1. Load and unloading behavior.

Material Model 120: Gurson

The Gurson flow function is defined as:

$$\Phi = \frac{\sigma_M^2}{\sigma_Y^2} + 2q_1 f^* \cosh\left(\frac{3q_2\sigma_H}{2\sigma_Y}\right) - 1 - \left(q_1 f^*\right)^2 = 0 \qquad (19.120.1)$$

where σ_M is the equivalent von Mises stress, σ_Y is the Yield stress, σ_H is the mean hydrostatic stress. The effective void volume fraction is defined as

$$f^*(f) = \begin{cases} f & f \le f_c \\ f_c + \dfrac{1/q_1 - f_c}{f_F - f_c}(f - f_c) & f > f_c \end{cases} \qquad (19.120.2)$$

The growth of void volume fraction is defined as

$$\dot{f} = \dot{f}_G + \dot{f}_N \qquad (19.120.3)$$

where the growth of existing voids is defined as

$$\dot{f}_G = (1-f)\dot{\varepsilon}_{kk}^p \tag{19.120.4}$$

and the nucleation of new voids is defined as

$$\dot{f}_N = A\dot{\varepsilon}_p \tag{19.120.5}$$

where

$$A = \frac{f_N}{S_N\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left(\frac{\varepsilon_p-\varepsilon_N}{S_N}\right)^2\right) \tag{19.120.6}$$

Material Model 120: Gurson RCDC

The Rc-Dc model is defined as follows

The damage D is given by

$$D = \int \omega_1\omega_2 d\varepsilon^p \tag{19.120.7}$$

where ε^p is the equivalent plastic strain,

$$\omega_1 = \left(\frac{1}{1-\gamma\sigma_m}\right)^\alpha \tag{19.120.8}$$

is the triaxial stress weighting term and

$$\omega_2 = (2-A_D)^\beta \tag{19.120.9}$$

is the asymmetric strain weighting term.

In the above σ_m is the mean stress and

$$A_D = \min\left(\left|\frac{S_2}{S_3}\right|, \left|\frac{S_3}{S_2}\right|\right) \tag{19.120.10}$$

Fracture is initiated when the accumulation of damage satisfies

$$\frac{D}{D_c} > 1 \tag{19.120.11}$$

where D_c is the critical damage given by

$$D_c = D_0\left(1 + b\left|\nabla D\right|^{\lambda}\right) \tag{19.120.12}$$

Material Model 124: Tension-Compression Plasticity

This is an isotropic elastic-plastic material where a unique yield stress versus plastic strain curve can be defined for compression and tension. Failure can occur based on plastic strain or a minimum time step size. Rate effects are modeled by using the Cowper-Symonds strain rate model.

The stress-strain behavior follows one curve in compression and another in tension. The sign of the mean stress determines the state where a positive mean stress (i.e., a negative pressure) is indicative of tension. Two load curves, $f_t(p)$ and $f_c(p)$, are defined, which give the yield stress, σ_y, versus effective plastic strain for both the tension and compression regimes. The two pressure values, p_t and p_c, when exceeded, determine if the tension curve or the compressive curve is followed, respectively. If the pressure, p, falls between these two values, a weighted average of the two curves are used:

$$if \quad -p_t \le p \le p_c \quad \begin{cases} scale = \dfrac{p_c - p}{p_c + p_t} \\ \sigma_y = scale \cdot f_t(p) + (1 - scale) \cdot f_c(p) \end{cases}$$

Strain rate is accounted for using the Cowper and Symonds model, which scales the yield stress with the factor

$$1 + \left(\frac{\dot{\varepsilon}}{C}\right)^{1/p} \tag{19.124.1}$$

where $\dot{\varepsilon}$ is the strain rate. $\dot{\varepsilon} = \sqrt{\dot{\varepsilon}_{ij}\dot{\varepsilon}_{ij}}$

Material Model 126: Metallic Honeycomb

For efficiency it is strongly recommended that the load curve ID's: LCA, LCB, LCC, LCS, LCAB, LCBC, and LCCA, contain exactly the same number of points with corresponding strain values on the abscissa. If this recommendation is followed the cost of the table lookup is insignificant. Conversely, the cost increases significantly if the abscissa strain values are not consistent between load curves.

The behavior before compaction is orthotropic where the components of the stress tensor are uncoupled, i.e., a component of strain will generate resistance in the local *a*- direction with no coupling to the local *b* and *c* directions. The elastic modulii vary from their initial values to the fully compacted values linearly with the relative volume:

$$E_{aa} = E_{aau} + \beta_{aa}(E - E_{aau}) \qquad G_{ab} = G_{abu} + \beta(G - G_{abu})$$

$$E_{bb} = E_{bbu} + \beta_{bb}(E - E_{bbu}) \qquad G_{bc} = G_{bcu} + \beta(G - G_{bcu}) \qquad (19.126.1)$$

$$E_{cc} = E_{ccu} + \beta_{cc}(E - E_{ccu}) \qquad G_{ca} = G_{cau} + \beta(G - G_{cau})$$

where

$$\beta = \max\left[\min\left(\tfrac{1-V}{1-V_f}, 1\right), 0\right] \qquad (19.126.2)$$

and G is the elastic shear modulus for the fully compacted honeycomb material

$$G = \frac{E}{2(1+\nu)}. \qquad (19.126.3)$$

The relative volume, ν, is defined as the ratio of the current volume over the initial volume, and typically, $\nu = 1$ at the beginning of a calculation.

The load curves define the magnitude of the stress as the material undergoes deformation. The first value in the curve should be less than or equal to zero corresponding to tension and increase to full compaction. **Care should be taken when defining the curves so the extrapolated values do not lead to negative yield stresses.**

At the beginning of the stress update we transform each element's stresses and strain rates into the local element coordinate system. For the uncompacted material, the trial stress components are updated using the elastic interpolated modulii according to:

$$\sigma_{aa}^{n+1^{trial}} = \sigma_{aa}^{n} + E_{aa}\Delta\varepsilon_{aa} \qquad \sigma_{ab}^{n+1^{trial}} = \sigma_{ab}^{n} + 2G_{ab}\Delta\varepsilon_{ab}$$

$$\sigma_{bb}^{n+1^{trial}} = \sigma_{bb}^{n} + E_{bb}\Delta\varepsilon_{bb} \qquad \sigma_{bc}^{n+1^{trial}} = \sigma_{bc}^{n} + 2G_{bc}\Delta\varepsilon_{bc} \qquad (19.126.4)$$

$$\sigma_{cc}^{n+1^{trial}} = \sigma_{cc}^{n} + E_{cc}\Delta\varepsilon_{cc} \qquad \sigma_{ca}^{n+1^{trial}} = \sigma_{ca}^{n} + 2G_{ca}\Delta\varepsilon_{ca}$$

We then independently check each component of the updated stresses to ensure that they do not exceed the permissible values determined from the load curves, e.g., if

$$\left|\sigma_{ij}^{n+1^{trial}}\right| > \lambda\sigma_{ij}(\varepsilon_{ij}) \qquad (19.126.5)$$

then

$$\sigma_{ij}^{n+1} = \sigma_{ij}(\varepsilon_{ij})\frac{\lambda\sigma_{ij}^{n+1^{trial}}}{\left|\sigma_{ij}^{n+1^{trial}}\right|} \qquad (19.126.6)$$

The components of $\sigma_{ij}(\varepsilon_{ij})$ are defined by load curves. The parameter λ is either unity or a value taken from the load curve number, LCSR, that defines λ as a function of strain-rate. Strain-rate is defined here as the Euclidean norm of the deviatoric strain-rate tensor.

For fully compacted material we assume that the material behavior is elastic-perfectly plastic and updated the stress components according to:

$$s_{ij}^{trial} = s_{ij}^{n} + 2G\Delta\varepsilon_{ij}^{dev^{n+\frac{1}{2}}} \tag{19.126.7}$$

where the deviatoric strain increment is defined as

$$\Delta\varepsilon_{ij}^{dev} = \Delta\varepsilon_{ij} - \tfrac{1}{3}\Delta\varepsilon_{kk}\delta_{ij} \ . \tag{19.126.8}$$

We now check to see if the yield stress for the fully compacted material is exceeded by comparing

$$s_{eff}^{trial} = \left(\tfrac{3}{2}s_{ij}^{trial}s_{ij}^{trial}\right)^{\frac{1}{2}} \tag{19.126.9}$$

the effective trial stress to the yield stress, σ_y. If the effective trial stress exceeds the yield stress, we simply scale back the stress components to the yield surface

$$s_{ij}^{n+1} = \frac{\sigma_y}{s_{eff}^{trial}} s_{ij}^{trial} \ . \tag{19.126.10}$$

We can now update the pressure using the elastic bulk modulus, K

$$p^{n+1} = p^{n} - K\Delta\varepsilon_{kk}^{n+\frac{1}{2}} \tag{19.126.11}$$

$$K = \frac{E}{3(1-2\nu)}$$

and obtain the final value for the Cauchy stress

$$\sigma_{ij}^{n+1} = s_{ij}^{n+1} - p^{n+1}\delta_{ij} \ . \tag{19.126.12}$$

After completing the stress update we transform the stresses back to the global configuration.

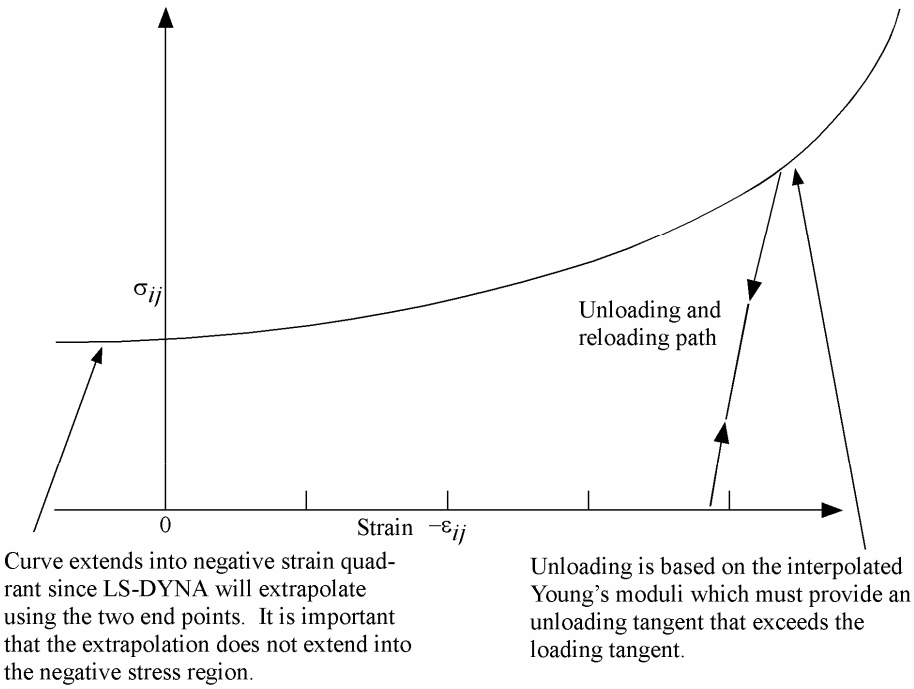


Figure 19.126.1. Stress quantity versus strain. Note that the "yield stress" at a strain of zero is nonzero. In the load curve definition the "time" value is the directional strain and the "function" value is the yield stress.

19.126.1 Stress Update

If LCA<0, a transversely anisotropic yield surface is obtained where the uniaxial limit stress, $\sigma^y\left(\varphi,\varepsilon^{vol}\right)$, can be defined as a function of angle φ with the strong axis and volumetric strain, ε^{vol}. Elastically, the new material model is assumed to behave exactly as material 126 (with the restriction $E_{22u}=E_{33u}$ and $G_{12u}=G_{13u}$), see the LS-DYNA Keyword User's Manual [Hallquist 2003]. As for the plastic behavior, a natural question that arises is how to define the limit stress for a general multiaxial stress state that reduces to the uniaxial limit stress requirement when the stress is uniaxial. Having given it some thought, we feel that it is most convenient to work with the principal stresses and the corresponding directions to achieve this goal.

Assume that the elastic update results in a trial stress $\boldsymbol{\sigma}^{\text{trial}}$ in the material coordinate system. This stress tensor is diagonalized to obtain the principal stresses σ_1^{trial}, σ_2^{trial} and σ_3^{trial} and the corresponding principal directions $\mathbf{q}_1$, $\mathbf{q}_2$ and $\mathbf{q}_3$ relative to the material coordinate system. The angle that each direction makes with the strong axis of anisotropy $\mathbf{e}_1$ is given by $\varphi_i=\arccos\left|\mathbf{q}_i\cdot\mathbf{e}_1\right|$, $i=1,2,3$.

Now a limit stress in the direction of a general multiaxial stress is determined as a convex combination of the uniaxial limit stress in each principal direction

$$\sigma^Y(\boldsymbol{\sigma}^{\text{trial}}) = \frac{\sum_{j=1}^{3} \sigma^Y(\varphi_j, \varepsilon^{\text{vol}})\sigma_j^{\text{trial}}\sigma_j^{\text{trial}}}{\sum_{k=1}^{3} \sigma_k^{\text{trial}}\sigma_k^{\text{trial}}}. \quad (19.126.13)$$

Each of the principal stresses is updated as

$$\sigma_i = \sigma_i^{\text{trial}} \min\left(1, \frac{\sigma^Y(\boldsymbol{\sigma}^{\text{trial}})}{\sqrt{\sum_{k=1}^{3} \sigma_k^{\text{trial}}\sigma_k^{\text{trial}}}}\right) \quad (19.126.14)$$

and the new stress is transformed back to the material coordinate system[1].

This stress update is not uniquely defined when the stress tensor possesses multiple eigenvalues, thus the following simple set of rules is applied. If all principal stresses are equal, one of the principal directions is chosen to coincide with the strong axis of anisotropy. If two principal stresses are equal, then one of the directions corresponding to this stress value is chosen perpendicular to the strong axis of anisotropy.

19.126.2 Support for Independent Shear and Hydrostatic Yield Stress Limit

The model just described turned out to be weak in shear [Okuda 2003] and there were no means of adding shear resistance without changing the behavior in pure uniaxial compression. We propose the following modification of the model where the user can prescribe the shear and hydrostatic resistance in the material without changing the uniaxial behavior.

Assume that the elastic update results in a trial stress $\boldsymbol{\sigma}^{\text{trial}}$ in the material coordinate system. This stress tensor is diagonalized to obtain the principal stresses σ_1^{trial}, σ_2^{trial} and σ_3^{trial} and the corresponding principal directions $\mathbf{q}_1$, $\mathbf{q}_2$ and $\mathbf{q}_3$ relative to the material coordinate system. For this discussion we assume that the principal stress values are ordered so that $\sigma_1^{\text{trial}} \le \sigma_2^{\text{trial}} \le \sigma_3^{\text{trial}}$. Two cases need to be treated.

If $\sigma_1^{\text{trial}} \le -\sigma_3^{\text{trial}}$ then the principal stress value with largest magnitude is σ_1^{trial}, and consequently $\sigma_1^{\text{trial}} \le 0$. Let

$$\sigma_u = \sigma_1^{\text{trial}} + \max(|\sigma_2^{\text{trial}}|, |\sigma_3^{\text{trial}}|) \quad (19.126.15)$$

$$\sigma_p = \frac{1}{3}\left\{-\max(|\sigma_2^{\text{trial}}|, |\sigma_3^{\text{trial}}|) + \sigma_2^{\text{trial}} + \sigma_3^{\text{trial}}\right\} \quad (19.126.16)$$

and finally

[1] Since each component of the stress tensor is scaled by the same factor in Equation 19.126.14, the stress is in practice not transformed back but the scaling is performed on the stress in the material coordinate system.

$$\begin{aligned}\sigma_d^1 &= -\max(\left|\sigma_2^{\text{trial}}\right|,\left|\sigma_3^{\text{trial}}\right|) - \sigma_p \\ \sigma_d^2 &= \sigma_2^{\text{trial}} - \sigma_p \\ \sigma_d^3 &= \sigma_3^{\text{trial}} - \sigma_p\end{aligned} \tag{19.126.17}$$

The total stress is the sum of a uniaxial stress represented by σ_u, a hydrostatic stress represented by σ_p and a deviatoric stress represented by σ_d^1, σ_d^2 and σ_d^3. The angle that the direction of σ_u makes with the strong axis of anisotropy $\mathbf{e}_1$ is given by $\varphi = \arccos\left|\mathbf{q}_1 \cdot \mathbf{e}_1\right|$.

Now a limit stress for the general multiaxial stress is determined as a convex combination of the three stress contributions as follows

$$\sigma^Y(\boldsymbol{\sigma}^{\text{trial}}) = \frac{\sigma_u^Y(\varphi,\varepsilon^{\text{vol}})\sigma_u^2 + 3\sqrt{3}\sigma_p^Y(\varepsilon^{\text{vol}})\sigma_p^2 + \sqrt{2}\sigma_d^Y(\varepsilon^{\text{vol}})\left\{(\sigma_d^1)^2 + (\sigma_d^2)^2 + (\sigma_d^3)^2\right\}}{\sigma_u^2 + 3\sigma_p^2 + (\sigma_d^1)^2 + (\sigma_d^2)^2 + (\sigma_d^3)^2}.$$

Here $\sigma_u^Y(\varphi,\varepsilon^{\text{vol}})$ is the prescribed uniaxial stress limit, $\sigma_p^Y(\varepsilon^{\text{vol}})$ is the hydrostatic stress limit and $\sigma_d^Y(\varepsilon^{\text{vol}})$ is the stress limit in simple shear. The input for the first of these is exactly as for the old model. The other two functions are for now written

$$\begin{aligned}\sigma_p^Y(\varepsilon^{\text{vol}}) &= \sigma_p^Y + \sigma^S(\varepsilon^{\text{vol}}) \\ \sigma_d^Y(\varepsilon^{\text{vol}}) &= \sigma_d^Y + \sigma^S(\varepsilon^{\text{vol}})\end{aligned} \tag{19.126.18}$$

where σ_p^Y and σ_d^Y are user prescribed constant stress limits and σ^S is the function describing the densification of the material when loaded in the direction of the strong material axis. We use the keyword parameters `ECCU` and `GCAU` for the new input as follows.

`ECCU` σ_d^Y, average stress limit (yield) in simple shear.
`GCAU` σ_p^Y, average stress limit (yield) in hydrostatic compression (pressure).

Both of these parameters should be positive.

Each of the principal stresses is updated as

$$\sigma_i = \sigma_i^{\text{trial}} \min\left(1, \frac{\sigma^Y(\boldsymbol{\sigma}^{\text{trial}})}{\sqrt{\sum_{k=1}^{3}\sigma_k^{\text{trial}}\sigma_k^{\text{trial}}}}\right) \tag{19.126.19}$$

and the new stress is transformed back to the material coordinate system.

If $\sigma_3^{\text{trial}} \geq -\sigma_1^{\text{trial}}$ then the principal stress value with largest magnitude is σ_3^{trial} and consequently $\sigma_3^{\text{trial}} \geq 0$. Let

$$\sigma_u = \sigma_3^{\text{trial}} - \max(\left|\sigma_2^{\text{trial}}\right|, \left|\sigma_1^{\text{trial}}\right|) \tag{19.126.20}$$

$$\sigma_p = \frac{1}{3}\left\{\max(\left|\sigma_2^{\text{trial}}\right|, \left|\sigma_1^{\text{trial}}\right|) + \sigma_2^{\text{trial}} + \sigma_1^{\text{trial}}\right\} \tag{19.126.21}$$

and finally

$$\begin{aligned} \sigma_d^1 &= \sigma_1^{\text{trial}} - \sigma_p \\ \sigma_d^2 &= \sigma_2^{\text{trial}} - \sigma_p \\ \sigma_d^3 &= \max(\left|\sigma_2^{\text{trial}}\right|, \left|\sigma_1^{\text{trial}}\right|) - \sigma_p \end{aligned} \tag{19.126.22}$$

The angle that the direction of σ_u makes with the strong axis of anisotropy $\mathbf{e}_1$ is given by $\varphi = \arccos\left|\mathbf{q}_3 \cdot \mathbf{e}_1\right|$. The rest of the treatment is the same as for the case when $\sigma_1^{\text{trial}} \leq -\sigma_3^{\text{trial}}$. To motivate the model, let us consider three states of stress.

1. For a uniaxial stress σ, we have $\sigma_u = \sigma$ and $\sigma_p = \sigma_d^1 = \sigma_d^2 = \sigma_d^3 = 0$. This leads us to $\sigma^Y(\boldsymbol{\sigma}^{\text{trial}}) = \sigma_u^Y(\varphi, \varepsilon^{\text{vol}})$ and hence the stress level will be limited by the user prescribed uniaxial stress limit.
2. For a simple shear σ, we have $\sigma_d^1 = -\sigma_d^3 = -\sigma$ and $\sigma_p = \sigma_d^2 = \sigma_u = 0$. Hence $\sigma^Y(\boldsymbol{\sigma}^{\text{trial}}) = \sqrt{2}\sigma_d^Y(\varepsilon^{\text{vol}})$ and the stress level will be limited by the user prescribed shear stress limit.
3. For a pressure σ, we have $\sigma_p = \sigma$ and $\sigma_u = \sigma_d^1 = \sigma_d^2 = \sigma_d^3 = 0$. Hence $\sigma^Y(\boldsymbol{\sigma}^{\text{trial}}) = \sqrt{3}\sigma_p^Y(\varepsilon^{\text{vol}})$ and the stress level will be limited by the user prescribed hydrostatic stress limit.

Material Model 127: Arruda-Boyce Hyperviscoelastic Rubber

This material model, described in the paper by Arruda and Boyce [1993], provides a rubber model that is optionally combined with linear viscoelasticity. Rubber is generally considered to be fully incompressible since the bulk modulus greatly exceeds the shear modulus in magnitude; therefore, to model the rubber as an unconstrained material, a hydrostatic work term, $W_H(J)$, is included in the strain energy functional which is a function of the relative volume, J, [Ogden, 1984]:

$$
\begin{aligned}
W(J_1,J_2,J) &= nk\theta\left[\tfrac{1}{2}(J_1-3)+\tfrac{1}{20N}(J_1^2-9)+\tfrac{11}{1050N^2}(J_1^3-27)\right] \\
&+nk\theta\left[\tfrac{19}{7000N^3}(J_1^4-81)+\tfrac{519}{673750N^4}(J_1^5-243)\right]+W_H(J) \\
J_1 &= I_1 J^{-1/3} \\
J_2 &= I_2 J
\end{aligned}
\tag{19.127.1}
$$

The hydrostatic work term is expressed in terms of the bulk modulus, K, and J, as:

$$
W_H(J) = \frac{K}{2}(J-1)^2 \tag{19.127.2}
$$

Rate effects are taken into account through linear viscoelasticity by a convolution integral of the form:

$$
\sigma_{ij} = \int_0^t g_{ijkl}(t-\tau)\frac{\partial \varepsilon_{kl}}{\partial \tau}\,d\tau \tag{19.127.3}
$$

or in terms of the second Piola-Kirchhoff stress, S_{ij}, and Green's strain tensor, E_{ij},

$$
S_{ij} = \int_0^t G_{ijkl}(t-\tau)\frac{\partial E_{kl}}{\partial \tau}\,d\tau \tag{19.127.4}
$$

where $g_{ijkl}(t-\tau)$ and $G_{ijkl}(t-\tau)$ are the relaxation functions for the different stress measures. This stress is added to the stress tensor determined from the strain energy functional.

If we wish to include only simple rate effects, the relaxation function is represented by six terms from the Prony series:

$$
g(t) = \alpha_0 + \sum_{m=1}^{N} \alpha_m e^{-\beta t} \tag{19.127.5}
$$

given by,

$$
g(t) = \sum_{i=1}^{n} G_i e^{-\beta_i t} \tag{19.127.6}
$$

This model is effectively a Maxwell fluid which consists of a dampers and springs in series. We characterize this in the input by shear modulii, G_i, and decay constants, β_i. The viscoelastic behavior is optional and an arbitrary number of terms may be used.

Material Model 128: Heart Tissue

This material model provides a tissue model described in the paper by Guccione, McCulloch, and Waldman [1991]

The tissue model is described in terms of the energy functional in terms of the Green strain components, E_{ij},

$$W(E,J) = \tfrac{C}{2}(e^Q - 1) + W_H(J)$$
$$Q = b_1 E_{11}^2 + b_2(E_{22}^2 + E_{33}^2 + E_{23}^2 + E_{32}^2) + b_3(E_{12}^2 + E_{21}^2 + E_{13}^2 + E_{31}^2) \quad (19.128.1)$$

where the hydrostatic work term is in terms of the bulk modulus, K, and the third invariant J, as:

$$W_H(J) = \frac{K}{2}(J-1)^2 \quad (19.128.2)$$

The Green components are modified to eliminate any effects of volumetric work following the procedures of Ogden.

Material Model 129: Isotropic Lung Tissue

This material model provides a lung tissue model described in the paper by Vawter [1980].

The material is described by a strain energy functional expressed in terms of the invariants of the Green Strain:

$$W(I_1,I_2) = \frac{C}{2\Delta}e^{(\alpha I_1^2 + \beta I_2)} + \frac{12C_1}{\Delta(1+C_2)}\left[A^{(1+C_2)} - 1\right]$$
$$A^2 = \tfrac{4}{3}(I_1 + I_2) - 1 \quad (19.129.1)$$

where the hydrostatic work term is in terms of the bulk modulus, K, and the third invariant J, as:

$$W_H(J) = \frac{K}{2}(J-1)^2 \quad (19.129.2)$$

Rate effects are taken into account through linear viscoelasticity by a convolution integral of the form:

$$\sigma_{ij} = \int_0^t g_{ijkl}(t-\tau)\frac{\partial \varepsilon_{kl}}{\partial \tau}d\tau \quad (19.129.3)$$

or in terms of the second Piola-Kirchhoff stress, S_{ij}, and Green's strain tensor, E_{ij},

$$S_{ij} = \int_0^t G_{ijkl}\left(t-\tau\right)\frac{\partial E_{kl}}{\partial \tau}d\tau \tag{19.129.4}$$

where $g_{ijkl}(t-\tau)$ and $G_{ijkl}\left(t-\tau\right)$ are the relaxation functions for the different stress measures. This stress is added to the stress tensor determined from the strain energy functional.

If we wish to include only simple rate effects, the relaxation function is represented by six terms from the Prony series:

$$g(t) = \alpha_0 + \sum_{m=1}^{N}\alpha_m e^{-\beta t} \tag{19.129.5}$$

given by,

$$g(t) = \sum_{i=1}^{n} G_i e^{-\beta_i t} \tag{19.129.6}$$

This model is effectively a Maxwell fluid which consists of a dampers and springs in series. We characterize this in the input by shear modulii, G_i, and decay constants, β_i. The viscoelastic behavior is optional and an arbitrary number of terms may be used.

Material Model 130: Special Orthotropic

The in-plane elastic matrix for in-plane plane stress behavior is given by:

$$C_{in\,plane} = \begin{bmatrix} Q_{11p} & Q_{12p} & 0 & 0 & 0 \\ Q_{12p} & Q_{22p} & 0 & 0 & 0 \\ 0 & 0 & Q_{44p} & 0 & 0 \\ 0 & 0 & 0 & Q_{55p} & 0 \\ 0 & 0 & 0 & 0 & Q_{66p} \end{bmatrix} \tag{19.130.1}$$

where the terms Q_{ijp} are defined as:

$$\begin{aligned} Q_{11p} &= \frac{E_{11p}}{1-\nu_{12p}\nu_{21p}} \\ Q_{22p} &= \frac{E_{22p}}{1-\nu_{12p}\nu_{21p}} \\ Q_{12p} &= \frac{\nu_{12p}E_{11p}}{1-\nu_{12p}\nu_{21p}} \\ Q_{44p} &= G_{12p} \\ Q_{55p} &= G_{23p} \\ Q_{66p} &= G_{31p} \end{aligned} \tag{19.130.2}$$

The elastic matrix for bending behavior is given by:

$$C_{bending} = \begin{bmatrix} Q_{11b} & Q_{12b} & 0 \\ Q_{12b} & Q_{22b} & 0 \\ 0 & 0 & Q_{44b} \end{bmatrix} \tag{19.130.3}$$

where the terms Q_{ijb} are similarly defined.

Material Model 131: Isotropic Smeared Crack

The following documentation is taken nearly verbatim from the documentation of that by Lemmen and Meijer [2001].

Three methods are offered to model progressive failure. The maximum principal stress criterion detects failure if the maximum (most tensile) principal stress exceeds σ_{max}. Upon failure, the material can no longer carry stress.

The second failure model is the smeared crack model with linear softening stress-strain curve using equivalent uniaxial strains. Failure is assumed to be perpendicular to the principal strain directions. A rotational crack concept is employed in which the crack directions are related to the current directions of principal strain. Therefore crack directions may rotate in time. Principal stresses are expressed as

$$\begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{pmatrix} = \begin{bmatrix} \bar{\mathrm{E}}_1 & 0 & 0 \\ 0 & \bar{\mathrm{E}}_2 & 0 \\ 0 & 0 & \bar{\mathrm{E}}_3 \end{bmatrix} \begin{pmatrix} \tilde{\varepsilon}_1 \\ \tilde{\varepsilon}_2 \\ \tilde{\varepsilon}_3 \end{pmatrix} = \begin{pmatrix} \bar{\mathrm{E}}_1 \tilde{\varepsilon}_1 \\ \bar{\mathrm{E}}_2 \tilde{\varepsilon}_2 \\ \bar{\mathrm{E}}_3 \tilde{\varepsilon}_3 \end{pmatrix} \tag{19.131.1}$$

with $\bar{\mathrm{E}}_1$, $\bar{\mathrm{E}}_2$ and $\bar{\mathrm{E}}_3$ being secant stiffness in the terms that depend on internal variables.

In the model developed for DYCOSS it has been assumed that there is no interaction between the three directions in which case stresses simply follow

$$\sigma_j(\tilde{\varepsilon}_j) = \begin{cases} \mathrm{E}\tilde{\varepsilon}_j & if \quad 0 \le \tilde{\varepsilon}_j \le \tilde{\varepsilon}_{j,ini} \\ \left(1 - \dfrac{\tilde{\varepsilon}_j - \tilde{\varepsilon}_{j,ini}}{\tilde{\varepsilon}_{j,ult} - \tilde{\varepsilon}_{j,ini}}\right)\bar{\sigma} & if \quad \tilde{\varepsilon}_{j,ini} < \tilde{\varepsilon}_j \le \tilde{\varepsilon}_{j,ult} \\ 0 & if \quad \tilde{\varepsilon}_j > \tilde{\varepsilon}_{j,ult} \end{cases} \tag{19.131.2}$$

with $\bar{\sigma}$ the ultimate stress, $\tilde{\varepsilon}_{j,ini}$ the damage threshold, and $\tilde{\varepsilon}_{j,ult}$ the ultimate strain in *j*-direction. The damage threshold is defined as

$$\tilde{\varepsilon}_{j,ini} = \frac{\bar{\sigma}}{\mathrm{E}} \tag{19.131.3}$$

The ultimate strain is obtained by relating the crack growth energy and the dissipated energy

$$\int\int \bar{\sigma}\, d\tilde{\varepsilon}_{j,ult}\, dV = GA \tag{19.131.4}$$

with G the energy release rate, V the element volume and A the area perpendicular to the principal strain direction. The one-point elements in LS-DYNA have a single integration point and the integral over the volume may be replaced by the volume. For linear softening it follows

$$\tilde{\varepsilon}_{j,ult} = \frac{2GA}{V\bar{\sigma}} \tag{19.131.5}$$

The above formulation may be regarded as a damage equivalent to the maximum principle stress criterion.

The third model is a damage model presented by Brekelmans et. al. [1991]. Here the Cauchy stress tensor σ is expressed as

$$\sigma = (1-D)\mathrm{E}\varepsilon \tag{19.131.6}$$

where D represents the current damage and the factor (1-D) is the reduction factor caused by damage. The scalar damage variable is expressed as a function of a so-called damage equivalent strain ε_d

$$D = D(\varepsilon_d) = 1 - \frac{\varepsilon_{ini}(\varepsilon_{ult} - \varepsilon_d)}{\varepsilon_d(\varepsilon_{ult} - \varepsilon_{ini})} \tag{19.131.7}$$

with ultimate and initial strains as defined by

$$\varepsilon_d = \frac{k-1}{2k(1-2v)} J_1 + \frac{1}{2k}\sqrt{\left(\frac{k-1}{1-2v} J_1\right)^2 + \frac{6k}{(1+v)^2} J_2} \tag{19.131.8}$$

where the constant k represents the ratio of the strength in tension over the strength in compression

$$k = \frac{\sigma_{ult,\,tension}}{\sigma_{ult,\,compression}} \tag{19.131.9}$$

J_1 and J_2 are the first and the second invariants of the strain tensor representing the volumetric and the deviatoric straining, respectively

$$\begin{aligned} J_1 &= \varepsilon_{ij} \\ J_2 &= \varepsilon'_{ij}\varepsilon'_{ij} \end{aligned} \tag{19.131.10}$$

where ε'_{ij} denotes the deviatoric part of the strain tensor.

If the compression and tension strength are equal the dependency on the volumetric strain vanishes in Equation (19.131.8) and failure is shear dominated. If the compressive strength is

much larger than the strength in tension, k becomes large and the J_1 terms in Equation (19.131.8) dominate the behavior.

Material Model 133: Barlat_YLD2000

The yield condition for this material can be written

$$f(\boldsymbol{\sigma},\boldsymbol{\alpha},\varepsilon_p) = \sigma_{\text{eff}}(\sigma_{xx}-2\alpha_{xx}-\alpha_{yy},\sigma_{yy}-2\alpha_{yy}-\alpha_{xx},\sigma_{xy}-\alpha_{xy}) - \sigma_Y^t(\varepsilon_p,\dot{\varepsilon}_p,\beta) \le 0 \tag{19.133.1}$$

where

$$\begin{aligned}\sigma_{\text{eff}}(s_{xx},s_{yy},s_{xy}) &= \left(\frac{1}{2}(\phi'+\phi'')\right)^{1/a}\\ \phi' &= \left|X'_1 - X'_2\right|^a \\ \phi'' &= \left|2X''_1 + X''_2\right|^a + \left|X''_1 + 2X''_2\right|^a\end{aligned} \tag{19.133.2}$$

The X'_i and X''_i are the 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

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

where

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

where α_1 to α_8 are the parameters that determine the shape of the yield surface.

The yield stress is expressed as

$$\sigma_Y^t(\varepsilon_p, \dot{\varepsilon}_p, \beta) = \sigma_Y^v(\varepsilon_p, \dot{\varepsilon}_p) + \beta(\sigma_0 - \sigma_Y^v(\varepsilon_p, \dot{\varepsilon}_p)) \qquad (19.133.3)$$

where β determines the fraction kinematic hardening and σ_0 is the initial yield stress. The yield stress for purely isotropic hardening is given by

$$\sigma_Y^v(\varepsilon_p, \dot{\varepsilon}_p) = \sigma_Y(\varepsilon_p)\left(1 + \left\{\frac{\dot{\varepsilon}_p}{C}\right\}^{1/p}\right) \qquad (19.133.4)$$

where C and p are the Cowper-Symonds material parameters for strain rate effects.

The evolution of back stress is given by

$$\dot{\boldsymbol{\alpha}} = \dot{\lambda}\beta \frac{\left(\frac{\partial \sigma_Y}{\partial \varepsilon_p}\left(1 + \left\{\frac{\dot{\varepsilon}_p}{C}\right\}^{1/p}\right) + \sigma_Y \frac{1}{pC\Delta t}\left\{\frac{\dot{\varepsilon}_p}{C}\right\}^{1/p-1}\right)}{\frac{\partial \sigma_{\text{eff}}}{\partial \mathbf{s}} \cdot \frac{\partial \sigma_{\text{eff}}}{\partial \mathbf{s}}} \frac{\partial \sigma_{\text{eff}}}{\partial \mathbf{s}} \qquad (19.133.5)$$

where Δt is the current time step size and $\dot{\lambda}$ is the rate of plastic strain multiplier.

For the plastic return algorithms, the gradient of effective stress is computed as

$$\begin{pmatrix} \frac{\partial \sigma_{\text{eff}}}{\partial s_{xx}} \\ \frac{\partial \sigma_{\text{eff}}}{\partial s_{yy}} \\ \frac{\partial \sigma_{\text{eff}}}{\partial s_{xy}} \end{pmatrix} = \frac{a\sigma_{\text{eff}}^{a-1}}{2}\begin{pmatrix} L'_{11} & L'_{21} & 0 \\ L'_{12} & L'_{22} & 0 \\ 0 & 0 & L'_{33} \end{pmatrix}\begin{pmatrix} \frac{\partial \phi'}{\partial X'_{11}} \\ \frac{\partial \phi'}{\partial X'_{22}} \\ \frac{\partial \phi'}{\partial X'_{12}} \end{pmatrix} + \frac{a\sigma_{\text{eff}}^{a-1}}{2}\begin{pmatrix} L''_{11} & L''_{21} & 0 \\ L''_{12} & L''_{22} & 0 \\ 0 & 0 & L''_{33} \end{pmatrix}\begin{pmatrix} \frac{\partial \phi''}{\partial X''_{11}} \\ \frac{\partial \phi''}{\partial X''_{22}} \\ \frac{\partial \phi''}{\partial X''_{12}} \end{pmatrix}$$

with the aid of

$$\frac{\partial \phi'}{\partial X'_{ij}} = a(X'_1 - X'_2)^{a-1} \frac{\partial (X'_1 - X'_2)}{\partial X'_{ij}}$$

$$\frac{\partial \phi''}{\partial X''_{ij}} = a\left|2X''_1 + X''_2\right|^{a-1} \operatorname{sgn}(2X''_1 + X''_2) \frac{\partial (2X''_1 + X''_2)}{\partial X''_{ij}} + a\left|2X''_2 + X''_1\right|^{a-1} \operatorname{sgn}(2X''_2 + X''_1) \frac{\partial (2X''_2 + X''_1)}{\partial X''_{ij}}$$

and

$$\frac{\partial (X'_1 - X'_2)}{\partial X'_{11}} = \frac{X'_{11} - X'_{22}}{\sqrt{(X'_{11} - X'_{22})^2 + 4X'^2_{12}}}$$

$$\frac{\partial (X'_1 - X'_2)}{\partial X'_{22}} = \frac{X'_{22} - X'_{11}}{\sqrt{(X'_{11} - X'_{22})^2 + 4X'^2_{12}}}$$

$$\frac{\partial (X'_1 - X'_2)}{\partial X'_{12}} = \frac{4X'_{12}}{\sqrt{(X'_{11} - X'_{22})^2 + 4X'^2_{12}}}$$

$$\frac{\partial (2X''_1 + X''_2)}{\partial X''_{11}} = \frac{3}{2} + \frac{1}{2} \frac{X''_{11} - X''_{22}}{\sqrt{(X''_{11} - X''_{22})^2 + 4X''^2_{12}}}$$

$$\frac{\partial (2X''_1 + X''_2)}{\partial X''_{22}} = \frac{3}{2} + \frac{1}{2} \frac{X''_{22} - X''_{11}}{\sqrt{(X''_{11} - X''_{22})^2 + 4X''^2_{12}}}$$

$$\frac{\partial (2X''_1 + X''_2)}{\partial X''_{12}} = \frac{2X''_{12}}{\sqrt{(X''_{11} - X''_{22})^2 + 4X''^2_{12}}}$$

$$\frac{\partial (2X''_2 + X''_1)}{\partial X''_{11}} = \frac{3}{2} - \frac{1}{2} \frac{X''_{11} - X''_{22}}{\sqrt{(X''_{11} - X''_{22})^2 + 4X''^2_{12}}}$$

$$\frac{\partial (2X''_2 + X''_1)}{\partial X''_{22}} = \frac{3}{2} - \frac{1}{2} \frac{X''_{22} - X''_{11}}{\sqrt{(X''_{11} - X''_{22})^2 + 4X''^2_{12}}}$$

$$\frac{\partial (2X''_2 + X''_1)}{\partial X''_{12}} = -\frac{2X''_{12}}{\sqrt{(X''_{11} - X''_{22})^2 + 4X''^2_{12}}}$$

The algorithm for the plane stress update as well as the formula for the tangent modulus is given in detail in Section 19.36.1 and is not repeated here.

19.133.1 Closest point projection algorithm

This section describes shortly the closest point projection algorithm that was implemented to improve accuracy, hence the implicit performance, of the model. The closest point projection comes down to solving the following system of equations

$$\mathbf{f}_1 = \mathbf{t} + \mathbf{A}\boldsymbol{\alpha} + (\sigma_Y^t(\Delta\lambda) - \sigma_Y^t(0))\mathbf{h} - \boldsymbol{\sigma}^{\text{trial}}(\Delta\varepsilon_{33}) + 2G\Delta\lambda\mathbf{D}\nabla\sigma_{\text{eff}}(\mathbf{t}) = \mathbf{0}$$

$$f_2 = -\sigma_{\text{eff}}(\mathbf{t}) + \sigma_Y^t(\Delta\lambda) = 0$$

$$f_3 = \sigma_{33}^{\text{trial}}(\Delta\varepsilon_{33}) + 2G\Delta\lambda(\nabla\sigma_{\text{eff}}^1(\mathbf{t}) + \nabla\sigma_{\text{eff}}^2(\mathbf{t})) = 0$$

where

$$\mathbf{h} = \frac{\beta}{\nabla\sigma_{\text{eff}}(\mathbf{t})^T\mathbf{B}\nabla\sigma_{\text{eff}}(\mathbf{t})}\mathbf{B}\nabla\sigma_{\text{eff}}(\mathbf{t})$$

in terms of the unknown variables $\mathbf{t}$ (stress), $\Delta\varepsilon_{33}$ (thickness strain increment) and $\Delta\lambda$ (plastic strain increment). In the above

$$\mathbf{D} = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 0.5 \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} 2 & 1 & \\ 1 & 2 & \\ & & 1 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 2 & 1 & \\ 1 & 2 & \\ & & 0.5 \end{bmatrix} \quad H = \frac{\partial\sigma_Y^t}{\partial\varepsilon_p}.$$

This system of equations is solved using a Newton method with an additional line search for robustness. Using the notation

$$\mathbf{f} = \begin{bmatrix} \mathbf{f}_1 \\ f_2 \\ f_3 \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} \mathbf{t} \\ \Delta\lambda \\ \Delta\varepsilon_{33} \end{bmatrix},$$

a Newton step is completed as

$$\mathbf{x}^+ = \mathbf{x}^- - s\left(\frac{\partial\mathbf{f}}{\partial\mathbf{x}}\right)^{-1}\mathbf{f}$$

for a step size $s \le 1$ chosen such that the norm of the objective function is decreasing. The gradient of the objective function is given by

$$\nabla\mathbf{f} = \nabla\mathbf{f}_{1-\beta} + \nabla\mathbf{f}_\beta$$

where

$$\nabla \mathbf{f}_{1-\beta} = \begin{bmatrix} \mathbf{I} + 2G\Delta\lambda \mathbf{D}\nabla^2\sigma_{\text{eff}} & 2G\mathbf{D}\nabla\sigma_{\text{eff}} & -\mathbf{C}_3 \\ -(\nabla\sigma_{\text{eff}})^T & H & 0 \\ 2G\Delta\lambda \mathbf{e}^T\nabla^2\sigma_{\text{eff}} & 2G\mathbf{e}^T\nabla\sigma_{\text{eff}} & C_{33} \end{bmatrix}$$

$$\nabla \mathbf{f}_{\beta} = \begin{bmatrix} \Delta\lambda H \dfrac{\partial \mathbf{h}}{\partial \mathbf{t}} & H\mathbf{h} & 0 \\ \mathbf{0}^T & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

and

$$\mathbf{e} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \quad \mathbf{C}_3 = (K - 2G/3)\mathbf{e} \quad C_{33} = (K + 4G/3)\,.$$

G and K stands for the shear and bulk modulus, respectively. This algorithm requires computation of the effective stress hessian. The derivation of this is quite straightforward but the expression for it is rather long and is hence omitted in this report.

Material Model 134: Viscoelastic Fabric

The viscoelastic fabric model is a variation on the general viscoelastic Material Model 76. This model is valid for 3 and 4 node membrane elements only and is strongly recommended for modeling isotropic viscoelastic fabrics where wrinkling may be a problem. For thin fabrics, buckling can result in an inability to support compressive stresses; thus, a flag is included for this option. If bending stresses are important, use a shell formulation with Model 76.

Rate effects are taken into account through linear viscoelasticity by a convolution integral of the form:

$$\sigma_{ij} = \int_0^t g_{ijkl}\left(t-\tau\right)\frac{\partial \varepsilon_{kl}}{\partial \tau}\,d\tau \qquad (19.134.1)$$

where $g_{ijkl}(t-\tau)$ is the relaxation function.

If we wish to include only simple rate effects for the deviatoric stresses, the relaxation function is represented by six terms from the Prony series:

$$g(t) = \sum_{m=1}^{N} G_m e^{-\beta_m t} \qquad (19.134.2)$$

We characterize this in the input by shear modulii, G_i, and decay constants, β_i. An arbitrary number of terms, up to 6, may be used when applying the viscoelastic model.

For volumetric relaxation, the relaxation function is also represented by the Prony series in terms of bulk modulii:

$$k(t) = \sum_{m=1}^{N} K_m e^{-\beta_{k_m} t} \qquad (19.134.3)$$

Material Model 139: Modified Force Limited

This material model is available for the Belytschko resultant beam element only. Plastic hinges form at the ends of the beam when the moment reaches the plastic moment. The plastic moment versus rotation relationship is specified by the user in the form of a load curve and scale factor. The points of the load curve are (plastic rotation in radians, plastic moment). Both quantities should be positive for all points, with the first point being (zero, initial plastic moment). Within this constraint any form of characteristic may be used, including flat or falling curves. Different load curves and scale factors may be specified at each node and about each of the local s and t axes.

Axial collapse occurs when the compressive axial load reaches the collapse load. Collapse load versus collapse deflection is specified in the form of a load curve. The points of the load curve are either (true strain, collapse force) or (change in length, collapse force). Both quantities should be entered as positive for all points, and will be interpreted as compressive. The first point should be (zero, initial collapse load).

The collapse load may vary with end moment as well as with deflections. In this case several load-deflection curves are defined, each corresponding to a different end moment. Each load curve should have the same number of points and the same deflection values. The end moment is defined as the average of the absolute moments at each end of the beam and is always positive.

Stiffness-proportional damping may be added using the damping factor λ. This is defined as follows:

$$\lambda = \frac{2 * \xi}{\omega}$$

where ξ is the damping factor at the reference frequency ω (in radians per second). For example if 1% damping at 2Hz is required

$$\lambda = \frac{2 * 0.01}{2\pi * 2} = 0.001592$$

If damping is used, a small timestep may be required. LS-DYNA does not check this so to avoid instability it may be necessary to control the timestep via a load curve. As a guide, the timestep required for any given element is multiplied by $0.3L/c\lambda$ when damping is present (L = element length, c = sound speed).

Moment Interaction

Plastic hinges can form due to the combined action of moments about the three axes. This facility is activated only when yield moments are defined in the material input. A hinge forms when the following condition is first satisfied.

$$\left(\frac{M_r}{M_{ryield}}\right)^2+\left(\frac{M_s}{M_{syield}}\right)^2+\left(\frac{M_t}{M_{tyield}}\right)^2\geq 1$$

where,

M_r, M_s, M_t = current moments

$M_{ryield}, M_{syield}, M_{tyield}$ = yield moments

Note that scale factors for hinge behavior defined in the input will also be applied to the yield moments: for example, M_{syield} in the above formula is given by the input yield moment about the local axis times the input scale factor for the local s-axis. For strain-softening characteristics, the yield moment should generally be set equal to the initial peak of the moment-rotation load curve.

On forming a hinge, upper limit moments are set. These are given by

$$M_{r_{upper}}=MAX\left(M_r,\frac{M_{r_{yield}}}{2}\right)$$

and similarly for M_s and M_t.

Thereafter the plastic moments will be given by:

M_{rp} = min (M_{rcurve} , M_{rcurve}) and similarly for s and t

where

M_{rp} = current plastic moment

M_{rcurve} = moment taken from load curve at the current rotation scaled according to the scale factor.

The effect of this is to provide an upper limit to the moment that can be generated; it represents the softening effect of local buckling at a hinge site. Thus if a member is bent about its local s-axis it will then be weaker in torsion and about its local t-axis. For moment-softening curves, the effect is to trim off the initial peak (although if the curves subsequently harden, the final hardening will also be trimmed off).

It is not possible to make the plastic moment vary with the current axial load, but it is possible to make hinge formation a function of axial load and subsequent plastic moment a function of the moment at the time the hinge formed. This is discussed in the next section.

Independent plastic hinge formation

In addition to the moment interaction equation, Cards 7 through 18 allow plastic hinges to form independently for the s-axis and t-axis at each end of the beam and also for the torsional axis. A plastic hinge is assumed to form if any component of the current moment exceeds the yield moment as defined by the yield moment vs. axial force curves input on cards 7 and 8. If any of the 5 curves is omitted, a hinge will not form for that component. The curves can be defined for both compressive and tensile axial forces. If the axial force falls outside the range of

the curve, the first or last point in the curve will be used. A hinge forming for one component of moment does not effect the other components.

Upon forming a hinge, the magnitude of that component of moment will not be permitted to exceed the current plastic moment. The current plastic moment is obtained by interpolating between the plastic moment vs. plastic rotation curves input on cards 10, 12, 14, 16, or 18. Curves may be input for up to 8 hinge moments, where the hinge moment is defined as the yield moment at the time that the hinge formed. Curves must be input in order of increasing hinge moment and each curve should have the same plastic rotation values. The first or last curve will be used if the hinge moment falls outside the range of the curves. If no curves are defined, the plastic moment is obtained from the curves on cards 4 through 6. The plastic moment is scaled by the scale factors on lines 4 to 6.

A hinge will form if either the independent yield moment is exceeded or if the moment interaction equation is satisfied. If both are true, the plastic moment will be set to the minimum of the interpolated value and M_{rp}.

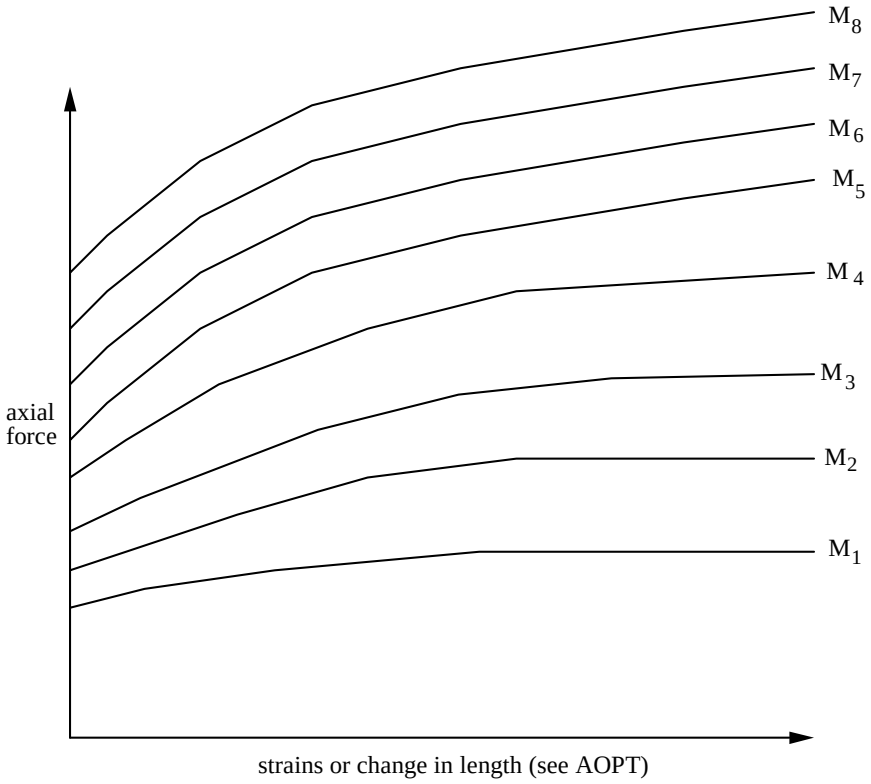


Figure 19.139.1. The force magnitude is limited by the applied end moment. For an intermediate value of the end moment LS-DYNA interpolates between the curves to determine the allowable force value.

Material Model 141: Rate Sensitive Polymer

$$\varepsilon_{ij} = D_o \exp\left[-\frac{1}{2}\left(\frac{Z_o^2}{3K_2}\right)\right]\left(\frac{S_{ij}-\Omega_{ij}}{\sqrt{K_2}}\right) \tag{19.141.1}$$

where D_o is the maximum inelastic strain rate, Z_o is the isotropic initial hardness of material, Ω_{ij} is the internal stress, S_{ij} is the deviatoric stress component, and K_2 is defined as follows:

$$K_2 = \frac{1}{2}(S_{ij}-\Omega_{ij})(S_{ij}-\Omega_{ij}) \tag{19.141.2}$$

and represent the second invariant of the overstress tensor. The elastic components of the strain are added to the inelastic strain to obtain the total strain. The following relationship defines the internal stress variable rate:

$$\Omega_{ij} = \frac{2}{3}q\Omega_m\dot{\varepsilon}_{ij}^I - q\Omega_{ij}\dot{\varepsilon}_e^I \tag{19.141.3}$$

where q is a material constant, Ω_m is a material constant that represents the maximum value of the internal stress, and $\dot{\varepsilon}_e^I$ is the effective inelastic strain.

Material Model 142: Transversely Anisotropic Crushable Foam

A new material model for low density, transversely isotropic crushable foams, has been developed at DaimlerChrysler by Hirth, Du Bois, and Weimar. Hirth, Du Bois, and Weimar determined that material model 26, MAT_HONEYCOMB, which is commonly used to model foams, can systematically over estimate the stress when it is loaded off-axis. Their new material model overcomes this problem without requiring any additional input. Their new model can possibly replace the MAT_HONEYCOMB material, which is currently used in the frontal offset and side impact barriers.

Many polymers used for energy absorption are low density, crushable foams with no noticeable Poisson effect. Frequently manufactured by extrusion, they are transversely isotropic. This class of material is used to enhance automotive safety in low velocity (bumper impact) and medium velocity (interior head impact) applications. These materials require a transversely isotropic, elastoplastic material with a flow rule allowing for large permanent volumetric deformations.

The MAT_HONEYCOMB model uses a local coordinate system defined by the user. One of the axes of the local system coincides with the extrusion direction of the honeycomb in the undeformed configuration. As an element deforms, its local coordinate system rotates with its mean rigid body motion. Each of the six stress components is treated independently, and each has its own law relating its flow stress to its plastic strain.

The effect of off-axis loading on the MAT_HONEYCOMB model can be estimated by restricting our considerations to plane strain in two dimensions. Our discussion is restricted to the response of the foam before it becomes fully compacted. After compaction, its response is modeled with conventional J_2 plasticity. The model reduces to

$$\begin{aligned} |\sigma_{11}| &\le \sigma_{11}^{y}(\varepsilon_V) \\ |\sigma_{22}| &\le \sigma_{22}^{y}(\varepsilon_V) \\ |\sigma_{12}| &\le \sigma_{12}^{y}(\varepsilon_V) \end{aligned} \tag{19.142.1}$$

where ε_V is the volumetric strain. For a fixed value of volumetric strain, the individual stress components respond in an elastic-perfectly plastic manner, i.e., the foam doesn't have any strain hardening.

In two dimensions, the stress tensor transforms according to

$$\begin{aligned} [\sigma] &= [R(\theta)]^T [\sigma_\theta][R(\theta)] \\ [R(\theta)] &= \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \end{aligned} \tag{19.142.2}$$

where θ is the angle of the local coordinate system relative to the global system. For uniaxial loading along the global 1-axis, the stress will be (accounting for the sign of the volume strain),

$$\sigma_{11} = \{[\cos(\theta)]^2 \sigma_{11}^{y} + [\sin(\theta)]^2 \sigma_{22}^{y} + 2 \cdot \sin(\theta)\cos(\theta)\sigma_{12}^{y}\}\,\mathrm{sgn}(\varepsilon_V)\} \tag{19.142.3}$$

assuming the strain is large enough to cause yielding in both directions.

If the shear strength is neglected, σ_{11} will vary smoothly between σ_{11}^{y} and σ_{22}^{y} and never exceed the maximum of the two yield stresses. This behavior is intuitively what we would like to see. However, if the value of shear yield stress isn't zero, σ_{11} will be greater than either σ_{11}^{y} or σ_{22}^{y}. To illustrate, if σ_{11}^{y} and σ_{22}^{y} are equal (a nominally isotropic response) the magnitude of the stress is

$$|\sigma_{11}| = \sigma_{11}^{y} + 2 \cdot \sin(\theta)\cos(\theta)\sigma_{12}^{y} \tag{19.142.4}$$

and achieves a maximum value at 45 degrees of

$$|\sigma_{11}| = \sigma_{11}^{y} + \sigma_{12}^{y}. \tag{19.142.5}$$

For cases where there is anisotropy, the maximum occurs at a different angle and will have a different magnitude, but it will exceed the maximum uniaxial yield stress. In fact, a simple calculation using Mohr's circle shows that the maximum value will be

$$\sigma_{\max}^{y} = \tfrac{1}{2}\left(\sigma_{11}^{y} + \sigma_{22}^{y}\right) + \tfrac{1}{2}\sqrt{\left(\sigma_{11}^{y} - \sigma_{22}^{y}\right)^2 + 4\sigma_{12}^{y}}\,. \tag{19.142.6}$$

To correct for the systematic overestimation of the off-axis strength by MAT_ HONEYCOMB, MAT_TRANSVERSELY_ISOTROPIC_CRUSHABLE_FOAM has been implemented in LS-DYNA. It uses a single yield surface, calculated dynamically from the six yield stresses specified by the user. The yield surface hardens and softens as a function of the volumetric strain through the yield stress functions. While the cost of the model is higher than for MAT_HONEYCOMB, its superior response off-axis makes it the model of choice for critical applications involving many types of low-density foams.

Material Model 143: Wood Model

The wood model is a transversely isotropic material and is available for solid elements. The development of this model was done by Murray [2002], who provided the documentation that follows, under a contract from the FHWA.

The general constitutive relation for an orthotropic material, written in terms of the principal material directions [Bodig & Jayne, 1993] is:

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{12} & C_{22} & C_{23} & 0 & 0 & 0 \\ C_{13} & C_{23} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & 2C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & 2C_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & 2C_{66} \end{bmatrix} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \end{bmatrix} \qquad (19.143.1)$$

The subscripts 1, 2, and 3 refer to the longitudinal, tangential, and radial, stresses and strains ($\sigma_1 = \sigma_{11}$, $\sigma_2 = \sigma_{22}$, $\sigma_3 = \sigma_{33}$, $\varepsilon_1 = \varepsilon_{11}$, $\varepsilon_2 = \varepsilon_{22}$, $\varepsilon_3 = \varepsilon_{33}$), respectively. The subscripts 4, 5, and 6 are in a shorthand notation that refers to the shearing stresses and strains $\sigma_4 = \sigma_{12}$, $\sigma_5 = \sigma_{13}$, $\sigma_6 = \sigma_{23}$, $\varepsilon_4 = \varepsilon_{12}$, $\varepsilon_5 = \varepsilon_{13}$, $\varepsilon_6 = \varepsilon_{23}$). As an alternative notation for wood, it is common to substitute L (longitudinal) for 1, T (tangential) for 2, and R (radial) for 3. The components of the constitutive matrix, C_{ij}, are listed here in terms of the nine independent elastic constants of an orthotropic material:

$$
\begin{aligned}
C_{11} &= E_{11}(1-\nu_{23}\nu_{32})/\Delta \\
C_{22} &= E_{22}(1-\nu_{31}\nu_{13})/\Delta \\
C_{33} &= E_{33}(1-\nu_{12}\nu_{21})/\Delta \\
C_{12} &= (\nu_{21}+\nu_{31}\nu_{23})E_{11}/\Delta \\
C_{13} &= (\nu_{31}+\nu_{21}\nu_{32})E_{11}/\Delta \\
C_{23} &= (\nu_{32}+\nu_{12}\nu_{31})E_{22}/\Delta \\
C_{44} &= G_{12} \\
C_{55} &= G_{13} \\
C_{66} &= G_{23} \\
\Delta &= 1-\nu_{12}\nu_{21}-\nu_{23}\nu_{32}-\nu_{31}\nu_{13}-2\nu_{21}\nu_{32}\nu_{13}
\end{aligned}
\qquad (19.143.2)
$$

The following identity, relating the dependent (minor Poisson's ratios ν_{21}, ν_{31}, and ν_{32}) and independent elastic constants, is obtained from symmetry considerations of the constitutive matrix:

$$
\frac{\nu_{ij}}{E_i} = \frac{\nu_{ji}}{E_j} \quad for \quad i, j = 1, 2, 3 \qquad (19.143.3)
$$

One common assumption is that wood materials are *transversely isotropic*. This means that the properties in the tangential and radial directions are modeled the same, *i.e.* $E_{22} = E_{33}$, $G_{12} = G_{13}$, and $\nu_{12} = \nu_{13}$. This reduces the number of independent elastic constants to five, E_{11}, E_{22}, ν_{12}, G_{12}, and G_{23}. Further, Poisson's ratio in the isotropic plane, ν_{23}, is not an independent quantity. It is calculated from the isotropic relation: $\nu = (E - 2G)/2G$ where $E = E_{22} = E_{33}$ and $G = G_{23}$. Transverse isotropy is a reasonable assumption because the difference between the tangential and radial properties of wood (particularly Southern yellow pine and Douglas fir) is small in comparison with the difference between the tangential and longitudinal properties.

The yield surfaces parallel and perpendicular to the grain are formulated from six ultimate strength measurements obtained from uniaxial and pure-shear tests on wood specimens:

X_T	Tensile strength parallel to the grain
X_C	Compressive strength parallel to the grain
Y_T	Tensile strength perpendicular to the grain
Y_C	Compressive strength perpendicular to the grain
$S_\parallel$	Shear strength parallel to the grain
$S_\perp$	Shear strength perpendicular to the grain

Here X and Y are the strengths parallel and perpendicular to the grain, respectively, and S are the shear strengths. The formulation is based on the work of Hashin [1980].

For the parallel modes, the yield criterion is composed of two terms involving two of the five stress invariants of a transversely isotropic material. These invariants are

$I_1 = \sigma_{11}$ *and* $I_4 = \sigma_{12}^2 + \sigma_{13}^2$. This criterion predicts that the normal and shear stresses are mutually weakening, *i.e.* the presence of shear stress reduces the strength below that measured in uniaxial stress tests. Yielding occurs when $f_{||} \geq 0$, where:

$$f_{\|} = \frac{\sigma_{11}^2}{X^2} + \frac{(\sigma_{12}^2 + \sigma_{13}^2)}{S_{\|}^2} - 1 \qquad X = \begin{cases} X_t & for \quad \sigma_{11} > 0 \\ X_c & for \quad \sigma_{11} < 0 \end{cases} \tag{19.143.4}$$

For the perpendicular modes, the yield criterion is also composed of two terms involving two of the five stress invariants of a transversely isotropic material. These invariants are $I_2 = \sigma_{22} + \sigma_{33}$ and $I_3 = \sigma_{23}^2 - \sigma_{22}\sigma_{33}$. Yielding occurs when $f_{\perp} \geq 0$, where:

$$f_{\perp} = \frac{(\sigma_{22} + \sigma_{33})^2}{Y^2} + \frac{(\sigma_{23}^2 - \sigma_{22}\sigma_{33})}{S_{\perp}^2} - 1 \qquad Y = \begin{cases} Y_t & for \quad \sigma_{22} + \sigma_{33} > 0 \\ Y_c & for \quad \sigma_{22} + \sigma_{33} < 0 \end{cases} \tag{19.143.5}$$

Each yield criterion is plotted in 3D in Figure 19.143.1 in terms of the parallel and perpendicular stresses. Each criterion is a smooth surface (no corners).

The plasticity algorithms limit the stress components once the yield criteria in [Murry 2002] are satisfied. This is done by returning the trial elastic stress state back to the yield surface. The stress and strain tensors are partitioned into elastic and plastic parts. Partitioning is done with a return mapping algorithm which enforces the *plastic consistency condition*.

Separate plasticity algorithms are formulated for the parallel and perpendicular modes by enforcing separate consistency conditions. The solution of each consistency condition determines the *consistency parameters*, $\Delta\lambda_{||}$ and $\Delta\lambda_{\perp}$. The $\Delta\lambda$ solutions, in turn, determine the stress updates. No input parameters are required.

The stresses are readily updated from the total strain increments and the consistency parameters, as follows:

$$\overline{\sigma}_{ij}^{n+1} = \sigma_{ij}^{*n+1} - C_{ijkl}\Delta\lambda \left.\frac{\partial f}{\partial \sigma_{kl}}\right|_n \tag{19.143.6}$$

$$\sigma_{ij}^{*n+1} = \sigma_{ij}^{n} + C_{ijkl}\Delta\varepsilon_{kl} \tag{19.143.7}$$

Here n denotes the n^{th} time step in the finite element analysis, and σ_{ij}^{*} are the trial elastic stress updates calculated from the total strain increments, $\Delta\varepsilon_{ij}$, prior to application of plasticity. Each normal stress update depends on the consistency parameters and yield surface functions for both the parallel ($\Delta\lambda = \Delta\lambda_{||}$ and $f = f_{||}$) and perpendicular ($\Delta\lambda = \Delta\lambda_{\perp}$ and $f = f_{\perp}$) modes. Each shear stress update depends on just one consistency parameter and yield surface function. If neither parallel nor perpendicular yielding occurs ($f_{\|}^{*} < 0$ *and* $f_{\perp}^{*} < 0$) then $\Delta\lambda = 0$ and the stress update is trivial: $\hat{\sigma}_{ij}^{n+1} = \sigma_{ij}^{*n+1}$.

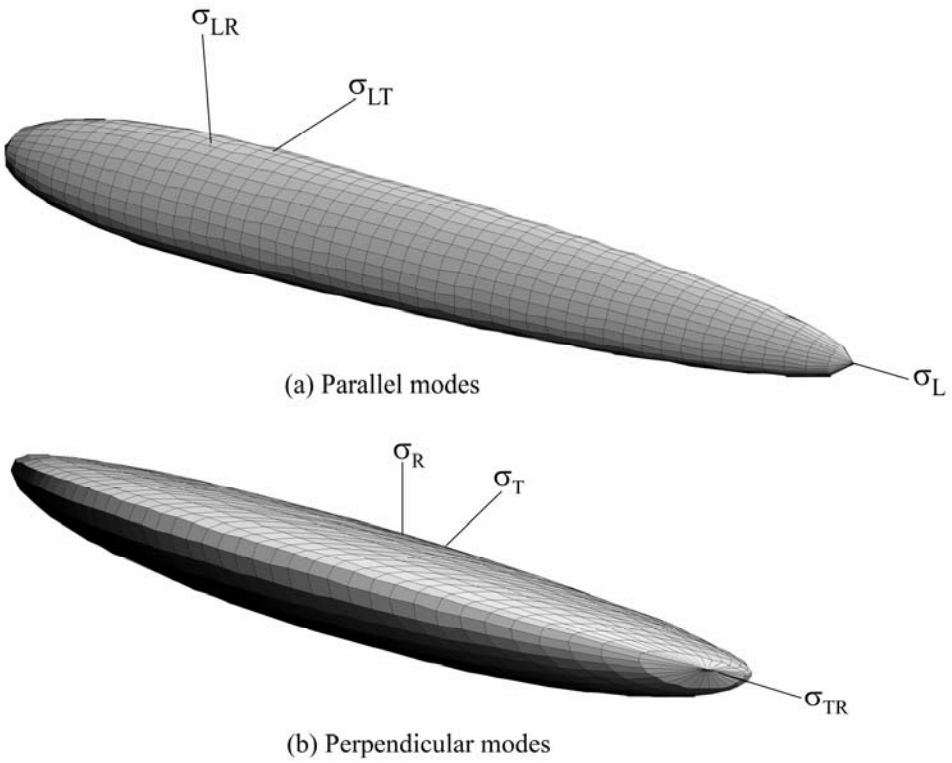
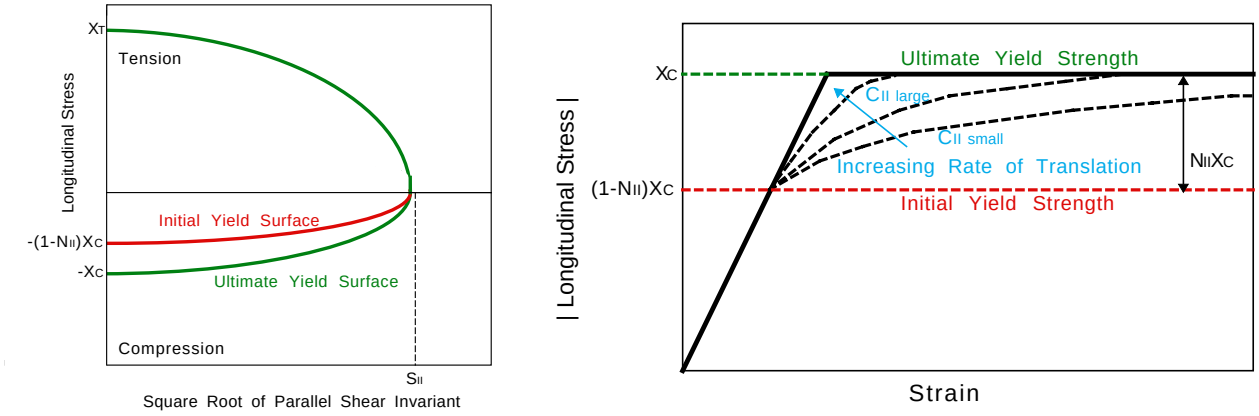


Figure 19.143.1. The yield criteria for wood produces smooth surfaces in stress space.

Wood exhibits pre-peak nonlinearity in compression parallel and perpendicular to the grain. Separate translating yield surface formulations are modeled for the parallel and perpendicular modes, which simulate gradual changes in moduli. Each initial yield surface hardens until it coincides with the ultimate yield surface. The initial location of the yield surface determines the onset of plasticity. The rate of translation determines the extent of the nonlinearity.



(a) Initial and ultimate yield surfaces. (b) Stress-strain behavior.

Figure 19.143.2. Pre-peak nonlinearity in compression is modeled with translating yield surfaces that allow the user to specify the hardening response.

For each mode (parallel and perpendicular), the user inputs two parameters: the initial yield surface location in uniaxial compression, *N*, and the rate of translation, *c*. Say the user wants pre-peak nonlinearity to initiate at 70% of the peak strength. The user will input $N = 0.3$ so that $1\text{-}N = 0.7$. If the user wants to harden rapidly, then a large value of *c* is input, like $c = 1$ msec. If the user wants to harden gradually, then a small value of *c* is input, like $c = 0.2$ msec.

The state variable that defines the translation of the yield surface is known as the *back stress*, and is denoted by α_{ij}. Hardening is modeled in compression, but not shear, so the only back stress required for the parallel modes is α_{11}. The value of the back stress is $\alpha_{11} = 0$ upon initial yield and $\alpha_{11} = -N_{||} X_c$ at ultimate yield (in uniaxial compression). The maximum back stress occurs at ultimate yield and is equal to the total translation of the yield surface in stress space. The back stress components required for the perpendicular modes are α_{22} and α_{33}. The value of the backstress sum is $\alpha_{22} + \alpha_{33} = 0$ upon initial yield and $\alpha_{22} + \alpha_{33} = -N_{\perp} Y_c$ at ultimate yield (biaxial compression without shear).

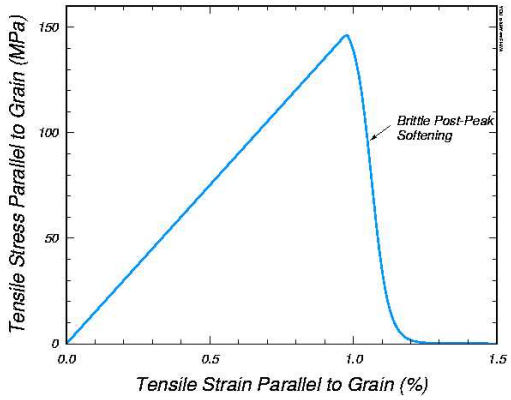
Separate damage formulations are modeled for the parallel and perpendicular modes. These formulations are loosely based on the work of Simo and Ju [1987]. If failure occurs in the parallel modes, then all six stress components are degraded uniformly. This is because parallel failure is catastrophic, and will render the wood useless. If failure occurs in the perpendicular modes, then only the perpendicular stress components are degraded. This is because perpendicular failure is not catastrophic: we expect the wood to continue to carry load in the parallel direction. Based on these assumptions, the following degradation model is implemented:

$$\begin{aligned}
d_m &= \max(d(\tau_{\parallel}), d(\tau_{\perp})) \\
d_{\parallel} &= d(\tau_{\parallel}) \\
\sigma_{11} &= (1-d_{\parallel})\bar{\sigma}_{11} \\
\sigma_{22} &= (1-d_m)\bar{\sigma}_{22} \\
\sigma_{33} &= (1-d_m)\bar{\sigma}_{33} \\
\sigma_{12} &= (1-d_{\parallel})\bar{\sigma}_{12} \\
\sigma_{13} &= (1-d_{\parallel})\bar{\sigma}_{13} \\
\sigma_{23} &= (1-d_m)\bar{\sigma}_{23}
\end{aligned} \qquad (19.143.8)$$

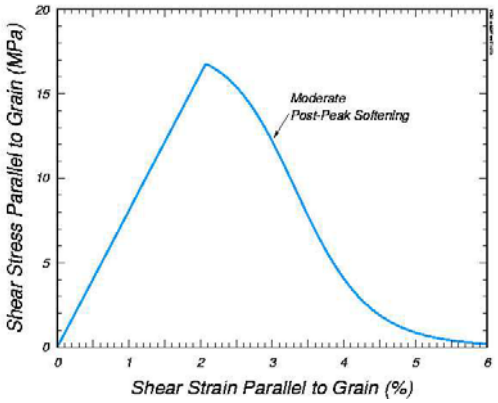
Here, each scalar damage parameter, *d*, transforms the stress tensor associated with the undamaged state, $\bar{\sigma}_{ij}$, into the stress tensor associated with the damaged state, σ_{ij}. The stress tensor $\bar{\sigma}_{ij}$ is calculated by the plasticity algorithm prior to application of the damage model. Each damage parameter ranges from zero for no damage and approaches unity for maximum damage. Thus *1-d* is a reduction factor associated with the amount of damage. Each damage parameter evolves as a function of a strain energy-type term. Mesh size dependency is regulated via a length scale based on the element size (cube root of volume). Damage-based softening is brittle in tension, less brittle in shear, and ductile (no softening) in compression, as demonstrated in Figure 19.143.2.

Element erosion occurs when an element fails in the parallel mode, and the parallel damage parameter exceeds $d_{||} = 0.99$. Elements do not automatically erode when an element fails in the perpendicular mode. A flag is available, which, when set, allows elements to erode

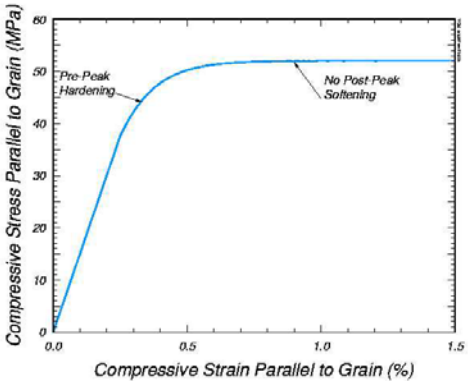
when the perpendicular damage parameter exceeds $d_{\perp} = 0.99$. Setting this flag is not recommended unless excessive perpendicular damage is causing computational difficulties.



(a) Tensile softening.



(b) Shear softening.



(c) Compressive yielding.

Figure 19.143.3. Softening response modeled for parallel modes of Southern yellow pine.

Data available in the literature for pine [Reid & Peng, 1997] indicates that dynamic strength enhancement is more pronounced in the perpendicular direction than in the parallel

direction. Therefore, separate rate effects formulations are modeled for the parallel and perpendicular modes. The formulations increase strength with increasing strain rate by expanding each yield surface:

$$\begin{aligned} \sigma_{11} &= X + E_{11}\dot{\varepsilon}\eta_{\parallel} \quad \textit{Parallel} \\ \sigma_{22} &= Y + E_{22}\dot{\varepsilon}\eta_{\perp} \quad \textit{Perpendicular} \end{aligned} \tag{19.143.9}$$

Here X and Y are the static strengths, σ_{11} and σ_{22} are the dynamic strengths, and $E_{11}\dot{\varepsilon}\eta_{\parallel}$ and $E_{22}\dot{\varepsilon}\eta_{\perp}$ are the excess stress components. The excess stress components depend on the value of the fluidity parameter, η, as well as the stiffness and strain rate. The user inputs two values, η_0 and n, to define each fluidity parameter:

$$\begin{aligned} \eta_{\parallel} &= \frac{\eta_{0_{\parallel}}}{\dot{\varepsilon}^{n_{\parallel}}} \\ \eta_{\perp} &= \frac{\eta_{0_{\perp}}}{\dot{\varepsilon}^{n_{\perp}}} \end{aligned} \tag{19.143.10}$$

The two parameter formulation [Murray, 1997] allows the user to model a nonlinear variation in dynamic strength with strain rate. Setting n=0 allows the user to model a linear variation in dynamic strength with strain rate.

Material Model 144: Pitzer Crushable Foam

The logarithmic volumetric strain is defined in terms of the relative volume, V, as:

$$\gamma = -\ln(V)$$

In defining the curves the stress and strain pairs should be positive values starting with a volumetric strain value of zero.

Material Model 147: FHWA Soil Model

A brief discussion of the FHWA soil model is given. The elastic properties of the soil are isotropic. The implementation of the modified Mohr-Coulomb plasticity surface is based on the work of Abbo and Sloan [1995]. The model is extended to include excess pore water effects, strain softening, kinematic hardening, strain rate effects, and element deletion.

The modified yield surface is a hyperbola fitted to the Mohr-Coulomb surface. At the crossing of the pressure axis (zero shear strength) the modified surface is a smooth surface and it is perpendicular to the pressure axis. The yield surface is given as

$$F = -P\sin\varphi + \sqrt{J_2 K(\theta)^2 + ahyp^2 \sin^2\varphi} - c\cos\varphi = 0 \tag{19.147.1}$$

where P is the pressure, φ is the internal friction angle, $K(\theta)$ is a function of the angle in

$$F = -P\sin\varphi + \sqrt{J_2 K(\theta)^2 + \text{ahyp}^2 \sin^2\varphi} - \text{c}\cos\varphi = 0$$

deviatoric plane, $\sqrt{\text{J}_2}$ is the square root of the second invariant of the stress deviator, c is the amount of cohesion and

$$\cos 3\theta = \frac{3\sqrt{3}J_3}{2J_2^{\frac{3}{2}}} \qquad (19.147.2)$$

J_3 is the third invariant of the stress deviator, ahyp is a parameter for determining how close to the standard Mohr-Coulomb yield surface the modified surface is fitted. If the user defined parameter, ahyp, is input as zero, the standard Mohr-Coulomb surface is recovered. The parameter ahyp should be set close to zero, based on numerical considerations, but always less than $c\cot\varphi$. It is best not to set the cohesion, c, to very small values as this causes excessive iterations in the plasticity routines.

To generalize the shape in the deviatoric plane, we have changed the standard Mohr-Coulomb $\text{K}(\boldsymbol{\theta})$ function to a function used by Klisinski [1985]

$$\text{K}(\theta) = \frac{4(1-\text{e}^2)\cos^2\theta + (2\text{e}-1)^2}{2(1-\text{e}^2)\cos\theta + (2\text{e}-1)[4(1-\text{e}^2)\cos^2\theta + 5\text{e}^2 - 4\text{e}]^{\frac{1}{2}}} \qquad (19.147.3)$$

where e is a material parameter describing the ratio of triaxial extension strength to triaxial compression strength. If e is set equal to 1, then a circular cone surface is formed. If e is set to 0.55, then a triangular surface is found, $\text{K}(\boldsymbol{\theta})$ is defined for $0.5 < e \le 1.0$.

To simulate non-linear strain hardening behavior the friction, angle φ is increased as a function of the effective plastic strain,

$$\Delta\phi = \text{E}_\text{t}\left(1 - \frac{\phi - \phi_{\text{init}}}{\text{A}_\text{n}\phi_{\max}}\right)\Delta\varepsilon_{\text{eff plas}}, \qquad (19.147.4)$$

where $\varepsilon_{\text{eff plas}}$ is the effective plastic strain. A_n is the fraction of the peak strength internal friction angle where nonlinear behavior begins, $0 < \text{A}_\text{n} \le 1$. The input parameter E_t determines the rate of the nonlinear hardening.

To simulate the effects of moisture and air voids including excess pore water pressure, both the elastic and plastic behaviors can be modified. The bulk modulus is

$$K = \frac{K_i}{1 + K_i D_1 n_{cur}} \qquad (19.147.5)$$

where

K_i = initial bulk modulus

n_{cur} = current porosity = $Max[0,(w-\varepsilon_v)]$

w = volumetric strain corresponding to the volume of air voids $= n(1-S)$

ε_v = total volumetric strain

D_1 = material constant controlling the stiffness before the air voids are collapsed

n = porosity of the soil = $\dfrac{e}{1+e}$

e = void ratio = $\dfrac{\gamma_{sp}(1+m_c)}{\rho}-1$

S = degree of saturation = $\dfrac{\rho m_c}{n(1+m_c)}$

and ρ, γ, m_c are the soil density, specific gravity, and moisture content, respectively.

Figure 19.147.1 shows the effect of the D_1 parameter on the pressure-volumetric strain relationship (bulk modulus). The bulk modulus will always be a monotonically increasing value, i.e.,

$$K_{j+1} = \begin{cases} \dfrac{K_i}{1+K_i D_1 n_{cur}} & if\ \varepsilon_{v\ j+1} > \varepsilon_{vj} \\ K_j & if\ \varepsilon_{v\ j+1} \le \varepsilon_{vj} \end{cases} \qquad (19.147.6)$$

Note that the model is following the standard practice of assuming compressive stresses and strains are positive. If the input parameter D_1 is zero, then the standard linear elastic bulk modulus behavior is used.

To simulate the loss of shear strength due to excess pore water effects, the model uses a standard soil mechanics technique [Holtz and Kovacs, 1981] of reducing the total pressure, P, by the excess pore water pressure, u, to get an " effective pressure", P′; therefore,

$$\mathrm{P}' = \mathrm{P} - \mathrm{u}$$

Figure 19.147.2 shows pore water pressure will affect the algorithm for the plasticity surface. The excess pore water pressure reduces the total pressure, which will lower the shear strength, $\sqrt{J_2}$. A large excess pore water pressure can cause the effective pressure to become zero.

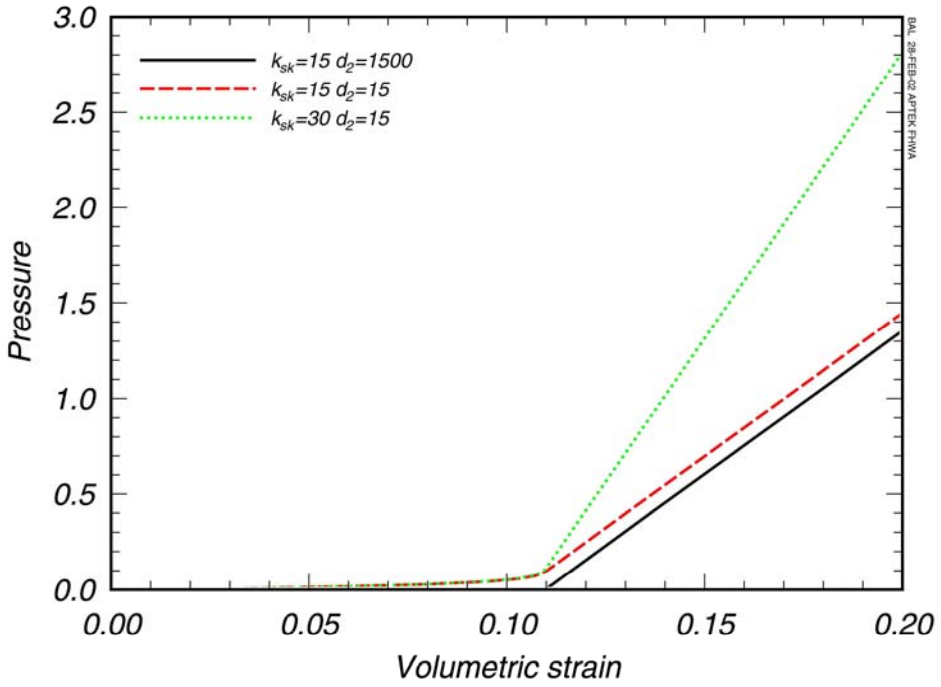


Figure 19.147.1. Pressure versus volumetric strain showing the effects of D1 parameter.

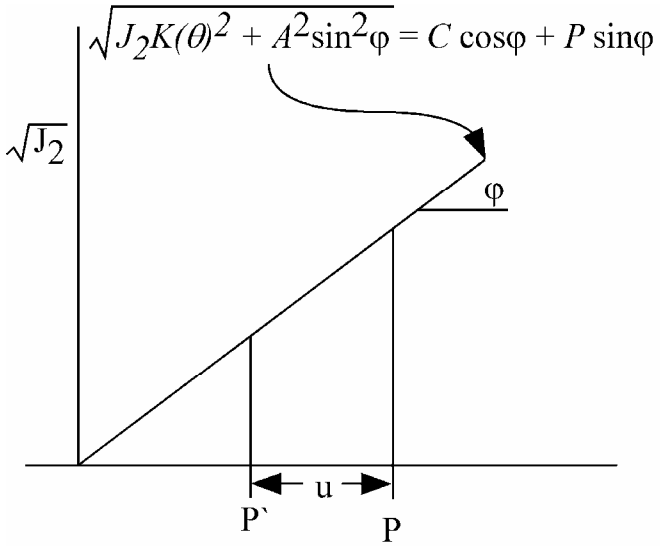


Figure 19.147.2. The effect on pressure due to pore water pressure.

To calculate the pore water pressure, u, the model uses an equation similar to the equation used for the moisture effects on the bulk modulus.

$$u = \frac{K_{sk}}{1 + K_{sk} D_2 n_{cur}} \varepsilon_v \qquad (19.147.7)$$

where

K_{sk} = bulk modulus for soil without air voids (skeletal bulk modulus)

n_{cur} = current porosity = $Max[0,(w - \varepsilon_v)]$

w = volumetric strain corresponding to the volume of air voids $= n(1 - S)$

ε_v = total volumetric strain

D_2 = material constant controlling the pore water pressure before the air voids are collapsed $D_2 \geq 0$

n = porosity of the soil = $\frac{e}{1+e}$

e = void ratio = $\frac{\gamma_{sp}(1+m_c)}{\rho} - 1$

S = degree of saturation = $\frac{\rho m_c}{n(1+m_c)}$

and ρ, γ, m_c are the soil density, specific gravity, and moisture content, respectively. The pore water pressure will not be allowed to become negative, $u \geq 0$.

Figure 19.147.3 is a plot of the pore pressure versus volumetric strain for different parameter values. With the D_2 parameter set relatively high compared to K_{sk} there is no pore pressure until the volumetric strain is greater than the strains associated with the air voids. However, as D_2 is lowered, the pore pressure starts to increase before the air voids are totally collapsed. The K_{sk} parameter affects the slope of the post-void collapse pressure - volumetric behavior.

The parameter D_2 can be found from Skempton pore water pressure parameter B, where B is defined as [Holtz and Kovacs, 1981]:

$$B = \frac{1}{1 + n\frac{K_{sk}}{K_v}} \qquad (19.147.8)$$

$$\therefore \quad D_2 = \frac{1 - B}{B\, K_{sk}[n(1-S)]}.$$

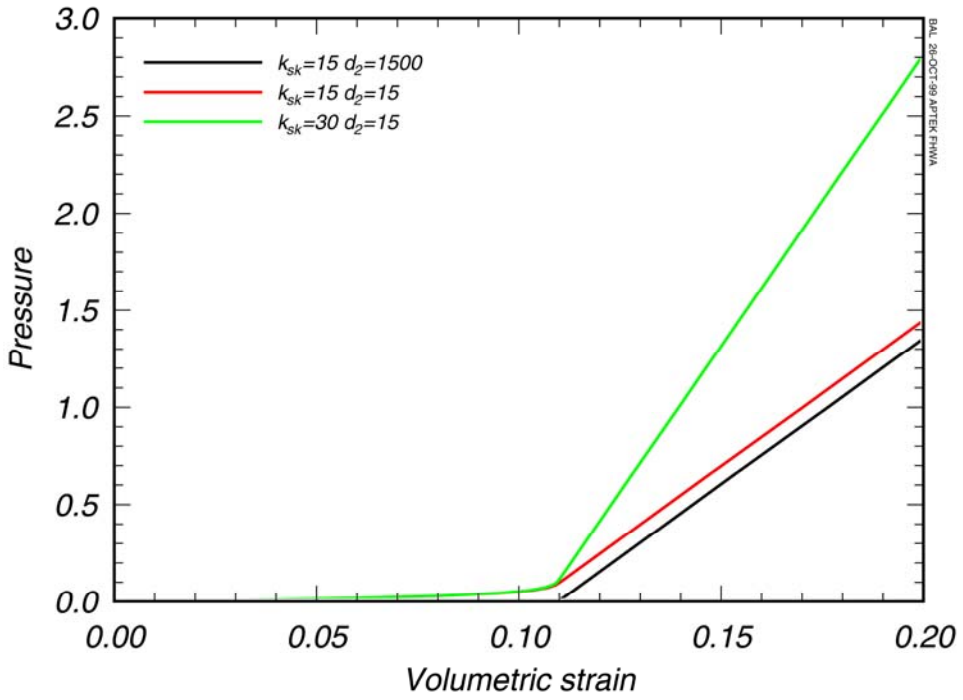


Figure 19.147.3. The effects of D_2 and K_{sk} parameters on pore water pressure.

To simulate strain softening behavior the FHWA soil model uses a continuum damage algorithm. The strain-based damage algorithm is based on the work of J. W. Ju and J. C. Simo [1987, 1989]. They proposed a strain based damage criterion, which is uncoupled from the plasticity algorithm.

For the damage criterion,

$$\xi = -\frac{1}{K_i} \int \bar{P} d\varepsilon_{pv} \qquad (19.147.9)$$

where $\bar{P}$ is the pressure and ε_{pv} is the plastic volumetric strain, the damaged stress is found from the undamaged stresses.

$$\sigma = (1-d)\bar{\sigma}$$

where d is the isotropic damage parameter. The damage parameter is found at step $j+1$ as:

$$\begin{aligned} d_{j+1} &= d_j \ldots\ldots\ldots\ldots\ldots\ldots\ldots if \quad \xi_{j+1} \le r_j \\ d_{j+1} &= \frac{\xi_{j+1}-\xi_0}{\alpha-\xi_0} \ldots\ldots\ldots\ldots\ldots if \quad \xi_{j+1} > r_j \end{aligned} \qquad (19.147.10)$$

where r_t is a damage threshold surface, $r_{j+1} = \max\{r_j, \xi_{j+1}\}$, and $\xi_0 = r_0$ (Dint). The mesh sensitivity parameter, α, will be described below.

Typically, the damage, d, varies from 0 to a maximum of 1. However, some soils can have a residual strength that is pressure dependent. The residual strength is represented by φ_{res}, the minimum internal friction angle

The maximum damage allowed is related to the internal friction angle of residual strength by:

$$d_{\max} = \frac{\sin\varphi - \sin\varphi_{res}}{\sin\varphi} \qquad (19.147.11)$$

If $\varphi_{res} > 0$, then $d_{\max}$, the maximum damage, will not reach 1, and the soil will have some residual strength.

When material models include strain softening, special techniques must be used to prevent mesh sensitivity. Mesh sensitivity is the tendency of the finite element model/analysis to produce significantly different results as the element size is reduced. The mesh sensitivity occurs because the softening in the model concentrates in one element. As the element size is reduced the failure becomes localized in smaller volumes, which causes less energy to be dissipated by the softening leading to instabilities or at least mesh sensitive behavior.

To eliminate or reduce the effects of strain softening mesh sensitivity, the softening parameter, α (the strain at full damage), must be modified as the element size changes. The FHWA soil model uses an input parameter, "void formation", G_f, that is like fracture energy material property for metals. The void formation parameter is the area under the softening region of the pressure volumetric strain curve times the cube root of the element volume, $V^{\frac{1}{3}}$.

$$G_f = V^{\frac{1}{3}} \int_{\xi_0}^{\alpha} P d\varepsilon_v = \frac{P_{peak}(\alpha - \xi_0)V^{\frac{1}{3}}}{2} \qquad (19.147.12)$$

with ξ_0, the volumetric strain at peak pressure (strain at initial damage, Dint). Then α can be found as a function of the volume of the element V:

$$\alpha = \frac{2G_f}{K\xi_0 V^{1/3}} + \xi_0 \qquad (19.147.13)$$

If G_f is made very small relative to $K\xi_0 V^{1/3}$, then the softening behavior will be brittle.

Strain-rate enhanced strength is simulated by a two-parameter Devaut-Lions viscoplastic update algorithm, developed by Murray [1997]. This algorithm interpolates between the elastic trial stress (beyond the plasticity surface) and the inviscid stress. The inviscid stresses ($\bar{\sigma}$) are on the plasticity surface.

$$\bar{\sigma}_{vp} = (1-\varsigma)\bar{\sigma} + \varsigma\bar{\sigma}_{trial} \qquad (19.147.14)$$

where $$\varsigma = \frac{\eta}{\Delta t + \eta}, \text{ and } \eta = (\frac{\gamma_r}{\dot{\varepsilon}})^{(vn-1)/vn}$$

As ζ approaches 1, then the viscoplastic stress becomes the elastic trial stress. Setting the input value $\gamma_r = 0$ eliminates any strain-rate enhanced strength effects.

The model allows element deletion, if needed. As the strain softening (damage) increases, the effective stiffness of the element can get very small, causing severe element distortion and hourglassing. The element can be "deleted" to remedy this behavior. There are two input parameters that affect the point of element deletion. DAMLEV is the damage threshold where element deletion will be considered. EPSPRMAX is the maximum principal strain where element will be deleted. Therefore,

$$d \geq \text{DAMLEV} \quad \text{and} \quad \varepsilon_{pr\,max} > \quad \text{EPSPRMAX} \qquad (19.147.15)$$

for element deletion to occur. If DAMLEV is set to zero, there is no element deletion. Care must be taken when employing element deletion to assure that the internal forces are very small (element stiffness is zero) or significant errors can be introduced into the analysis.

The keyword option, **NEBRASKA,** gives the soil parameters used to validate the material model with experiments performed at University of Nebraska at Lincoln. The units for this default inputs are milliseconds, kilograms, and millimeters. There are no required input parameters except material id (MID). If different units are desired the unit conversion factors that need to *multiply* the default parameters can be input.

Material Model 154: Deshpande-Fleck Foam

$$\Phi = \hat{\sigma} - \sigma_Y \qquad (19.154.1)$$

The equivalent stress, $\hat{\sigma}$, is given by:

$$\hat{\sigma}^2 = \frac{\sigma_{VM}^2 + \alpha^2 \sigma_m^2}{1 + (\alpha/3)^2} \qquad (19.154.2)$$

where, σ_{VM}, is the von Mises effective stress,

$$\sigma_{VM} = \sqrt{\frac{3}{2}\boldsymbol{\sigma}^{dev} : \boldsymbol{\sigma}^{dev}} \qquad (19.154.3)$$

and, σ_m and $\boldsymbol{\sigma}^{dev}$, is the mean and deviatoric stress

$$\begin{aligned} \sigma_m &= \text{tr}(\boldsymbol{\sigma}) \\ \boldsymbol{\sigma}^{dev} &= \boldsymbol{\sigma} - \sigma_m \mathbf{I} \end{aligned} \qquad (19.154.4)$$

The yield stress σ_Y can be expressed as

$$\sigma_Y = \sigma_p + \gamma \frac{\hat{\varepsilon}}{\varepsilon_D} + \alpha_2 \left(\frac{1}{1-\left(\hat{\varepsilon}/\varepsilon_D\right)^{\beta}} \right) \tag{19.154.5}$$

Here, σ_p, α_2, γ and β are material parameters. The densification strain, ε_D, is defined as

$$\varepsilon_D = -\ln\left(\frac{\rho_f}{\rho_{f0}} \right) \tag{19.154.6}$$

where ρ_f is the foam density and ρ_{f0} is the density of the virgin material.

Material Model 156: Muscle

The material behavior of the muscle model is adapted from *MAT_SPRING_MUSCLE, the spring muscle model and treated here as a standard material. The initial length of muscle is calculated automatically. The force, relative length and shortening velocity are replaced by stress, strain and strain rate. A new parallel damping element is added.

The strain and normalized strain rate are defined respectively as

$$\begin{aligned} \varepsilon &= \frac{l}{l_o} - 1 = L - 1 \\ \dot{\varepsilon} &= \frac{\dot{l}}{l_o\, \dot{\varepsilon}_{\max}} = \frac{V^M}{l_o * \left(SRM * SFR\right)} = \frac{V^M}{\left(l_o * SRM\right) * SFR} = \frac{V^M}{V_{\max} * SFR} = V \end{aligned} \tag{19.156.1}$$

where $l_o =$, is the original muscle length.

From the relation above, it is known:

$$l_o = \frac{l_0}{1+\varepsilon_0} \tag{19.156.2}$$

where $\varepsilon_0 = SNO$; $l_0 =$ muscle length at time 0.

Stress of Contractile Element is:

$$\sigma_1 = \sigma_{\max}\, a(t)\, f(\varepsilon)\, g(\dot{\varepsilon}) \tag{19.156.3}$$

where $\sigma_{\max} = PIS$; $a(t) = ALM$; $f(\varepsilon) = SVS$; $g(\dot{\varepsilon}) = SVR$.

Stress of Passive Element is:

$$\sigma_2 = \sigma_{\max} h(\varepsilon) \tag{19.156.4}$$

For exponential relationship: $h(\varepsilon) = \begin{cases} 0 & \varepsilon \le 0 \\ \dfrac{1}{\exp(c)-1}\left[\exp\left(\dfrac{c\varepsilon}{L_{\max}}\right)-1\right] & \varepsilon > 0 \quad c \ne 0 \\ \varepsilon / L_{\max} & \varepsilon > 0 \quad c = 0 \end{cases}$

where $L_{\max} = 1 + SSM$; and $c = CER$.

Stress of Damping Element is:

$$\boldsymbol{\sigma}_3 = D\dot{\boldsymbol{\varepsilon}}_{\max}\dot{\boldsymbol{\varepsilon}} \tag{19.156.5}$$

Total Stress is:

$$\sigma = \sigma_1 + \sigma_2 + \sigma_3 \tag{19.156.6}$$

Material Model 158: Rate Sensitive Composite Fabric

See material type 58, Laminated Composite Fabric, for the treatment of the composite material.

Rate effects are taken into account through a Maxwell model using linear viscoelasticity by a convolution integral of the form:

$$\sigma_{ij} = \int_0^t g_{ijkl}(t-\tau)\frac{\partial \varepsilon_{kl}}{\partial \tau} d\tau \tag{19.158.1}$$

where $g_{ijkl}(t-\tau)$ is the relaxation function for different stress measures. This stress is added to the stress tensor determined from the strain energy functional. Since we wish to include only simple rate effects, the relaxation function is represented by six terms from the Prony series:

$$g(t) = \sum_{m=1}^{N} G_m e^{-\beta_m t} \tag{19.158.2}$$

We characterize this in the input by the shear moduli, G_i, and the decay constants, β_i. An arbitrary number of terms, not exceeding 6, may be used when applying the viscoelastic model. The composite failure is not directly affected by the presence of the viscous stress tensor.

Material Model 159: Continuous Surface Cap Model

This is a cap model with a smooth intersection between the shear yield surface and hardening cap, as shown in Figure 19.159.1. The initial damage surface coincides with the yield surface. Rate effects are modeled with viscoplasticity.

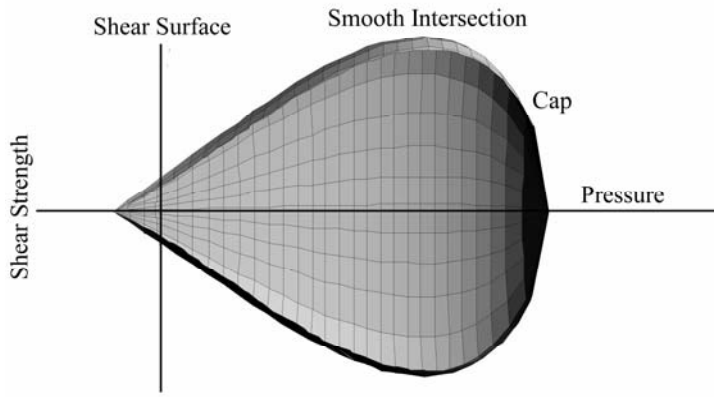


Figure 19.159.1. General Shape of the concrete model yield surface in two-dimensions.

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:

$$\begin{aligned} J_1 &= 3P \\ J_2' &= \frac{1}{2} S_{ij} S_{ij} \\ J_3' &= \frac{1}{3} S_{ij} S_{jk} S_{ki} \end{aligned} \qquad (19.159.1)$$

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

$$f\left(J_1, J_2', J_3', \kappa\right) = J_2' - \Re^2 F_f^2 F_c \qquad (19.159.2)$$

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:

$$F_f(J_1) = \alpha - \lambda \exp^{-\beta J_1} + \theta J_1 \qquad (19.159.3)$$

Here the values of $\alpha, \beta, \lambda, \text{and } \theta$ 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 $\Re$ determines the strength of concrete for any state of stress relative to the strength for TXC, via $\Re F_f$. Strength in torsion is modeled as $Q_1 F_f$. Strength in TXE is modeled as $Q_2 F_f$, where:

$$\begin{aligned} Q_1 &= \alpha_1 - \lambda_1 exp^{-\beta_1 J_1} + \theta_1 J_1 \\ Q_2 &= \alpha_2 - \lambda_2 exp^{-\beta_2 J_1} + \theta_2 J_1 \end{aligned} \tag{19.159.4}$$

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:

$$F_c(J_1, \kappa) = 1 - \frac{\left[J_1 - L(\kappa)\right]\left[\left|J_1 - L(\kappa)\right| + J_1 - L(\kappa)\right]}{2\left[X(\kappa) - L(\kappa)\right]^2} \tag{19.159.5}$$

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

$$L(\kappa) = \begin{cases} \kappa & if \quad \kappa > \kappa_0 \\ \kappa_0 & otherwise \end{cases} \tag{19.159.6}$$

The equation for F_c is equal to unity for $J_1 \le L(\kappa)$. 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:

$$X(\kappa) = L(\kappa) + RF_f(L(\kappa)) \tag{19.159.7}$$

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:

$$\varepsilon_v^p = W\left(1-\exp^{-D_1(X-X_0)-D_2(X-X_0)^2}\right) \qquad (19.159.8)$$

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.

$$\sigma_{ij}^d = (1-d)\sigma_{ij}^{vp} \qquad (19.159.9)$$

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:

$$\tau_c = \sqrt{\frac{1}{2}\sigma_{ij}\varepsilon_{ij}} \qquad (19.159.10)$$

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:

$$\tau_t = \sqrt{E\,\varepsilon_{max}^2} \qquad (19.159.11)$$

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 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:

$$A = A(dmax + 0.001)^{pmod} \qquad (19.159.12)$$

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 specifies 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.

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

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

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

Element Erosion. An element loses all strength and stiffness as $d \rightarrow 1$. 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}:

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

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:

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

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

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

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:

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

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:

$$G_f^{rate} = G_f\left(1+\frac{E\dot{\varepsilon}\eta}{f'}\right)^{repow} \qquad (19.159.13)$$

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.

Material Models 161 and 162: Composite MSC

The unidirectional and fabric layer failure criteria and the associated property degradation models for material 161 are described as follows. All the failure criteria are expressed in terms of stress components based on ply level stresses $(\sigma_a, \sigma_b, \sigma_c, \tau_{ab}, \tau_{bc}, \tau_{ca})$ and the associated elastic moduli are $(E_a, E_b, E_c, G_{ab}, G_{bc}, G_{ca})$. Note that for the unidirectional model, a, b and c denote the fiber, in-plane transverse and out-of-plane directions, respectively, while for the fabric model, a, b and c denote the in-plane fill, in-plane warp and out-of-plane directions, respectively.

Unidirectional Lamina Model

Three criteria are used for fiber failure, one in tension/shear, one in compression and another one in crush under pressure. They are chosen in terms of quadratic stress forms as follows:

Tensile/shear fiber mode:

$$f_1 = \left(\frac{\langle \sigma_a \rangle}{S_{aT}} \right)^2 + \left(\frac{\tau_{ab}^2 + \tau_{ca}^2}{S_{FS}^2} \right) - 1 = 0 \qquad (19.161.1)$$

Compression fiber mode:

$$f_2 = \left(\frac{\langle \sigma_a^{'} \rangle}{S_{aC}} \right)^2 - 1 = 0, \quad \sigma_a^{'} = -\sigma_a + \left\langle -\frac{\sigma_b + \sigma_c}{2} \right\rangle \qquad (19.161.2)$$

Crush mode:

$$f_3 = \left(\frac{\langle p \rangle}{S_{FC}} \right)^2 - 1 = 0, \quad p = -\frac{\sigma_a + \sigma_b + \sigma_c}{3} \qquad (19.161.3)$$

where $\langle\ \rangle$ are Macaulay brackets, S_{aT} and S_{aC} are the tensile and compressive strengths in the fiber direction, and S_{FS} and S_{FC} are the layer strengths associated with the fiber shear and crush failure, respectively.

Matrix mode failures must occur without fiber failure, and hence they will be on planes parallel to fibers. For simplicity, only two failure planes are considered: one is perpendicular to the planes of layering and the other one is parallel to them. The matrix failure criteria for the failure plane perpendicular and parallel to the layering planes, respectively, have the forms:
Perpendicular matrix mode:

$$f_4 = \left(\frac{\langle \sigma_b \rangle}{S_{bT}}\right)^2 + \left(\frac{\tau_{bc}}{S'_{bc}}\right)^2 + \left(\frac{\tau_{ab}}{S_{ab}}\right)^2 - 1 = 0 \qquad (19.161.4)$$

Parallel matrix mode (Delamination):

$$f_5 = S^2\left\{\left(\frac{\langle \sigma_c \rangle}{S_{bT}}\right)^2 + \left(\frac{\tau_{bc}}{S''_{bc}}\right)^2 + \left(\frac{\tau_{ca}}{S_{ca}}\right)^2\right\} - 1 = 0 \qquad (19.161.5)$$

where S_{bT} is the transverse tensile strength. Based on the Coulomb-Mohr theory, the shear strengths for the transverse shear failure and the two axial shear failure modes are assumed to be the forms,

$$S_{ab} = S_{ab}^{(0)} + \tan(\varphi)\langle -\sigma_b \rangle \qquad (19.161.6a)$$

$$S'_{bc} = S_{bc}^{(0)} + \tan(\varphi)\langle -\sigma_b \rangle \qquad (19.161.6b)$$

$$S_{ca} = S_{ca}^{(0)} + \tan(\varphi)\langle -\sigma_c \rangle \qquad (19.161.6c)$$

$$S''_{bc} = S_{bc}^{(0)} + \tan(\varphi)\langle -\sigma_c \rangle \qquad (19.161.6d)$$

where φ is a material constant as tan(φ) is similar to the coefficient of friction, and $S_{ab}^{(0)}$, $S_{ca}^{(0)}$ and $S_{bc}^{(0)}$ are the shear strength values of the corresponding tensile modes.

Failure predicted by the criterion of f_4 can be referred to as transverse matrix failure, while the matrix failure predicted by f_5, which is parallel to the layer, can be referred as the delamination mode when it occurs within the elements that are adjacent to the ply interface. Note that a scale factor S is introduced to provide better correlation of delamination area with experiments. The scale factor S can be determined by fitting the analytical prediction to experimental data for the delamination area.

When fiber failure in tension/shear mode is predicted in a layer by f_1, the load carrying capacity of that layer is completely eliminated. All the stress components are reduced to zero instantaneously (100 time steps to avoid numerical instability). For compressive fiber failure, the layer is assumed to carry a residual axial load, while the transverse load carrying capacity is reduced to zero. When the fiber compressive failure mode is reached due to f_2, the axial layer compressive strength stress is assumed to reduce to a residual value S_{RC} ($= SFFC * S_{AC}$). The axial stress is then assumed to remain constant, i.e., $\sigma_a = -S_{RC}$, for continuous compressive loading, while the subsequent unloading curve follows a reduced axial modulus to zero axial stress and strain state. When the fiber crush failure occurs, the material is assumed to behave elastically for compressive pressure, p > 0, and to carry no load for tensile pressure, p < 0.

When a matrix failure (delamination) in the a-b plane is predicted, the strength values for $S_{ca}^{(0)}$ and $S_{bc}^{(0)}$ are set to zero. This results in reducing the stress components σ_c, τ_{bc} and τ_{ca} to the fractured material strength surface. For tensile mode, $\sigma_c > 0$, these stress components are reduced to zero. For compressive mode, $\sigma_c < 0$, the normal stress σ_c is assumed to deform elastically for the closed matrix crack. Loading on the failure envelop, the shear stresses are assumed to 'slide' on the fractured strength surface (frictional shear stresses) like in an ideal

plastic material, while the subsequent unloading shear stress-strain path follows reduced shear moduli to the zero shear stress and strain state for both τ_{bc} and τ_{ca} components.

The post failure behavior for the matrix crack in the a-c plane due to f_4 is modeled in the same fashion as that in the a-b plane as described above. In this case, when failure occurs, $S_{ab}^{(0)}$ and $S_{bc}^{(0)}$ are reduced to zero instantaneously. The post fracture response is then governed by failure criterion of f_5 with $S_{ab}^{(0)} = 0$ and $S_{bc}^{(0)} = 0$. For tensile mode, $\sigma_b > 0$, σ_b, τ_{ab} and τ_{bc} are zero. For compressive mode, $\sigma_b < 0$, σ_b is assumed to be elastic, while τ_{ab} and τ_{bc} 'slide' on the fracture strength surface as in an ideal plastic material, and the unloading path follows reduced shear moduli to the zero shear stress and strain state. It should be noted that τ_{bc} is governed by both the failure functions and should lie within or on each of these two strength surfaces.

Fabric Lamina Model

The fiber failure criteria of Hashin for a unidirectional layer are generalized to characterize the fiber damage in terms of strain components for a plain weave layer. The fill and warp fiber tensile/shear failure are given by the quadratic interaction between the associated axial and shear stresses, i.e.

$$f_6 = \left(\frac{\langle \sigma_a \rangle}{S_{aT}} \right)^2 + \frac{\left(\tau_{ab}^2 + \tau_{ca}^2 \right)}{S_{aFS}^2} - 1 = 0$$

$$f_7 = \left(\frac{\langle \sigma_b \rangle}{S_{bT}} \right)^2 + \frac{\left(\tau_{ab}^2 + \tau_{bc}^2 \right)}{S_{bFS}^2} - 1 = 0 \qquad (19.161.7)$$

where S_{aT} and S_{bT} are the axial tensile strengths in the fill and warp directions, respectively, and S_{aFS} and S_{bFS} are the layer shear strengths due to fiber shear failure in the fill and warp directions. These failure criteria are applicable when the associated σ_a or σ_b is positive. It is assumed S_{aFS} = SFS, and

$$S_{bFS} = SFS * S_{bT} / S_{aT} . \qquad (19.161.8)$$

When σ_a or σ_b is compressive, it is assumed that the in-plane compressive failure in both the fill and warp directions are given by the maximum stress criterion, i.e.

$$f_8 = \left[\frac{\langle \sigma_a^{'} \rangle}{S_{aC}} \right]^2 - 1 = 0, \quad \sigma_a^{'} = -\sigma_a + \langle -\sigma_c \rangle$$

$$f_9 = \left[\frac{\langle \sigma_b^{'} \rangle}{S_{bC}} \right]^2 - 1 = 0, \quad \sigma_b^{'} = -\sigma_b + \langle -\sigma_c \rangle \qquad (19.161.9)$$

where S_{aC} and S_{bC} are the axial compressive strengths in the fill and warp directions, respectively. The crush failure under compressive pressure is

$$f_{10}=\left(\frac{\langle p\rangle}{S_{FC}}\right)^2-1=0,\quad p=-\frac{\sigma_a+\sigma_b+\sigma_c}{3} \tag{19.161.10}$$

A plain weave layer can fail under in-plane shear stress without the occurrence of fiber breakage. This in-plane matrix failure mode is given by

$$f_{11}=\left(\frac{\tau_{ab}}{S_{ab}}\right)^2-1=0 \tag{19.161.11}$$

where S_{ab} is the layer shear strength due to matrix shear failure.

Another failure mode, which is due to the quadratic interaction between the thickness stresses, is expected to be mainly a matrix failure. This through the thickness matrix failure criterion is

$$f_{12}=S^2\left\{\left(\frac{\langle\sigma_c\rangle}{S_{cT}}\right)^2+\left(\frac{\tau_{bc}}{S_{bc}}\right)^2+\left(\frac{\tau_{ca}}{S_{ca}}\right)^2\right\}-1=0 \tag{19.161.12}$$

where S_{cT} is the through the thickness tensile strength, and S_{bc}, and S_{ca} are the shear strengths assumed to depend on the compressive normal stress s_C, i.e.,

$$\begin{Bmatrix}S_{ca}\\S_{bc}\end{Bmatrix}=\begin{Bmatrix}S_{ca}^{(0)}\\S_{bc}^{(0)}\end{Bmatrix}+\tan(\varphi)\langle-\sigma_c\rangle \tag{19.161.13}$$

When failure predicted by this criterion occurs within elements that are adjacent to the ply interface, the failure plane is expected to be parallel to the layering planes, and, thus, can be referred to as the delamination mode. Note that a scale factor S is introduced to provide better correlation of delamination area with experiments. The scale factor S can be determined by fitting the analytical prediction to experimental data for the delamination area.

Similar to the unidirectional model, when fiber tensile/shear failure is predicted in a layer by f_6 or f_7, the load carrying capacity of that layer in the associated direction is completely eliminated. For compressive fiber failure due to f_8 or f_9, the layer is assumed to carry a residual axial load in the failed direction, while the load carrying capacity transverse to the failed direction is assumed unchanged. When the compressive axial stress in a layer reaches the compressive axial strength S_{aC} or S_{bC}, the axial layer stress is assumed to be reduced to the residual strength S_{aRC} or S_{bRC} where $S_{aRC}=SFFC*S_{aC}$ and $S_{bRC}=SFFC*S_{bC}$. The axial stress is assumed to remain constant, i.e., $\sigma_a=-S_{aCR}$ or $\sigma_b=-S_{bCR}$, for continuous compressive loading, while the subsequent unloading curve follows a reduced axial modulus. When the fiber crush

failure has occurred, the material is assumed to behave elastically for compressive pressure, p > 0, and to carry no load for tensile pressure, p < 0.

When the in-plane matrix shear failure is predicted by f_{11} the axial load carrying capacity within a failed element is assumed unchanged, while the in-plane shear stress is assumed to be reduced to zero.

For through the thickness matrix (delamination) failure given by equations f_{12}, the in-plane load carrying capacity within the element is assumed to be elastic, while the strength values for the tensile mode, $S_{ca}^{(0)}$ and $S_{bc}^{(0)}$, are set to zero. For tensile mode, σ_C >0, the through the thickness stress components are reduced to zero. For compressive mode, σ_C <0, σ_C is assumed to be elastic, while τ_{bc} and τ_{ca} 'slide' on the fracture strength surface as in an ideal plastic material, and the unloading path follows reduced shear moduli to the zero shear stress and strain state.

The effect of strain-rate on the layer strength values of the fiber failure modes is modeled by the strain-rate dependent functions for the strength values $\{S_{RT}\}$ as

$$\{S_{RT}\} = \{S_0\}\left(1 + C_{rate1} \ln \frac{\{\dot{\bar{\varepsilon}}\}}{\dot{\varepsilon}_0}\right) \tag{19.161.14}$$

$$\{S_{RT}\} = \begin{Bmatrix} S_{aT} \\ S_{aC} \\ S_{bT} \\ S_{bC} \\ S_{FC} \\ S_{FS} \end{Bmatrix} \text{ and } \{\dot{\bar{\varepsilon}}\} = \begin{Bmatrix} |\dot{\varepsilon}_a| \\ |\dot{\varepsilon}_a| \\ |\dot{\varepsilon}_b| \\ |\dot{\varepsilon}_b| \\ |\dot{\varepsilon}_c| \\ \left(\dot{\varepsilon}_{ca}^2 + \dot{\varepsilon}_{bc}^2\right)^{1/2} \end{Bmatrix} \tag{19.161.15}$$

where C_{rate} is the strain-rate constants, and $\{S_0\}$ are the strength values of $\{S_{RT}\}$ at the reference strain-rate $\dot{\varepsilon}_0$.

Damage Model

The damage model is a generalization of the layer failure model of Material 161 by adopting the MLT damage mechanics approach, Matzenmiller et al. [1995], for characterizing the softening behavior after damage initiation. Complete model description is given in Yen [2001]. The damage functions, which are expressed in terms of ply level engineering strains, are converted from the above failure criteria of fiber and matrix failure modes by neglecting the Poisson's effect. Elastic moduli reduction is expressed in terms of the associated damage parameters ϖ_i:

$$E_i^{'} = (1 - \varpi_i) E_i \tag{19.161.16}$$

$$\varpi_i = 1 - \exp\left(-r_i^{m_i} / m_i\right) \quad r_i \geq 0 \quad i = 1,...,6$$

where E_i are the initial elastic moduli, $E_i^{'}$ are the reduced elastic moduli, r_i are the damage thresholds computed from the associated damage functions for fiber damage, matrix damage and delamination, and m_i are material damage parameters, which are currently assumed to be independent of strain-rate. The damage function is formulated to account for the overall nonlinear elastic response of a lamina including the initial 'hardening' and the subsequent softening beyond the ultimate strengths.

In the damage model (Material 162), the effect of strain-rate on the nonlinear stress-strain response of a composite layer is modeled by the strain-rate dependent functions for the elastic moduli $\{E_{RT}\}$ as

$$\{E_{RT}\} = \{E_0\}\left(1 + \{C_{rate}\}\ln\frac{\{\dot{\bar{\varepsilon}}\}}{\dot{\varepsilon}_0}\right) \tag{19.161.17}$$

$$\{E_{RT}\} = \begin{Bmatrix} E_a \\ E_b \\ E_c \\ G_{ab} \\ G_{bc} \\ G_{ca} \end{Bmatrix}, \{\dot{\bar{\varepsilon}}\} = \begin{Bmatrix} |\dot{\varepsilon}_a| \\ |\dot{\varepsilon}_b| \\ |\dot{\varepsilon}_c| \\ |\dot{\varepsilon}_{ab}| \\ |\dot{\varepsilon}_{bc}| \\ |\dot{\varepsilon}_{ca}| \end{Bmatrix} \text{ and } \{C_{rate}\} = \begin{Bmatrix} C_{rate2} \\ C_{rate2} \\ C_{rate4} \\ C_{rate3} \\ C_{rate3} \\ C_{rate3} \end{Bmatrix}$$

where $\{C_{rate}\}$ are the strain-rate constants. $\{E_0\}$ are the modulus values of $\{E_{RT}\}$ at the reference strain-rate $\dot{\varepsilon}_0$.

Material Model 163: Modified Crushable Foam

The volumetric strain is defined in terms of the relative volume, V, as:

$$\gamma = 1.\text{-}V$$

The relative volume is defined as the ratio of the current to the initial volume. In place of the effective plastic strain in the D3PLOT database, the integrated volumetric strain is output. This material is an extension of material 63, *MAT_CRUSHABLE_FOAM. It allows the yield stress to be a function of both volumetric strain rate and volumetric strain. Rate effects are accounted for by defining a table of curves using *DEFINE_TABLE. Each curve defines the yield stress versus volumetric strain for a different strain rate. The yield stress is obtained by interpolating between the two curves that bound the strain rate.

To prevent high frequency oscillations in the strain rate from causing similar high frequency oscillations in the yield stress, a modified volumetric strain rate is used when interpolating to obtain the yield stress. The modified strain rate is obtained as follows. If NCYCLE is >1, then the modified strain rate is obtained by a time average of the actual strain rate over NCYCLE solution cycles. For SRCLMT>0, the modified strain rate is capped so that during each cycle, the modified strain rate is not permitted to change more than SRCLMT multiplied by the solution time step.

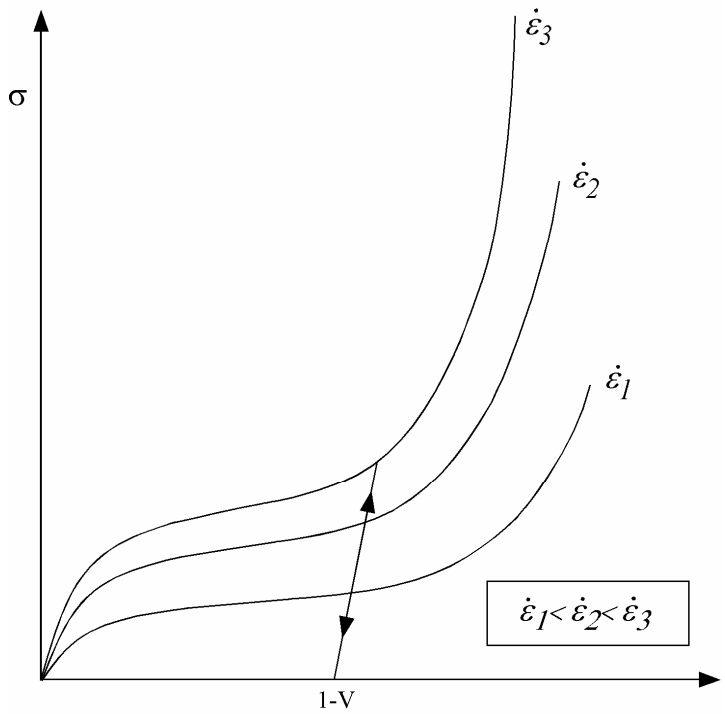


Figure 19.163.1. Rate effects are defined by a family of curves giving yield stress versus volumetric strain.

Material Model 164: Brain Linear Viscoelastic

The shear relaxation behavior is described by the Maxwell model as:

$$G(t) = G + (G_0 - G_\infty)e^{-\beta t} \tag{19.164.1}$$

A Jaumann rate formulation is used

$$\overset{\nabla}{\sigma}'_{ij} = 2\int_0^t G(t-\tau)\, D'_{ij}(\tau)dt \tag{19.164.2}$$

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:

$$\dot{s}_{ij} + \frac{1}{\tau}s_{ij} = \left(1+\delta_{ij}\right)\left(G_0 + \frac{G_\infty}{\tau}\right)\dot{e}_{ij} \tag{19.164.3}$$

The strain data as written to the LS-DYNA database may be used to predict damage, see [Bandak 1991].

Material Model 166: Moment Curvature Beam

Curvature rate can be decomposed into elastic part and plastic part:

$$\dot{\varepsilon} = \dot{\varepsilon}^e + \dot{\varepsilon}^p \Rightarrow \frac{\dot{\varepsilon}}{y} = \frac{\dot{\varepsilon}^p}{y} + \frac{\dot{\varepsilon}^p}{y} \Rightarrow \dot{\kappa} = \dot{\kappa}^e + \kappa^p \tag{19.166.1}$$

Moment rate is the product of elastic bending stiffness and elastic curvature:

$$\begin{aligned} \dot{M} &= \int_A \dot{\sigma} y dA = \int_A E^e \dot{\varepsilon}^e y dA = \int_A E^e \dot{\kappa}^e y^2 dA = \int_A E^e \left(\dot{\kappa} - \dot{\kappa}^p\right) y^2 dA \\ &= E^e \left(\dot{\kappa} - \dot{\kappa}^p\right) \int_A y^2 dA = \left(EI\right)^e \left(\dot{\kappa} - \dot{\kappa}^p\right) \end{aligned} \tag{19.166.2}$$

Plastic flow rule: $\psi = |M|$ (Isotropic hardening)

$$\dot{\kappa}^p = \dot{\lambda} \frac{\partial \psi}{\partial M} = \dot{\lambda} \operatorname{sign}\left(M\right), \ \dot{\bar{\kappa}}^p = \sqrt{\dot{\kappa}^p \dot{\kappa}^p} = \dot{\lambda} \tag{19.166.3}$$

Yield condition:

$$f = |M| - M_Y\left(\bar{\kappa}^p\right) = 0 \tag{19.166.4}$$

Loading and unloading conditions:

$$\dot{\lambda} \geq 0, \ f \leq 0, \dot{\lambda} f = 0 \tag{19.166.5}$$

Consistency condition:

$$\begin{aligned} &\dot{f} = 0 \Rightarrow \dot{M} \operatorname{sign}\left(M\right) - \frac{\partial M_Y}{\partial \bar{\kappa}^p} \dot{\bar{\kappa}}^p = 0 \\ &\Rightarrow \dot{\lambda} \equiv \dot{\bar{\kappa}}^p = \frac{M \operatorname{sign}\left(M\right)}{\left(EI\right)^p} = \frac{\left(EI\right)^e}{\left(EI\right)^p} \left(\dot{\kappa} - \dot{\kappa}^p\right) \operatorname{sign}\left(M\right) = \frac{\left(EI\right)^e}{\left(EI\right)^p} \left[\dot{\kappa} - \dot{\lambda} \operatorname{sign}\left(M\right)\right] \operatorname{sign}\left(M\right) \\ &\Rightarrow \dot{\lambda} \equiv \dot{\bar{\kappa}} = \frac{\left(EI\right)^e \dot{\kappa} \operatorname{sign}\left(M\right)}{\left(EI\right)^p + \left(EI\right)^e} \end{aligned}$$

Moment rate is also the product of tangential bending stiffness and total curvature:

$$\dot{M} = \left(EI\right)^{ep} \dot{\kappa} \tag{19.166.6}$$

Elastic, plastic, and tangential stiffnesses are obtained from user-defined curves:

$$(EI)^{ep} = \frac{dM}{d\kappa}, (EI)^{p} = \frac{dM}{d\bar{\kappa}^{p}}. \quad (19.166.7)$$

Both are obtained from user-defined curves.

$$(EI)^{e} = \frac{(EI)^{ep}(EI)^{p}}{(EI)^{p} - (EI)^{ep}} \quad (19.166.8)$$

For Torsion-Twist, simply replace *M* by *T*, κ by β, (EI) by (GJ). For Force-Strain, simply replace *M* by *N*, κ by ε, (EI) by (EA).

Material Model 169: Arup Adhesive

The through-thickness direction is identified from the smallest dimension of each element. It is expected that this dimension will be much smaller than in-plane dimensions (typically 2mm compared with 10mm)

In-plane stresses are set to zero: it is assumed that the stiffness and strength of the substrate is large compared with that of the adhesive, given the relative thicknesses.

If the substrate is modeled with shell elements, it is expected that these will lie at the mid-surface of the substrate geometry. Therefore the solid elements representing the adhesive will be thicker than the actual bond.

The yield and failure surfaces are treated as a power-law combination of direct tension and shear across the bond:

$$\left(\sigma/\sigma_{max}\right)^{PWRT} + \left(\tau/\tau_{max}\right)^{PWRS} = 1.0 \quad \text{at yield}$$

The stress-displacement curves for tension and shear are shown in the diagrams below. In both cases, Gc is the area under the curve.

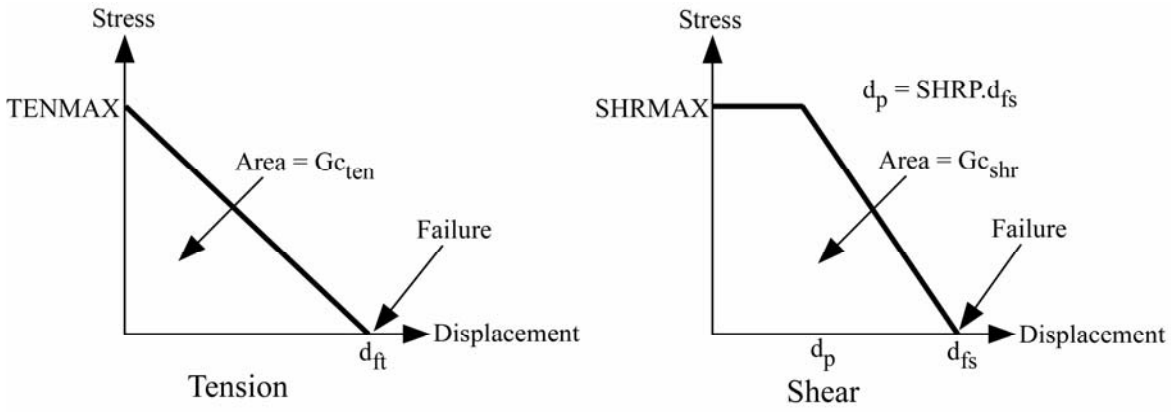


Figure 19.169.1

Because of the algorithm used, yielding in tension across the bond does not require strains in the plane of the bond – unlike the plasticity models, plastic flow is not treated as volume-conserving.

The Plastic Strain output variable has a special meaning:

$0 < ps < 1$: ps is the maximum value of the yield function experienced since time zero

$1 < ps < 2$: the element has yielded and the strength is reducing towards failure – yields at ps=1, fails at ps=2.

Material Model 170: Resultant Anisotropic

The in-plane elastic matrix for in-plane plane stress behavior is given by:

$$C_{in\ plane} = \begin{bmatrix} Q_{11p} & Q_{12p} & 0 & 0 & 0 \\ Q_{12p} & Q_{22p} & 0 & 0 & 0 \\ 0 & 0 & Q_{44p} & 0 & 0 \\ 0 & 0 & 0 & Q_{55p} & 0 \\ 0 & 0 & 0 & 0 & Q_{66p} \end{bmatrix} \tag{19.170.1}$$

The terms Q_{ijp} are defined as:

$$\begin{aligned} Q_{11p} &= \frac{E_{11p}}{1-\nu_{12p}\nu_{21p}} \\ Q_{22p} &= \frac{E_{22p}}{1-\nu_{12p}\nu_{21p}} \\ Q_{12p} &= \frac{\nu_{12p}E_{11p}}{1-\nu_{12p}\nu_{21p}} \\ Q_{44p} &= G_{12p} \\ Q_{55p} &= G_{23p} \\ Q_{66p} &= G_{31p} \end{aligned} \tag{19.170.2}$$

The elastic matrix for bending behavior is given by:

$$C_{bending} = \begin{bmatrix} Q_{11b} & Q_{12b} & 0 \\ Q_{12b} & Q_{22b} & 0 \\ 0 & 0 & Q_{44b} \end{bmatrix} \tag{19.170.3}$$

The terms Q_{ijb} are similarly defined.

Material Model 175: Viscoelastic Maxwell

Rate effects are taken into accounted through linear viscoelasticity by a convolution integral of the form:

$$\sigma_{ij} = \int_0^t g_{ijkl}\left(t-\tau\right)\frac{\partial \varepsilon_{kl}}{\partial \tau}d\tau \qquad (19.175.1)$$

where $g_{ijkl}\left(t-\tau\right)$ is the relaxation function for different stress measures. This stress is added to the stress tensor determined from the strain energy functional.

If we wish to include only simple rate effects, the relaxation function is represented by six terms from the Prony series:

$$g\left(t\right) = \sum_{m=1}^{N} G_m e^{-\beta_m t} \qquad (19.175.2)$$

We characterize this in the input by shear moduli, G_i, and the decay constants, β_i. An arbitrary number of terms, up to 6, may be used when applying the viscoelastic model.

For volumetric relaxation, the relaxation function is also represented by the Prony series in terms of bulk moduli:

$$k\left(t\right) = \sum_{m=1}^{N} K_m e^{-\beta_{k_m} t} \qquad (19.175.3)$$

The Arrhenius and Williams-Landau-Ferry (WLF) shift functions account for the effects of the temperature on the stress relaxation. A scaled time, *t'*,

$$t' = \int_0^t \Phi(T)dt \qquad (19.175.4)$$

is used in the relaxation function instead of the physical time. The Arrhenius shift function is

$$\Phi(T) = \exp(-A\{\frac{1}{T} - \frac{1}{T_{REF}}\}) \qquad (19.175.5)$$

and the Williams-Landau-Ferry shift function is

$$\Phi(T) = \exp(-A\frac{T - T_{REF}}{B + T - T_{REF}}) \qquad (19.175.6)$$

Material Model 176: Quasilinear Viscoelastic

The equations for this model are given as:

$$\sigma(t) = \int_0^t G(t-\tau)\frac{\partial \sigma_\varepsilon\left[\varepsilon(\tau)\right]}{\partial \varepsilon}\frac{\partial \varepsilon}{\partial \tau}d\tau$$

$$G(t) = \sum_{i=1}^{n} G_i e^{-\beta t}$$

$$\sigma_\varepsilon(\varepsilon) = \sum_{i=1}^{k} C_i \varepsilon^i \qquad (19.176.1)$$

where G is the shear modulus. In place of the effective plastic strain in the D3PLOT database, the effective strain is output:

$$\varepsilon^{effective} = \sqrt{\frac{2}{3}\varepsilon_{ij}\varepsilon_{ij}} \qquad (19.176.2)$$

The polynomial for instantaneous elastic response should contain only odd terms if symmetric tension-compression response is desired.

Material Models 177 and 178: Hill Foam and Viscoelastic Hill Foam

19.177.1 Hyperelasticity Using the Principal Stretch Ratios

Material types 177 and 178 in LS-DYNA are highly compressible Ogden models combined with viscous stress contributions. The latter model also allows for an additive viscoelastic stress contribution. As for the rate independent part, the constitutive law is determined by a strain energy function that is expressed in terms of the principal stretches, i.e., $W = W(\lambda_1, \lambda_2, \lambda_3)$. To obtain the Cauchy stress σ_{ij}, as well as the constitutive tensor of interest, C^{TC}_{ijkl}, they are first calculated in the principal basis after which they are transformed back to the "base frame", or standard basis. The complete set of formulas is given by Crisfield [1997] and is for the sake of completeness recapitulated here.

The principal Kirchhoff stress components are given by

$$\tau^E_{ii} = \lambda_i \frac{\partial W}{\partial \lambda_i} \text{ (no sum)}$$

that are transformed to the standard basis using the standard formula

$$\tau_{ij} = q_{ik} q_{jl} \tau^E_{kl}.$$

The q_{ij} are the components of the orthogonal tensor containing the eigenvectors of the principal basis. The Cauchy stress is then given by

$$\sigma_{ij} = J^{-1}\tau_{ij},$$

where $J = \lambda_1\lambda_2\lambda_3$ is the relative volume change.

The constitutive tensor that relates the rate of deformation to the Truesdell (convected) rate of Kirchhoff stress in the principal basis can be expressed as

$$\left.\begin{aligned} C_{iijj}^{TKE} &= \lambda_j \frac{\partial \tau_{ii}^E}{\partial \lambda_j} - 2\tau_{ii}^E \delta_{ij} \\ C_{ijij}^{TKE} &= \frac{\lambda_j^2 \tau_{ii}^E - \lambda_i^2 \tau_{jj}^E}{\lambda_i^2 - \lambda_j^2}, \quad i \neq j, \lambda_i \neq \lambda_j \\ C_{ijij}^{TKE} &= \frac{\lambda_i}{2}\left(\frac{\partial \tau_{ii}^E}{\partial \lambda_i} - \frac{\partial \tau_{ii}^E}{\partial \lambda_j}\right), \quad i \neq j, \lambda_i = \lambda_j \end{aligned}\right\} \text{ (no sum)}$$

These components are transformed to the standard basis according to

$$C_{ijkl}^{TK} = q_{ip} q_{jq} q_{kr} q_{ls} C_{pqrs}^{TKE}, \tag{19.177.1}$$

and finally the constitutive tensor relating the rate of deformation to the Truesdell rate of Cauchy stress is obtained through

$$C_{ijkl}^{TC} = J^{-1} C_{ijkl}^{TK}. \tag{19.177.2}$$

19.177.2 Hill's Strain Energy Function

The strain energy function for materials 177 and 178 is given by

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

where n, μ_m and α_m are material parameters. To apply the formulas in the previous section, we require

$$\tau_{ii}^E = \lambda_i \frac{\partial W}{\partial \lambda_i} = \sum_{m=1}^{n} \frac{\mu_m}{J}(\lambda_i^{\alpha_m} - J^{-n\alpha_m}).$$

Proceeding with the constitutive tensor, we have

$$\lambda_j \frac{\partial \tau_{ii}^E}{\partial \lambda_j} = \sum_{m=1}^{n} \mu_m \alpha_m (\lambda_i^{\alpha_m} \delta_{ij} + nJ^{-n\alpha_m}).$$

In addition to the hyperelastic stress described above, a viscous stress is added. Converting to Voigt notation, this stress can be written

$$\boldsymbol{\sigma} = \mathbf{C}\mathbf{D}\,, \tag{19.177.3}$$

where $\boldsymbol{\sigma}$ denotes Cauchy stress, $\mathbf{D}$ is the rate-of-deformation and $\mathbf{C}$ is an isotropic constitutive matrix representing the viscosity. In element m, the constitutive matrix depends on the element deformation according to

$$\mathbf{C} = \frac{d_m}{J}\mathbf{C}_0\,,$$

where d_m is the diameter[1] of element m and $\mathbf{C}_0$ is a constitutive matrix that depends only on the material parameters. The stress contribution to the internal force can be written

$$\mathbf{f}^{\text{int}} = \int_{\Omega_m} \mathbf{B}^T\boldsymbol{\sigma}\, d\Omega_m$$

and the corresponding material time derivative is

$$\dot{\mathbf{f}}^{\text{mat}} = \int_{\Omega_m} \mathbf{B}^T\boldsymbol{\sigma}^{\nabla T}\, d\Omega_m\,. \tag{19.177.4}$$

Here Ω_m is the current configuration of element m, $\mathbf{B}$ is the strain-displacement matrix and ∇T denotes the Truesdell rate of Cauchy stress. The aim is to identify the material tangent modulus through

$$\dot{\mathbf{f}}^{\text{mat}} = \int_{\Omega_m} \mathbf{B}^T\mathbf{C}^{\text{mat}}\mathbf{B}\, d\Omega_m\dot{\mathbf{u}} \tag{19.177.5}$$

for the viscous stress with $\dot{\mathbf{u}}$ being the nodal velocity. The Truesdell rate of the viscous stress can be written

$$\boldsymbol{\sigma}^{\nabla T} = \dot{\mathbf{C}}\mathbf{D} + \mathbf{C}\dot{\mathbf{D}} + tr(\mathbf{D})\boldsymbol{\sigma} - \mathbf{L}\boldsymbol{\sigma} - \boldsymbol{\sigma}\mathbf{L}^T\,, \tag{19.177.6}$$

where $\mathbf{L}$ is the velocity gradient. The terms on the right hand side can be treated as follows. For the first term, we can assume that $d_m \propto J^{1/3}$ and then approximate

$$\dot{\mathbf{C}} = -\frac{2}{3}tr\left(\mathbf{D}\right)\mathbf{C}\,. \tag{19.177.7}$$

Using Equation (19.177.3), Equation (19.177.4), the first term on the right hand side of Equation (19.177.6), Equation (19.177.7) and the expression

[1] Experiments indicate that d_m is the smallest dimension of the element.

$$\mathbf{D} = \mathbf{B}\dot{\mathbf{u}}, \tag{19.177.8}$$

a material tangent modulus contribution can be identified in Equation (19.177.5) as

$$-\frac{2}{3}\boldsymbol{\sigma}\boldsymbol{\delta}^T$$

where $\boldsymbol{\delta}$ denotes the identity matrix in Voigt notation.

For the second term in Equation (19.177.6), we differentiate Equation (19.177.8) to see that

$$\dot{\mathbf{D}} = \dot{\mathbf{B}}\dot{\mathbf{u}} + \mathbf{B}\ddot{\mathbf{u}}. \tag{19.177.9}$$

Post-poning the treatment of the first term, the second of these two terms can be treated easily as this gives the following contribution to the material time derivative

$$\int_{\Omega_m} \mathbf{B}^T\mathbf{C}\mathbf{B}d\Omega_m\dot{\mathbf{u}}\frac{\gamma}{\beta\Delta t},$$

where γ and β are parameters in the Newmark scheme and Δt is the time step. From this expression, a material tangent modulus can through Equation (19.177.5) be identified as

$$\mathbf{C}^{\text{mat}} = \frac{\gamma}{\beta\Delta t}\mathbf{C}. \tag{19.177.10}$$

The third term in Equation (19.177.6) contributes to the material tangent modulus as

$$\boldsymbol{\sigma}\boldsymbol{\delta}^T, \tag{19.177.11}$$

resulting in a material tangent modulus given so far by

$$\frac{\gamma}{\beta\Delta t}\mathbf{C} + \frac{1}{3}\boldsymbol{\sigma}\boldsymbol{\delta}^T. \tag{19.177.12}$$

19.177.3 Viscous Stress

From the remaining terms, i.e., the last two terms in Equation (19.177.6) and the first term in Equation (19.177.9), we see it impossible to identify contributions to a material tangent modulus. We believe that these terms must be treated in some other manner. We are thus left with two choices, either to approximate these terms within the existing framework or to attempt a thorough implementation of the correct tangent stiffness using a different, and probably demanding, approach. We reason as follows.

Since this stress contribution is viscous and proportional to the meshsize, it is our belief that it serves as a stabilizing stress in the occurrence of a coarse mesh and/or large deformation rates, and really has little or nothing to do with the actual material models. If only the simulation

process is slow (which it often is in an implicit analysis) and/or the mesh is sufficiently fine, this stress should be negligable compared to the other stress(es). With this in mind, we feel that it is not crucial to derive an exact tangent for this stress but we can be satisfied with an approximation. Even if attempting a more thorough derivation of the tangent stiffness, we would most certainly have to make approximations along the way. Hence we do not see this as an attractive approach.

In the implementation we have simply neglected all terms involving stresses since the experience from earlier work is that such terms generally have a negative effect on the tangent if they are not absolutely correct. In addition, most of the terms involving stresses contribute to a nonsymmetric tangent stiffness, which cannot be supported by LS-DYNA at the moment. Hence the material tangent modulus for the viscous stress is given by Equation (19.177.10). We are aware of that this may be a crude approximation, and if experiments show that it is a poor one, we will take a closer look at it.

In material type 178, the viscous stress acts only in the direction of the principal stretches and in compression. With $\mathbf{C}$ being an isotropic tensor, we evaluate the tangent stiffness modulus in the principal basis according to Equation (19.177.10), modify it to account for the mentioned conditions and then transform it back to the global frame of reference.

19.177.4 Viscoelastic Stress Contribution

For material 178, an optional viscoelastic stress contribution can be added. The evolution of this stress in time can be stated as

$$\sigma_{ij}^{\nabla} = \sum_{m=1}^{12} 2G_m s_{ij}^{m\nabla} \qquad (19.177.13)$$

where

$$s_{ij}^{m\nabla} = D_{ij} - \beta_m s_{ij}^{m}. \qquad (19.177.14)$$

Here G_m and β_m are material constants, and D_{ij} is the rate-of-deformation tensor. Referring to Borrvall [2002], we state that the tangent stiffness modulus for this stress contribution can be written

$$C_{ijkl} = \sum_{m=1}^{12} G_m(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}). \qquad (19.177.15)$$

Just as for the viscous stress, this stress acts only in the direction of the principal stretches. Hence the tangent modulus is formed in the principal basis, modified to account for this condition and then transformed back to the global frame of reference.

19.177.5 Material Tangent Modulus for the Fully Integrated Brick

To avoid locking tendencies for the fully integrated brick element in LS-DYNA, the stress is modified as

$$\boldsymbol{\sigma}_{S/R} = \boldsymbol{\sigma} + (p - \overline{p})\mathbf{I} \qquad (19.177.16)$$

where p is the pressure and $\bar{p}$ is the mean pressure in the element. This affects the tangent stiffness since one has to take into account that the pressure is constant in the element. Deriving the material time derivative of the internal force results in

$$\dot{\mathbf{f}}^{\text{mat}} = \int_{\Omega_m} \mathbf{B}^T \mathbf{C}\mathbf{B} d\Omega_m \dot{\mathbf{u}} + \int_{\Omega_m} (p-\bar{p})\mathbf{B}^T(\mathbf{I}\otimes\mathbf{I})\mathbf{B} d\Omega_m \dot{\mathbf{u}} - 2\int_{\Omega_m} (p-\bar{p})\mathbf{B}^T\mathbf{B} d\Omega_m \dot{\mathbf{u}} + \int_{\Omega_m} (\dot{p}-\dot{\bar{p}})\mathbf{B}^T d\Omega_m$$

To implement this tangent, the last term is the most difficult to deal with as it involves the time derivative (or variation) of the pressure. For certain types of material models, for instance material type 77 in LS-DYNA, the pressure is a function of the relative volume

$$p = p(J) \tag{19.177.17}$$

and with the approximation

$$\bar{p} = p(\bar{J}) \tag{19.177.18}$$

the last term can be evaluated to

$$\int_{\Omega_m} Jp'(J)\mathbf{B}^T(\mathbf{I}\otimes\mathbf{I})\mathbf{B} d\Omega_m \dot{\mathbf{u}} - \int_{\Omega_m} \bar{J}p'(\bar{J})\bar{\mathbf{B}}^T(\mathbf{I}\otimes\mathbf{I})\bar{\mathbf{B}} d\Omega_m \dot{\mathbf{u}} \tag{19.177.19}$$

and a symmetric tangent stiffness can quite easily be implemented. We have here used $\bar{J}$ and $\bar{\mathbf{B}}$ for the mean values of J and $\mathbf{B}$, respectively. For other types of material models, such as the ones described in this document or material type 27 in LS-DYNA, the expression for the pressure is more complicated. A characterizing feature is that a non-zero pressure can occur under constant volume. This will in general complicate the implementation of the last term and will also contribute to a non-symmetric tangent stiffness that cannot be handled in LS-DYNA at the moment. For material 27, neglecting this had a tremendous impact on the performance of the implicit solution procedure, (see material type 27). For the current material models, it seems to be of less importance, and we believe that this is due to the higher compressibility allowed.

Material Models 179 and 180: Low Density Synthetic Foam

Material types 179 and 180 in LS-DYNA are highly compressible synthetic foam models with no Poisson's ratio effects combined with an optional visco-elastic and a stabilizing viscous stress constribution. The tensile behavior of the materials is linear where the stress cannot exceed a user prescribed cutoff stress. In compression the materials show a hysteresis on unloading similar to material 57. In addition, the first load cycle damages the material so that the stress level on reloading is significantly reduced. For material 179 the damage is isotropic while it is orthotropic for material 180.

19.179.1 Hyperelasticity Using the Principal Stretch Ratios

As for the rate independent part of the stress, the constitutive law is mainly determined by a strain energy function that is expressed in terms of the principal stretches, i.e.,

$W = W(\lambda_1, \lambda_2, \lambda_3)$. To obtain the Cauchy stress σ_{ij}, as well as the constitutive tensor of interest, C^{TC}_{ijkl}, they are first calculated in the principal basis after which they are transformed back to the "base frame", or standard basis. The complete set of formulas is given by Crisfield [1997] and is for the sake of completeness recapitulated here.

The principal Kirchhoff stress components are given by

$$\tau^E_{ii} = \lambda_i \frac{\partial W}{\partial \lambda_i} \text{ (no sum)} \tag{19.179.1}$$

that are transformed to the standard basis using the standard formula

$$\tau_{ij} = q_{ik} q_{jl} \tau^E_{kl} . \tag{19.179.2}$$

The q_{ij} are the components of the orthogonal tensor containing the eigenvectors of the principal basis. The Cauchy stress is then given by

$$\sigma_{ij} = J^{-1} \tau_{ij} , \tag{19.179.3}$$

where $J = \lambda_1 \lambda_2 \lambda_3$ is the relative volume change.

The constitutive tensor that relates the rate of deformation to the Truesdell (convected) rate of Kirchhoff stress can in the principal basis be expressed as

$$\begin{aligned} C^{TKE}_{iijj} &= \lambda_j \frac{\partial \tau^E_{ii}}{\partial \lambda_j} - 2\tau^E_{ii} \delta_{ij} \\ C^{TKE}_{ijij} &= \frac{\lambda_j^2 \tau^E_{ii} - \lambda_i^2 \tau^E_{jj}}{\lambda_i^2 - \lambda_j^2}, \quad i \neq j, \lambda_i \neq \lambda_j \quad \text{(no sum)} \\ C^{TKE}_{ijij} &= \frac{\lambda_i}{2} \left(\frac{\partial \tau^E_{ii}}{\partial \lambda_i} - \frac{\partial \tau^E_{ii}}{\partial \lambda_j} \right), \quad i \neq j, \lambda_i = \lambda_j \end{aligned} \tag{19.179.4}$$

These components are transformed to the standard basis according to

$$C^{TK}_{ijkl} = q_{ip} q_{jq} q_{kr} q_{ls} C^{TKE}_{pqrs} , \tag{19.179.5}$$

and finally the constitutive tensor relating the rate of deformation to the Truesdell rate of Cauchy stress is obtained through

$$C^{TC}_{ijkl} = J^{-1} C^{TK}_{ijkl} . \tag{19.179.6}$$

19.179.2 Strain Energy Function

The strain energy function for materials 179 and 180 is given by

$$W = \sum_{m=1}^{3} w(\lambda_m) \tag{19.179.7}$$

where

$$w(\lambda) = \begin{cases} s\left(\lambda - 1 - \dfrac{s}{2E}\right) & \text{if } \lambda \geq \dfrac{s}{E} + 1 \\ \dfrac{E}{2}(\lambda - 1)^2 & \text{if } 1 \leq \lambda < \dfrac{s}{E} + 1. \\ \displaystyle\int_1^{\lambda} f_s(1-\mu)d\mu & \text{otherwise} \end{cases} \tag{19.179.7}$$

Here s is the nominal tensile cutoff stress and E is the stiffness coefficient relating a change in principal stretch to a corresponding change in nominal stress. The function $f_s(\leq 0)$ gives the nominal compressive stress as a function of the strain in compression for the second and all subsequent load cycles and is supplied by the user. To apply the formulas in the previous section, we require

$$\tau_{ii}^E = \lambda_i \frac{\partial w}{\partial \lambda_i} = \begin{cases} s\lambda_i & \text{if } \lambda_i \geq \dfrac{s}{E} + 1 \\ E\lambda_i(\lambda_i - 1) & \text{if } 1 \leq \lambda_i < \dfrac{s}{E} + 1. \\ \lambda_i f_s(1-\lambda_i) & \text{otherwise} \end{cases} \tag{19.179.8}$$

Proceeding with the constitutive tensor, we have

$$\lambda_j \frac{\partial \tau_{ii}^E}{\partial \lambda_j} = \delta_{ij} \begin{cases} s\lambda_i & \text{if } \lambda_i \geq \dfrac{s}{E} + 1 \\ E\lambda_i(2\lambda_i - 1) & \text{if } 1 \leq \lambda_i < \dfrac{s}{E} + 1. \\ \lambda_i\left(f_s(1-\lambda_i) - \lambda_i f'_s(1-\lambda_i)\right) & \text{otherwise} \end{cases} \tag{19.179.9}$$

19.179.3 Modeling of the Hysteresis

The hyperelastic part of the Cauchy stress is scaled by a factor κ given by

$$\kappa = E / \bar{E} \tag{19.179.10}$$

where

$$E = \int J\boldsymbol{\sigma} : d\boldsymbol{\varepsilon} \tag{19.179.11}$$

is the stored energy in the material and

$$\bar{E} = E_{\max} \exp(-\beta(t-s)). \qquad (19.179.12)$$

Here s stands for the time point when E has its maximum $E_{\max}$ in the interval $[0,t]$. The factor κ is introduced to model the hysteresis that characterizes this material (and material 57). The decay coefficient β is introduced to get a reloading curve similar to the original loading curve. This factor κ is treated as a constant in the determination of the tangent stiffness matrix.

19.179.4 Viscous Stress

In addition to the hyperelastic stress described above, a viscous stress is added. Converting to Voigt notation, this stress can be written

$$\boldsymbol{\sigma} = \mathbf{C}\mathbf{D}, \qquad (19.179.13)$$

where $\boldsymbol{\sigma}$ denotes Cauchy stress, $\mathbf{D}$ is the rate-of-deformation and $\mathbf{C}$ is an isotropic constitutive matrix representing the viscosity. In element m, the constitutive matrix depends on the element deformation according to

$$\mathbf{C} = \frac{d_m}{J}\mathbf{C}_0, \qquad (19.179.14)$$

where d_m is the diameter[1] of element m and $\mathbf{C}_0$ is a constitutive matrix that depends only on the material parameters. Following material models 177 and 178 we use the following material tangent stiffness for this stress contribution

$$\mathbf{C}^{\text{mat}} = \frac{\gamma}{\beta\Delta t}\mathbf{C}. \qquad (19.179.15)$$

where γ and β are parameters in the Newmark scheme and Δt is the time step.

19.179.5 Viscoelastic Stress Contribution

An optional viscoelastic stress contribution can be added. The evolution of this stress in time can be stated as

$$\sigma_{ij}^{\nabla} = E_d s_{ij}^{\nabla} \qquad (19.179.16)$$

where

$$s_{ij}^{\nabla} = D_{ij} - \beta_1 s_{ij}. \qquad (19.179.17)$$

[1] Experiments indicate that d_m is the smallest dimension of the element.

Here E_d and β_1 are material constants, D_{ij} is the rate-of-deformation tensor and ∇ stands for an objective rate. Referring to material models 177 and 178, we state that the tangent stiffness modulus for this stress contribution can be written

$$C_{ijkl} = \frac{E_d}{2}(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) . \qquad (19.179.18)$$

This stress acts only in the direction of the principal stretches. Hence the tangent modulus is formed in the principal basis, modified to account for this condition and then transformed back to the global frame of reference.

19.179.6 Stress Corresponding to First Load Cycle

We define a contribution to the principal Kirchhoff stress as

$$^1\tau_{ii}^E = \lambda_i\{g_s(1-\lambda_i) - f_s(1-\lambda_i)\}\xi . \qquad (19.179.19)$$

When the damage is isotropic the factor ξ is given by

$$\xi = \max(0, 1 - \frac{\varepsilon_h}{0.0001 + \varepsilon_m}) \qquad (19.179.20)$$

where ε_h is the damage parameter that is initially zero and ε_m is the maximum compressive volumetric strain during the entire simulation thus far. Damage evolves when the material is in compression and unloads

$$\Delta\varepsilon_h = \begin{cases} 0 & \text{if } J \geq 1 \\ \max(0, \Delta J) & \text{otherwise} \end{cases} \qquad (19.179.21)$$

where J is the jacobian of the deformation. The first load cycle will result in a total stress that follows load curve g_s since there is no damage. After a complete load cycle, i.e., unloading has occurred, the material is completely damaged, i.e., $\varepsilon_h \approx \varepsilon_m$, and the nominal stress will for the second and subsequent load cycles be given by the load curve f_s.

In the orthotropic case the principal Kirchhoff stress contribution is instead given by

$$^1\tau_{ii}^E = \lambda_i\{g_s(1-\lambda_i) - f_s(1-\lambda_i)\}\xi_i . \qquad (19.179.22)$$

For the damage to be orthotropic we introduce a symmetric and positive definite damage tensor ε_h^{ij}. This tensor is initially the zero tensor corresponding to no damage. The evolution of damage begins with a half step Jaumann rotation of the tensor to maintain objectivity. After that the local increment is performed. As for the isotropic case, damage evolves in compression in combination with unloading. We introduce the local damage increment as

$$\Delta\varepsilon_{\text{loc}}^{ij} = \delta_{ij}\begin{cases} 0 & \text{if } \lambda_i \geq 1 \\ \max(0, \Delta\lambda_i) & \text{otherwise} \end{cases} \tag{19.179.23}$$

which is a diagonal tensor. The global damage tensor increment is given by

$$\Delta\varepsilon_h^{ij} = q_{ik} q_{jl} \Delta\varepsilon_{\text{loc}}^{kl} \tag{19.179.24}$$

which is used to increment the damage tensor ε_h^{ij}. The factor ξ_i is now given by

$$\xi_i = \max(0, 1 - \frac{q_{ki} q_{li} \varepsilon_h^{kl}}{0.0001 + \varepsilon_m}) \tag{19.179.25}$$

where the quantity ε_m in the orthotropic case is the maximum compressive principal strain in any direction during the simulation thus far. As for the isotropic case, the material is completely damaged after one load cycle and reloading will follow load curve f_s. In addition, the directions corresponding to no loading will remain unaffected.

The factors ξ and ξ_i are treated as constants in the determination of the tangent stiffness so the contribution is regarded as hyperelastic and follows the exposition given in Section 19.179.1.

The reason for not differentiating the coefficients κ, ξ and ξ_i is that they are always non-differentiable. Their changes depend on whether the material is loaded or unloaded, i.e., the direction of the load. Even if they were differentiable their contributions would occasionally result in a non-symmetric tangent stiffness matrix and any attempt to symmetrize this would probably destroy its properties. After all, we believe that the one-dimensional nature and simplicity of this foam will be enough for good convergence properties even without differentiating these coefficients.

Material Model 181: Simplified Rubber Foam

Material type 181 in LS-DYNA is a simplified "quasi"-hyperelastic rubber model defined by a single uniaxial load curve or by a family of curves at discrete strain rates. The term "quasi" is used because there is really no strain energy function for determining the stresses used in this model. However, for deriving the tangent stiffness matrix we use the formulas as if a strain energy function were present. In addition, a frequency independent damping stress is added to model the energy dissipation commonly observed in rubbers.

19.181.1 Hyperelasticity Using the Principal Stretch Ratios

A hyperelastic constitutive law is determined by a strain energy function that here is expressed in terms of the principal stretches, i.e., $W = W(\lambda_1, \lambda_2, \lambda_3)$. To obtain the Cauchy stress σ_{ij}, as well as the constitutive tensor of interest, C_{ijkl}^{TC}, they are first calculated in the principal basis after which they are transformed back to the "base frame", or standard basis. The complete set of formulas is given by Crisfield [1997] and is for the sake of completeness recapitulated here.

The principal Kirchhoff stress components are given by

$$\tau_{ii}^E = \lambda_i \frac{\partial W}{\partial \lambda_i} \text{ (no sum)} \tag{19.181.1}$$

that are transformed to the standard basis using the standard formula

$$\tau_{ij} = q_{ik} q_{jl} \tau_{kl}^E . \tag{19.181.2}$$

The q_{ij} are the components of the orthogonal tensor containing the eigenvectors of the principal basis. The Cauchy stress is then given by

$$\sigma_{ij} = J^{-1} \tau_{ij} , \tag{19.181.3}$$

where $J = \lambda_1 \lambda_2 \lambda_3$ is the relative volume change.

The constitutive tensor that relates the rate of deformation to the Truesdell (convected) rate of Kirchhoff stress can in the principal basis be expressed as

$$\begin{aligned} C_{iijj}^{TKE} &= \lambda_j \frac{\partial \tau_{ii}^E}{\partial \lambda_j} - 2\tau_{ii}^E \delta_{ij} \\ C_{ijij}^{TKE} &= \frac{\lambda_j^2 \tau_{ii}^E - \lambda_i^2 \tau_{jj}^E}{\lambda_i^2 - \lambda_j^2}, \quad i \neq j, \lambda_i \neq \lambda_j \qquad \text{(no sum)} \\ C_{ijij}^{TKE} &= \frac{\lambda_i}{2} \left(\frac{\partial \tau_{ii}^E}{\partial \lambda_i} - \frac{\partial \tau_{ii}^E}{\partial \lambda_j} \right), \quad i \neq j, \lambda_i = \lambda_j \end{aligned} \tag{19.181.4}$$

These components are transformed to the standard basis according to

$$C_{ijkl}^{TK} = q_{ip} q_{jq} q_{kr} q_{ls} C_{pqrs}^{TKE} , \tag{19.181.5}$$

and finally the constitutive tensor relating the rate of deformation to the Truesdell rate of Cauchy stress is obtained through

$$C_{ijkl}^{TC} = J^{-1} C_{ijkl}^{TK} . \tag{19.181.6}$$

19.181.2 Stress and Tangent Stiffness

The principal Kirchhoff stress is in material model 181 given by

$$\tau_{ii}^E = f(\lambda_i) + K(J-1) - \frac{1}{3} \sum_{k=1}^{3} f(\lambda_k) \tag{19.181.7}$$

where f is a load curve determined from uniaxial data (possibly at different strain rates). Furthermore, K is the bulk modulus and J is the relative volume change of the material. This

stress cannot be deduced from a strain energy function unless $f(\lambda) = E \ln \lambda$ for some constitutive parameter E. A consequence of this is that when using the formulas in the previous section the resulting tangent stiffness matrix is not necessarily symmetric. We remedy this by symmetrizing the formulas according to

$$\left(\lambda_j \frac{\partial \tau_{ii}^E}{\partial \lambda_j} \right)_{\text{symm}} = \frac{1}{2}\left(\lambda_j \frac{\partial \tau_{ii}^E}{\partial \lambda_j} + \lambda_i \frac{\partial \tau_{jj}^E}{\partial \lambda_i} \right) = KJ + \begin{cases} \frac{2}{3}\lambda_i f'(\lambda_i) & \text{if } i = j \\ -\frac{1}{6}\left(\lambda_i f'(\lambda_i) + \lambda_j f'(\lambda_j)\right) & \text{otherwise} \end{cases} \quad (19.181.8)$$

19.181.3 Two Remarks

The function f introduced in the previous section depends not only on the stretches but for some choices of input also on the strain rate. Strain rate effects complicate things for an implicit analyst and here one also has to take into account whether the material is in tension/compression or in a loading/unloading stage. We believe that it is of little importance to take into account the strain rate effects when deriving the tangent stiffness matrix and therefore this influence has been disregarded.

For the fully integrated brick element we have used the approach in material model 77 to account for the constant pressure when deriving the tangent stiffness matrix. Experiments have shown that this is crucial to obtain a decent implicit performance for nearly incompressible materials.

19.181.3 Modeling of the Frequency Independent Damping

An elastic-plastic stress $\boldsymbol{\sigma}_d$ is added to model the frequency independent damping properties of rubber. This stress is deviatoric and determined by the shear modulus G and the yield stress σ_Y. This part of the stress is updated incrementally as

$$\tilde{\boldsymbol{\sigma}}_d^{n+1} = \boldsymbol{\sigma}_d^n + 2G\mathbf{I}^{\text{dev}}\Delta\boldsymbol{\varepsilon} \quad (19.181.9)$$

where $\Delta\boldsymbol{\varepsilon}$ is the strain increment. The trial stress is then radially scaled (if necessary) to the yield surface according to

$$\boldsymbol{\sigma}_d^{n+1} = \tilde{\boldsymbol{\sigma}}_d^{n+1} \min(1, \frac{\sigma_Y}{\sigma_{\text{eff}}}) \quad (19.181.10)$$

where σ_{eff} is the effective von Mises stress for the trial stress $\tilde{\boldsymbol{\sigma}}_d^{n+1}$.

The elastic tangent stiffness contribution is given by

$$\mathbf{C}_d = 2G\mathbf{I}^{\text{dev}} \quad (19.181.11)$$

and if yield has occurred in the last time step the elastic-plastic tangent is used

$$\mathbf{C}_d = 2G\mathbf{I}^{\text{dev}} - \frac{3G}{\sigma_Y^2}\boldsymbol{\sigma}_d \otimes \boldsymbol{\sigma}_d \,. \quad (19.181.12)$$

Here $\mathbf{I}^{dev}$ is the deviatoric 4th order identity tensor.

Material Model 196: General Spring Discrete Beam

If TYPE=0, elastic behavior is obtained. In this case, if the linear spring stiffness is used, the force, *F*, is given by:

$$F = F_0 + K\Delta L + D\Delta\dot{L} \tag{19.196.1}$$

but if the load curve ID is specified, the force is then given by:

$$F = F_0 + K\ f\left(\Delta L\right)\left[1 + C1\cdot\Delta\dot{L} + C2\cdot\mathrm{sgn}\left(\Delta\dot{L}\right)\ln\left(\max\left\{1., \frac{\left|\Delta\dot{L}\right|}{DLE}\right\}\right)\right] + D\Delta\dot{L} + g\left(\Delta L\right)h\left(\Delta\dot{L}\right)$$

In these equations, ΔL is the change in length

$$\Delta L = current\ length - initial\ length$$

If TYPE=1, inelastic behavior is obtained. In this case, the yield force is taken from the load curve:

$$F^Y = F_y\left(\Delta L^{plastic}\right) \tag{19.196.2}$$

where $L^{plastic}$ is the plastic deflection. A trial force is computed as:

$$F^T = F^n + K\Delta\dot{L}\Delta t \tag{19.196.3}$$

and is checked against the yield force to determine F:

$$F = \begin{cases} F^Y\ if\ F^T > F^Y \\ F^T\ if\ F^T \le F^Y \end{cases} \tag{19.196.4}$$

The final force, which includes rate effects and damping, is given by:

$$F^{n+1} = F\cdot\left[1 + C1\cdot\Delta\dot{L} + C2\cdot\mathrm{sgn}\left(\Delta\dot{L}\right)\ln\left(\max\left\{1., \frac{\left|\Delta\dot{L}\right|}{DLE}\right\}\right)\right] + D\Delta\dot{L} + g\left(\Delta L\right)h\left(\Delta\dot{L}\right)$$

Unless the origin of the curve starts at (0,0), the negative part of the curve is used when the spring force is negative where the negative of the plastic displacement is used to interpolate, F_y. The positive part of the curve is used whenever the force is positive.

The cross sectional area is defined on the section card for the discrete beam elements, See *SECTION_BEAM. The square root of this area is used as the contact thickness offset if these elements are included in the contact treatment.

20. EQUATION OF STATE MODELS

LS-DYNA has 10 equation of state models which are described in this section.

1. Linear Polynomial
2. JWL High Explosive
3. Sack "Tuesday" High Explosive
4. Gruneisen
5. Ratio of Polynomials
6. Linear Polynomial With Energy Deposition
7. Ignition and Growth of Reaction in High Explosives
8. Tabulated Compaction
9. Tabulated
10. Propellant-Deflagration

The forms of the first five equations of state are given in the KOVEC user's manual [Woodruff 1973] as well as below.

20.1 Equation of State Form 1: Linear Polynomial

This polynomial equation of state, linear in the internal energy per initial volume, E, is given by

$$p = C_0 + C_1\mu + C_2\mu^2 + C_3\mu^3 + \left(C_4 + C_5\mu + C_6\mu^2\right)E \qquad (20.1.1)$$

Here, C_0, C_1, C_2, C_3, C_4, C_5, and C_6 are user defined constants and

$$\mu = \frac{1}{V} - 1. \qquad (20.1.2)$$

where V is the relative volume. In expanded elements, the coefficients of μ^2 are set to zero, i.e.,

$$C_2 = C_6 = 0$$

The linear polynomial equation of state may be used to model gas with the gamma law equation of state. This may be achieved by setting:

$$C_0 = C_1 = C_2 = C_3 = C_6 = 0$$

and

$$C_4 = C_5 = \gamma - 1$$

where γ is the ratio of specific heats. The pressure is then given by:

$$p = (\gamma - 1)\frac{\rho}{\rho_0}E$$

Note that the units of E are the units of pressure.

20.2 Equation of State Form 2: JWL High Explosive

The JWL equation of state defines pressure as a function of relative volume, V, and internal energy per initial volume, E, as

$$p = A\left(1-\frac{\omega}{R_1 V}\right)e^{-R_1 V} + B\left(1-\frac{\omega}{R_2 V}\right)e^{-R_2 V} + \frac{\omega E}{V} \qquad (20.2.1)$$

where $\omega, A, B, R_1,$ and R_2 are user defined input parameters. The JWL equation of state is used for determining the pressure of the detonation products of high explosives in applications involving metal accelerations. Input parameters for this equation are given by Dobratz [1981] for a variety of high explosive materials.

This equation of state is used with the explosive burn (material model 8) material model which determines the lighting time for the explosive element.

20.3 Equation of State Form 3: Sack "Tuesday" High Explosives

Pressure of detonation products is given in terms of the relative volume, V, and internal energy per initial volume, E, as [Woodruff 1973]:

$$p = \frac{A_3}{V^{A_1}}e^{-A_2 V}\left(1-\frac{B_1}{V}\right)+\frac{B_2}{V}E \qquad (20.3.1)$$

where $A_1, A_2, A_3, B_1,$ and B_2 are user-defined input parameters.

This equation of state is used with the explosive burn (material model 8) material model which determines the lighting time for the explosive element.

20.4 Equation of State Form 4: Gruneisen

The Gruneisen equation of state with cubic shock velocity-particle velocity defines pressure for compressed material as

$$p = \frac{\rho_0 C^2 \mu\left[1+\left(1-\frac{\gamma_0}{2}\right)\mu-\frac{a}{2}\mu^2\right]}{\left[1-(S_1-1)\mu-s_2\frac{\mu^2}{\mu+1}-s_3\frac{\mu^3}{(\mu+1)^2}\right]}+(\gamma_0+a\mu)E \qquad (20.4.1)$$

where E is the internal energy per initial volume, C is the intercept of the u_s-u_p curve, $S_1, S_2,$ and S_3 are the coefficients of the slope of the u_s-u_p curve, γ_0 is the Gruneisen gamma, and a is the first order volume correction to γ_0. Constants, $C, S_1, S_2, S_3, \gamma_0,$ and a, are user defined input parameters. The compression is defined in terms of the relative volume, V, as:

$$\mu = \frac{1}{V} - 1.$$

For expanded materials as the pressure is defined by:

$$p = \rho_0 C^2 \mu + (\gamma_0 + \alpha\mu) E \tag{20.4.2}$$

20.5 Equation of State Form 5: Ratio of Polynomials

The ratio of polynomials equation of state defines the pressure as

$$p = \frac{F_1 + F_2 E + F_3 E^2 + F_4 E^3}{F_5 + F_6 E + F_7 E^2} (1 + \alpha\mu) \tag{20.5.1}$$

where

$$F_i = \sum_{j=0}^{n} A_{ij} m^j . \quad n = 4 \; if \; i < 3 \qquad n = 3 \; if \; i \geq 3 \tag{20.5.2}$$

$$\mu = \frac{\rho}{\rho_0 - 1}$$

In expanded zoned F_1 is replaced by $F_1' = F_1 + \beta\mu^2$ Constants A_{ij}, α, and β are user input.

20.6 Equation of State Form 6: Linear With Energy Deposition

This polynomial equation of state, linear in the internal energy per initial volume, E, is given by

$$p = C_0 + C_1\mu + C_2\mu^2 + C_3\mu^3 + (C_4 + C_5\mu + C_6\mu^2) E \tag{20.6.1}$$

Here, C_0, C_1, C_2, C_3, C_4, C_5, and C_6 are user defined constants and

$$\mu = \frac{1}{V} - 1. \tag{20.6.2}$$

where V is the relative volume. In expanded elements, we set the coefficients of μ^2 to zero, i.e.,

$$C_2 = C_6 = 0$$

Internal energy, E, is increased according to an energy deposition rate versus time curve whose ID is defined in the input.

20.7 Equation of State Form 7: Ignition and Growth Model

A JWL equation of state defines the pressure in the unreacted high explosive as

$$P_e = A_e\left(1-\frac{\omega_e}{R1_e V_e}\right)e^{-R1_e V_e} + B_e\left(1-\frac{\omega_e}{R2_e V_e}\right)e^{-R2_e V_e} + \frac{\omega_e E}{V_e} \qquad (20.7.1)$$

where V_e is the relative volume, E_e is the internal energy, and the constants A_e, B_e, ω_e, $R1_e$, and $R2_e$ are input constants. Similarly, the pressure in the reaction products is defined by another JWL form

$$P_p = A_p\left(1-\frac{\omega_p}{R1_p V_p}\right)e^{-R1_p V_p} + B_p\left(1-\frac{\omega_e}{R2_p V_p}\right)e^{-R2_p V_p} + \frac{\omega_p E}{V_p} \qquad (20.7.2)$$

The mixture of unreacted explosive and reaction products is defined by the fraction reacted F ($F = 0$ implies no reaction, $F = 1$ implies complete conversion from explosive to products). The pressures and temperature are assumed to be in equilibrium, and the relative volumes are assumed to be additive:

$$V = (1-F)V_e + FV_p \qquad (20.7.3)$$

The rate of reaction is defined as

$$\frac{\partial F}{\partial t} = I\,(FCRIT-F)^y\left(V_e^{-1}-1\right)^3\left[1+G\left(V_e^{-1}-1\right)\right] + H\,(1-F)^y F^x P^z\left(V_p^{-1}-1\right)^m \qquad (20.7.4)$$

where $I, G, H, x, y, z,$ and m (generally $m = 0$) are input constants.

The JWL equations of state and the reaction rates have been fitted to one- and two-dimensional shock initiation and detonation data for four explosives: PBX-9404, RX-03-BB, PETN, and cast TNT. The details of calculational method are described by Cochran and Chan [1979]. The detailed one-dimensional calculations and parameters for the four explosives are given by Lee and Tarver [1980]. Two-dimensional calculations with this model for PBX 9404 and LX-17 are discussed by Tarver and Hallquist [1981].

20.8 Equation of State Form 8: Tabulated Compaction

Pressure is positive in compression, and volumetric strain ε_V is positive in tension. The tabulated compaction model is linear in internal energy per unit volume. Pressure is defined by

$$p = C(\varepsilon_V) + \gamma T(\varepsilon_V)E \qquad (20.8.1)$$

during loading (compression). Unloading occurs at a slope corresponding to the bulk modulus at the peak (most compressive) volumetric strain, as shown in Figure 20.1. Reloading follows the unloading path to the point where unloading began, and then continues on the loading path described by Equation (20.8.1).

20.9 Equation of State Form 9: Tabulated

The tabulated equation of state model is linear in internal energy. Pressure is defined by

$$p = C(\varepsilon_V) + \gamma T(\varepsilon_V) E \qquad (20.9.1)$$

The volumetric strain ε_V is given by the natural algorithm of the relative volume. Up to 10 points and as few as 2 may be used when defining the tabulated functions. The pressure is extrapolated if necessary. Loading and unloading are along the same curve unlike equation of state form 8.

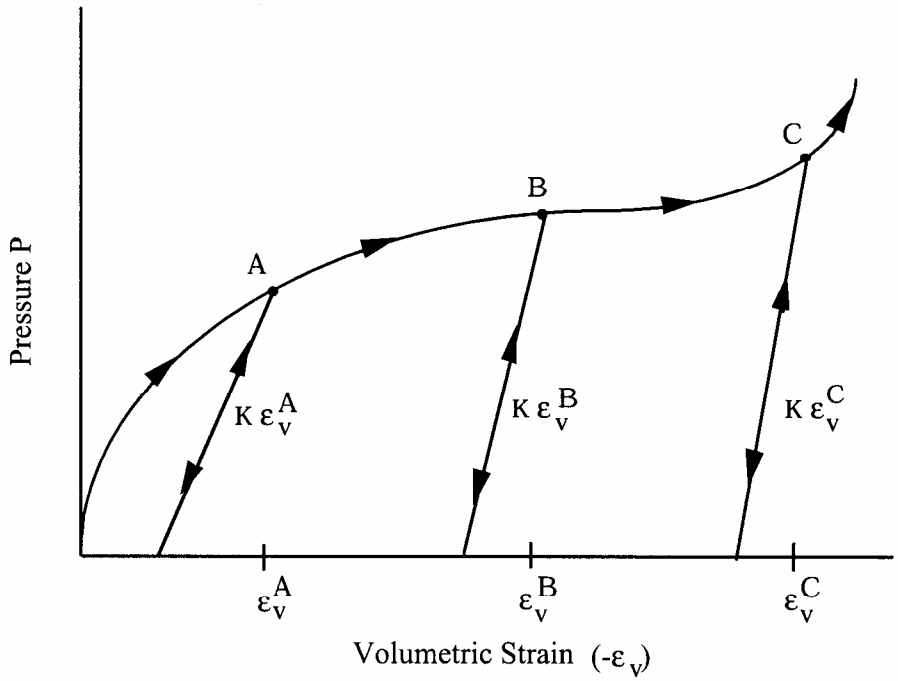


Figure 20.1. Pressure versus volumetric strain curve for equation of state form 8 with compaction. In the compacted states, the bulk unloading modulus depends on the peak volumetric strain.

20.10 Equation of State Form 10: Propellant-Deflagration

A deflagration (burn rate) reactive flow model requires an unreacted solid equation-of-state, a reaction product equation-of-state, a reaction rate law and a mixture rule for the two (or more) species. The mixture rule for the standard ignition and growth model [Lee and Tarver 1980] assumes that both pressures and temperatures are completely equilibrated as the reaction proceeds. However, the mixture rule can be modified to allow no thermal conduction or partial heating of the solid by the reaction product gases. For this relatively slow process of airbag propellant burn, the thermal and pressure equilibrium assumptions are valid. The equations-of-state currently used in the burn model are the JWL, Gruneisen, the van der Waals co-volume, and

the perfect gas law, but other equations-of-state can be easily implemented. In this propellant burn, the gaseous nitrogen produced by the burning sodium azide obeys the perfect gas law as it fills the airbag but may have to be modeled as a van der Waal's gas at the high pressures and temperatures produced in the propellant chamber. The chemical reaction rate law is pressure, particle geometry and surface area dependant, as are most high-pressure burn processes. When the temperature profile of the reacting system is well known, temperature dependent Arrhenius chemical kinetics can be used.

Since the airbag propellant composition and performance data are company private information, it is very difficult to obtain the required information for burn rate modeling. However, Imperial Chemical Industries (ICI) Corporation supplied pressure exponent, particle geometry, packing density, heat of reaction, and atmospheric pressure burn rate data which allowed us to develop the numerical model presented here for their $NaN_3 + Fe_2O_3$ driver airbag propellant. The deflagration model, its implementation, and the results for the ICI propellant are presented in the are described by [Hallquist, et. al., 1990].

The unreacted propellant and the reaction product equations-of-state are both of the form:

$$p = Ae^{-R_1V} + Be^{-R_2V} + \frac{\omega C_v T}{V - d} \qquad (20.10.1)$$

where p is pressure (in Mbars), V is the relative specific volume (inverse of relative density), ω is the Gruneisen coefficient, C_V is heat capacity (in Mbars -cc/cc°K), T is temperature in $°K$, d is the co-volume, and A, B, R_1 and R_2 are constants. Setting $A=B=0$ yields the van der Waal's co-volume equation-of-state. The JWL equation-of-state is generally useful at pressures above several kilobars, while the van der Waal's is useful at pressures below that range and above the range for which the perfect gas law holds. Of course, setting $A=B=d=0$ yields the perfect gas law. If accurate values of ω and C_v plus the correct distribution between "cold" compression and internal energies are used, the calculated temperatures are very reasonable and thus can be used to check propellant performance.

The reaction rate used for the propellant deflagration process is of the form:

$$\frac{\partial F}{\partial t} = \underset{\text{for } 0<F<F_{limit1}}{Z(1-F)^y F^x p^w} + \underset{\text{for } F_{limit2}<F<1}{V(1-F)^u Frp^s} \qquad (20.10.2)$$

where F is the fraction reacted ($F = 0$ implies no reaction, $F = 1$ is complete reaction), t is time, and p is pressure (in Mbars), r,s,u,w,x,y,F_{limit1} and F_{limit2} are constants used to describe the pressure dependence and surface area dependence of the reaction rates. Two (or more) pressure dependent reaction rates are included in case the propellant is a mixture or exhibited a sharp change in reaction rate at some pressure or temperature. Burning surface area dependences can be approximated using the $(1-F)^y F^x$ terms. Other forms of the reaction rate law, such as Arrhenius temperature dependent $e^{-E/RT}$ e-E/RT type rates, can be used, but these require very accurate temperatures calculations. Although the theoretical justification of pressure dependent burn rates at kilobar type pressures is not complete, a vast amount of experimental burn rate

versus pressure data does demonstrate this effect and hydrodynamic calculations using pressure dependent burn accurately simulate such experiments.

The deflagration reactive flow model is activated by any pressure or particle velocity increase on one or more zone boundaries in the reactive material. Such an increase creates pressure in those zones and the decomposition begins. If the pressure is relieved, the reaction rate decreases and can go to zero. This feature is important for short duration, partial decomposition reactions. If the pressure is maintained, the fraction reacted eventually reaches one and the material is completely converted to product molecules. The deflagration front rates of advance through the propellant calculated by this model for several propellants are quite close to the experimentally observed burn rate versus pressure curves.

To obtain good agreement with experimental deflagration data, the model requires an accurate description of the unreacted propellant equation-of-state, either an analytical fit to experimental compression data or an estimated fit based on previous experience with similar materials. This is also true for the reaction products equation-of-state. The more experimental burn rate, pressure production and energy delivery data available, the better the form and constants in the reaction rate equation can be determined.

Therefore the equations used in the burn subroutine for the pressure in the unreacted propellant

$$P_u = R1 \cdot e^{-R5 \cdot V_u} + R2 \cdot e^{-R6 \cdot V_u} + \frac{R3 \cdot T_u}{V_u - FRER} \tag{20.10.3}$$

where V_u and T_u are the relative volume and temperature respectively of the unreacted propellant. The relative density is obviously the inverse of the relative volume. The pressure P_p in the reaction products is given by:

$$P_p = A \cdot e^{-XP1 \cdot V_p} + B \cdot e^{-XP2 \cdot V_p} + \frac{G \cdot Tp}{V_p - CCRIT} \tag{20.10.4}$$

As the reaction proceeds, the unreacted and product pressures and temperatures are assumed to be equilibrated *(Tu = Tp = T, p = Pu = Pp)* and the relative volumes are additive:

$$V = (1 - F) \cdot V_u + F \cdot V_p \tag{20.10.5}$$

where V is the total relative volume. Other mixture assumptions can and have been used in different versions of DYNA2D/3D. The reaction rate law has the form:

$$\begin{aligned} \frac{\partial F}{\partial t} &= GROW1(p + freq)^{em}(F + fmxig)^{ar1}(1 - F + fmxig)^{es1} \\ &+ GROW2(p + freq)^{en}(F + fmxig \end{aligned} \tag{20.10.6}$$

If F exceeds $FMXGR$, the *GROW1* term is set equal to zero, and, if F is less than $FMNGR$, the *GROW2* term is zero. Thus, two separate (or overlapping) burn rates can be used to describe the rate at which the propellant decomposes.

This equation-of-state subroutine is used together with a material model to describe the propellant. In the airbag propellant case, a null material model (type #10) can be used. Material type #10 is usually used for a solid propellant or explosive when the shear modulus and yield strength are defined. The propellant material is defined by the material model and the unreacted equation-of-state until the reaction begins. The calculated mixture states are used until the reaction is complete and then the reaction product equation-of-state is used. The heat of reaction, ENQ, is assumed to be a constant and the same at all values of F but more complex energy release laws could be implemented.

21. ARTIFICIAL BULK VISCOSITY

Bulk viscosity is used to treat shock waves. Proposed in one spatial dimension by von Neumann and Richtmyer [1950], the bulk viscosity method is now used in nearly all wave propagation codes. A viscous term q is added to the pressure to smear the shock discontinuities into rapidly varying but continuous transition regions. With this method the solution is unperturbed away from a shock, the Hugoniot jump conditions remain valid across the shock transition, and shocks are treated automatically. In our discussion of bulk viscosity we draw heavily on works by Richtmyer and Morton [1967], Noh [1976], and Wilkins [1980]. The following discussion of the bulk viscosity applies to solid elements since strong shocks are not normally encountered in structures modeled with shell and beam elements.

21.1 Shock Waves

Shock waves result from the property that sound speed increases with increasing pressure. A smooth pressure wave can gradually steepen until it propagates as a discontinuous disturbance called a shock. See Figure 21.1. Shocks lead to jumps in pressure, density, particle velocity, and energy.

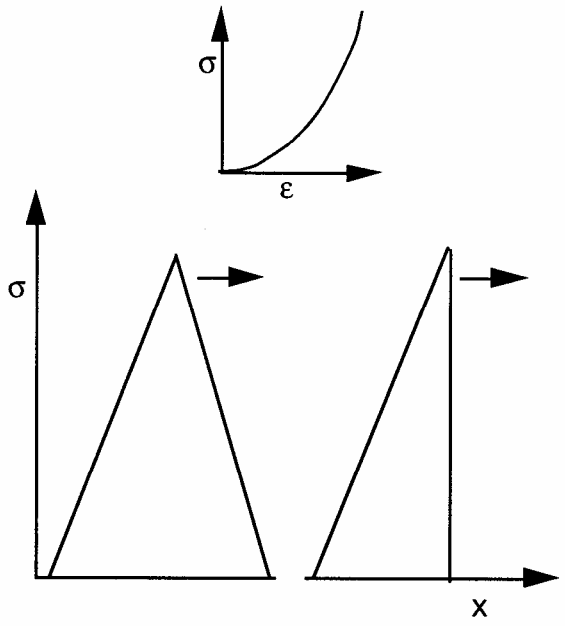


Figure 21.1. If the sound speed increases as the stress increases the traveling wave above will gradually steepen as it moves along the x-coordinate to form a shock wave.

Consider a planar shock front moving through a material. Mass, momentum, and energy are conserved across the front. Application of these conservation laws leads to the well-known Rankine-Hugoniot jump conditions

$$u_s = \frac{\rho(u - u_0)}{\rho - \rho_0} \tag{21.1a}$$

$$\rho - \rho_0 = \rho_0 u_s \left(u - u_0\right) \tag{21.1b}$$

$$e - e_0 = \frac{p - p_0}{2} \frac{\rho - \rho_0}{\rho_0 \rho} \tag{21.1c}$$

where Equation (21.1c) is an expression of the energy jump condition using the results of mass conservation, Equation (21.1a), and momentum conservation, Equation (21.1b). Here, u_s is the shock velocity, u is the particle velocity, ρ is the density, e is the specific internal energy, p is the pressure, and the subscript, $_0$, indicates the state ahead of the shock.

The energy equation relating the thermodynamic quantities density, pressure, and energy must be satisfied for all shocks. The equation of state

$$p = p\left(\rho, e\right) \tag{21.2}$$

which defines all equilibrium states that can exist in a material and relating the same quantities as the energy equation, must also be satisfied. We may use this equation to eliminate energy from

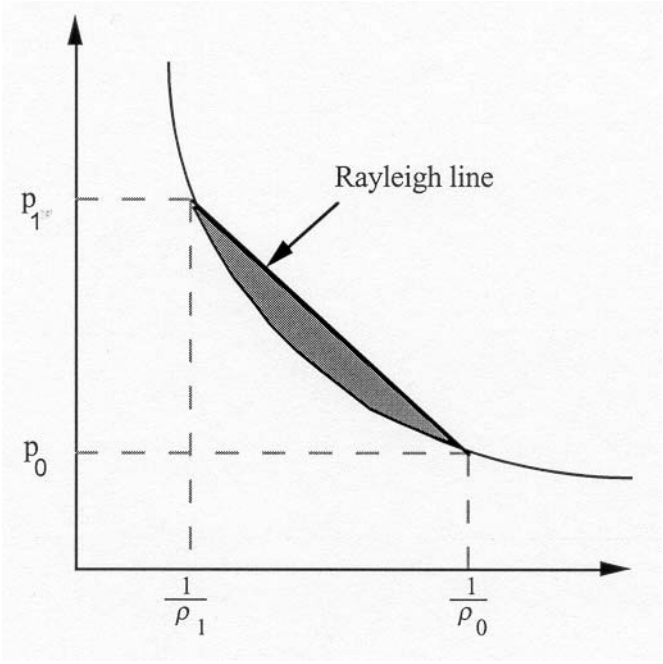


Figure 21.2. Shocking takes place along the Rayleigh line, and release closely follows the Hugoniot. The cross-hatched area is the difference between the internal energy behind the shock and the internal energy lost on release.

Equation (21.1c) and obtain a unique relationship between pressure and compression. This relation, called the Hugoniot, determines all pressure-compression states achievable behind the shock. Shocking takes place along the Rayleigh line and not the Hugoniot (Figure 21.1) and

because the Hugoniot curve closely approximates an isentrope, we may often assume the unloading follows the Hugoniot. Combining Equations (21.1a) and (21.1b), we see that the slope of the Rayleigh line is related to the shock speed:

$$u_s = \frac{1}{\rho_0}\left[\frac{p_1 - p_0}{\frac{1}{\rho_0} - \frac{1}{\rho}}\right]^{1/2} \tag{21.3}$$

For the material of Figure 21.2, increasing pressure increases shock speed.

Consider a γ-law gas with the equation of state

$$p = (\gamma - 1)\rho e \tag{21.4}$$

where γ is the ratio of specific heats. Using the energy jump condition, we can eliminate e and obtain the Hugoniot

$$\frac{p}{p_0} = p^* = \frac{2V_0 + (\gamma - 1)(V_0 - V)}{2V - (\gamma - 1)(V_0 - V)} \tag{21.5}$$

where V is the relative volume. Figure 21.3 shows a plot of the Hugoniot and adiabat where it is noted that for $p^*=1$, the slopes are equal. Thus for weak shocks, Hugoniot and adiabat agree to the first order and can be ignored in numerical calculations. However, special treatment is

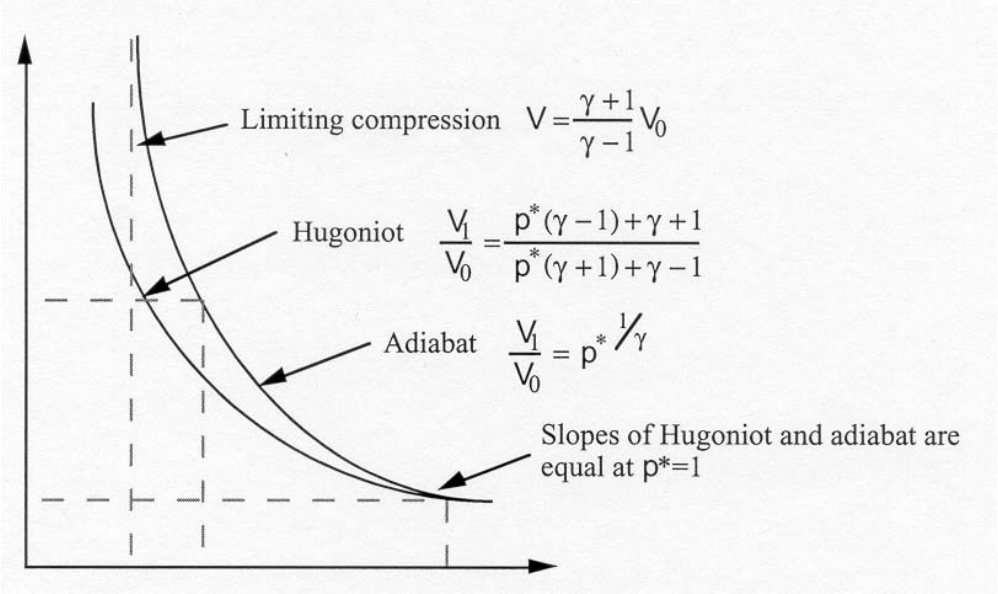


Figure 21.3. Hugoniot curve and adiabat for a g-law gas (from [Noh 1976]).

required for strong shocks, and in numerical calculations this special treatment takes the form of bulk viscosity.

21.2 Bulk Viscosity

In the presence of shocks, the governing partial differential equations can give multiple weak solutions. In their discussion of the Rankine-Hugoniot jump conditions, Richtmyer and Morton [1967] report that the unmodified finite difference (element) equations often will not produce even approximately correct answers. One possible solution is to employ shock fitting techniques and treat the shocks as interior boundary conditions. This technique has been used in one spatial dimension but is too complex to extend to multi-dimensional problems of arbitrary geometry. For these reasons the pseudo-viscosity method was a major breakthrough for computational mechanics.

The viscosity proposed by von Neumann and Richtmyer [1950] in one spatial dimension has the form

$$q = C_0 \rho (\Delta x)^2 \left(\frac{\partial \dot{x}}{\partial x} \right)^2 \qquad if \ \frac{\partial \dot{x}}{\partial x} < 0 \tag{21.6}$$

$$q = 0 \qquad if \ \frac{\partial \dot{x}}{\partial x} \geq 0$$

where C_0 is a dimensionless constant and q is added to the pressure in both the momentum and energy equations. When q is used, they proved the following for steady state shocks: the hydrodynamic equations possess solutions without discontinuities; the shock thickness is independent of shock strength and of the same order as the Δx used in the calculations; the q term is insignificant outside the shock layer; and the jump conditions are satisfied. According to Noh, it is generally believed that these properties:

> "hold for all shocks, and this has been borne out over the years by countless numerical experiments in which excellent agreement has been obtained either with exact solutions or with hydrodynamical experiments."

In 1955, Landshoff [1955] suggested the addition of a linear term to the q of von Neumann and Richtmyer leading to a q of the form

$$q = C_0 \rho (\Delta x)^2 \left(\frac{\partial \dot{x}}{\partial x} \right)^2 \qquad if \ \frac{\partial \dot{x}}{\partial x} < 0 \tag{21.7}$$

$$q = 0 \qquad if \ \frac{\partial \dot{x}}{\partial x} \geq 0$$

where C_1 is a dimensionless constant and a is the local sound speed. The linear term rapidly damps numerical oscillations behind the shock front (Figure 21.3). A similar form was proposed independently by Noh about the same time.

In an interesting aside, Wilkins [1980] discusses work by Kuropatenko who, given an equation of state, derived a q by solving the jump conditions for pressure in terms of a change in the particle velocity, Δu . For an equation of state of the form

$$p = K\left(-\frac{\rho}{\rho_0}-1\right), \tag{21.8}$$

pressure across the shock front is given by [Wilkins 1980]

$$p = p_0 + \rho_0 \frac{(\Delta u)^2}{2} + \rho_0 |\Delta u| \left[\frac{(\Delta u)^2}{2} + a^2\right]^{1/2} \tag{21.9}$$

where a is a sound speed

$$a = \left(\frac{K}{\rho_0}\right)^{1/2} \tag{21.10}$$

For a strong shock, $\Delta u^2 >> a^2$, we obtain the quadratic form

$$q = \rho_0 \Delta u^2 \tag{21.11}$$

and for a weak shock, $\Delta u^2 << a^2$, the linear form

$$q = \rho_0 a \Delta u \tag{21.12}$$

Thus linear and quadratic forms for q can be naturally derived. According to Wilkins, the particular expressions for q obtained by Kuropatenko offer no particular advantage over the expressions currently used in most computer programs.

In extending the one-dimensional viscosity formulations to multi-dimensions, most code developers have simply replaced the divergence of the velocity with, $\dot{\varepsilon}_{kk}$, the trace of the strain rate tensor, and the characteristic length with the square root of the area A, in two dimensions and the cubic root of the volume v in three dimensions. These changes also give the default viscosities in the LS-DYNA codes:

$$q = \rho l \left(C_0 l \, \dot{\varepsilon}_{kk}^2 - C_l a \dot{\varepsilon}_{kk}\right) \quad if \ \dot{\varepsilon}_{kk} < 0 \tag{21.13}$$

$$q = 0 \qquad if \ \dot{\varepsilon}_{kk} \geq 0$$

where C_0 and C_1 are dimensionless constants which default to 1.5 and .06, respectively

where $1=\sqrt{A}$ in 2D, and $\sqrt[3]{v}$ in 3D, a is the local sound speed, C_0 defaults to 1.5 and C_1 defaults to .06.

In converging two- and three-dimensional geometries, the strain rate $\dot{\varepsilon}_{kk}$ is negative and the q term in Equation (21.13) is nonzero, even though no shocks may be generated. This results in nonphysical q heating. When the aspect ratios of the elements are poor (far from unity), the use of a characteristic length based on $\sqrt{A}$ or $\sqrt[3]{v}$ can also result in nonphysical q heating and even occasional numerical instabilities. Wilkins uses a bulk viscosity that is based in part on earlier work by Richards [1965] that extends the von Neumann and Richtmyer formulations in a way that avoids these problems. This latter q may be added in the future if the need arises.

Wilkins' q is defined as:

$$\begin{aligned} q &= C_0\ \rho\ l^2\left(\frac{ds}{dt}\right)^2 - C_l\rho\ l\ a^*\frac{ds}{dt} \quad if\ \dot{\varepsilon}_{kk} < 0 \\ q &= 0 \qquad\qquad\qquad\qquad\qquad\qquad if\ \frac{ds}{dt} \geq 0 \end{aligned} \tag{21.14}$$

where l and $\frac{ds}{dt}$ are the thickness of the element and the strain rate in the direction of the acceleration, respectively, and a^* is the sound speed defined by $(p/\rho)^{1/2}$ if $p>0$. We use the local sound speed in place of a^* to reduce the noise at the low stress levels that are typical of our applications.

Two disadvantages are associated with Equation (21.14). To compute the length parameter and the strain rate, we need to know the direction of the acceleration through the element. Since the nodal force vector becomes the acceleration vector in the explicit integration scheme, we have to provide extra storage to save the direction. In three dimensions our present storage capacity is marginal at best and sacrificing this storage for storing the direction would make it even more so. Secondly, we need to compute l and $\frac{ds}{dt}$ which results in a noticeable increase in computer cost even in two dimensions. For most problems the additional refinement of Wilkins is not needed. However, users must be aware of the pitfalls of Equation (21.13), i.e., when the element aspect ratios are poor or the deformations are large, an anomalous q may be generated.

22. TIME STEP CONTROL

During the solution we loop through the elements to update the stresses and the right hand side force vector. We also determine a new time step size by taking the minimum value over all elements.

$$\Delta t^{n+1} = a \cdot \min\left\{\Delta t_1, \Delta t_2, \Delta t_3, ..., \Delta t_N\right\} \tag{22.1}$$

where N is the number of elements. For stability reasons the scale factor a is typically set to a value of .90 (default) or some smaller value.

22.1 Time Step Calculations for Solid Elements

A critical time step size, Δt_e, is computed for solid elements from

$$\Delta t_e = \frac{L_e}{\left\{\left[Q + \left(Q^2 + c^2\right)^{1/2}\right]\right\}} \tag{22.2}$$

where Q is a function of the bulk viscosity coefficients C_0 and C_1:

$$Q = \begin{cases} C_1 c + C_0 L_e \left|\dot{\varepsilon}_{kk}\right| & for\ \dot{\varepsilon}_{kk} < 0 \\ 0 & for\ \dot{\varepsilon}_{kk} \geq 0 \end{cases} \tag{22.3}$$

L_e is a characteristic length:

8 node solids: $L_e = \dfrac{\upsilon_e}{A_{e\max}}$

4 node tetrahedras: L_e = minimum altitude

υ_e is the element volume, $A_{e\max}$ is the area of the largest side, and c is the adiabatic sound speed:

$$c = \left[\frac{4G}{3\rho_0} + \frac{\partial p}{\partial \rho}\right)_s\Bigg]^{1/2} \tag{22.4}$$

where ρ is the specific mass density. Noting that:

$$\frac{\partial p}{\partial \rho}\Big)_s = \frac{\partial p}{\partial \rho}\Big)_{\mathrm{E}} + \frac{\partial p}{\partial \mathrm{E}}\Big)_\rho \frac{\partial \mathrm{E}}{\partial \rho}\Big)_s \tag{22.5}$$

and that along an isentrope the incremental energy, E, in the units of pressure is the product of pressure, p, and the incremental relative volume, dV:

$$dE = -pdV \tag{22.6}$$

we obtain

$$c = \left[\frac{4G}{3\rho_0} + \frac{\partial p}{\partial \rho} \right)_{\mathrm{E}} + \frac{pV^2}{\rho_0} \frac{\partial p}{\partial \mathrm{E}} \Big)_{\rho} \Bigg]^{1/2} \tag{22.7}$$

For elastic materials with a constant bulk modulus the sound speed is given by:

$$c = \sqrt{\frac{E(1-\upsilon)}{(1+\upsilon)(1-2\upsilon)\rho}} \tag{22.8}$$

where E is Young's modulus, and υ is Poisson's ratio.

22.2 Time Step Calculations for Beam and Truss Elements

For the Hughes-Liu beam and truss elements, the time step size is give by:

$$\Delta t_e = \frac{L}{c} \tag{22.9}$$

where L is the length of the element and c is the wave speed:

$$c = \sqrt{\frac{E}{\rho}} \tag{22.10}$$

For the Belytschko beam the time step size given by the longitudinal sound speed is used (Equation (22.9)), unless the bending-related time step size given by [Belytschko and Tsay 1982]

$$\Delta t_e = \frac{.5L}{c\sqrt{3I\left[\dfrac{3}{12I + AL^2} + \dfrac{1}{AL^2}\right]}} \tag{22.11}$$

is smaller, where I and A are the maximum value of the moment of inertia and area of the cross section, respectively.

Comparison of critical time steps of the truss versus the elastic solid element shows that it if Poisson's ratio, υ, is nonzero the solid elements give a considerably smaller stable time step size. If we define the ratio, α, as:

$$\alpha = \frac{\Delta t_{continuum}}{\Delta t_{rod}} = \frac{C_{rod}}{C_{continuum}} = \sqrt{\frac{(1+\upsilon)(1-2\upsilon)}{1-\upsilon}}, \tag{22.12}$$

we obtain the results in Table 22.1 where we can see that as υ approaches .5 $\alpha \to 0$.

υ	0	0.2	0.3	0.4	0.45	0.49	0.50
α	1.	0.949	0.862	0.683	0.513	0.242	0.0

Table 22.1. Comparison of critical time step sizes for a truss versus a solid element.

22.3 Time Step Calculations for Shell Elements

For the shell elements, the time step size is given by:

$$\Delta t_e = \frac{L_s}{c} \tag{22.13}$$

where L_s is the characteristic length and c is the sound speed:

$$c = \sqrt{\frac{E}{\rho(1-\nu^2)}} \tag{22.14}$$

Three user options exists for choosing the characteristic length. In the default or first option the characteristic length is given by:

$$L_s = \frac{(1+\beta)A_s}{\max(L_1, L_2, L_3, (1-\beta)L_4)} \tag{22.15}$$

where $\beta = 0$ for quadrilateral and 1 for triangular shell elements, A_s is the area, and $Li, (i= 1. . . .4)$ is the length of the sides defining the shell elements. In the second option a more conservative value of L_s is used:

$$L_s = \frac{(1+\beta)A_s}{\max(D_1, D_2)} \tag{22.16}$$

where D_i $(i=1,2)$ is the length of the diagonals. The third option provides the largest time step size and is frequently used when triangular shell elements have very short altitudes. The bar wave speed, Equation (21.10), is used to compute the time step size and L_s is give by

$$L_s = \max\left[\frac{(1+\beta)A_s}{\max(L_1, L_2, L_3, (1-\beta)L_4)}, \min(L_1, L_2, L_3, L_4 + \beta 10^{20})\right] \quad (22.17)$$

A comparison of critical time steps of truss versus shells is given in Table 22.2 with β defined as:

$$\beta = \frac{\Delta t_{2D-continuum}}{\Delta t_{rod}} = \frac{C_{rod}}{C} = \sqrt{1-\upsilon^2} \quad (22.18)$$

υ	0	0.2	0.3	0.4	0.5
β	1.0	0.98	0.954	0.917	0.866

Table 22.2. Comparison of critical time step sizes for a truss versus a shell element.

22.4 Time Step Calculations for Solid Shell Elements

A critical time step size, Δt_e is computed for solid shell elements from

$$\Delta t_e = \frac{\upsilon_e}{cA_{e_{\max}}} \quad (22.19)$$

where υ_e is the element volume, $A_{e_{\max}}$ is the area of the largest side, and c is the plane stress sound speed given in Equation (22.14).

22.5 Time Step Calculations for Discrete Elements

For spring elements such as that in Figure 22.1 there is no wave propagation speed c to calculate the critical time step size.

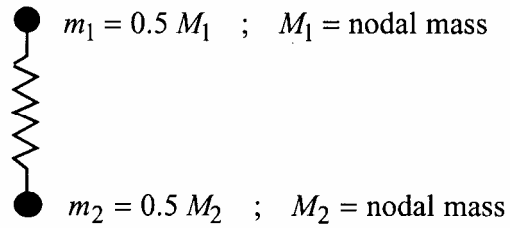


Figure 22.1. Lumped spring mass system.

The eigenvalue problem for the free vibration of spring with nodal masses m_1 and m_2, and stiffness, k, is given by:

$$\begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} - \omega^2 \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{22.20}$$

Since the determinant of the characteristic equation must equal zero, we can solve for the maximum eigenvalue:

$$\det \begin{bmatrix} k - \omega^2 m_1 & -k \\ -k & k - \omega^2 m_2 \end{bmatrix} = 0 \quad \rightarrow \quad \omega_{\max}^2 = \frac{k\left(m_1 + m_2\right)}{m_1 \cdot m_2} \tag{22.21}$$

Recalling the critical time step of a truss element:

$$\left.\begin{aligned} \Delta t &\le \frac{\ell}{c} \\ \omega_{\max} &= \frac{2c}{\ell} \end{aligned}\right\} \quad \Delta t \le \frac{2}{\omega_{\max}} \tag{22.22}$$

and approximating the spring masses by using 1/2 the actual nodal mass, we obtain:

$$\Delta t = 2\sqrt{\frac{m_1 m_2}{m_1 + m_2}\frac{1}{k}} \tag{22.23}$$

Therefore, in terms of the nodal mass we can write the critical time step size as:

$$\Delta t_e = 2\sqrt{\frac{2M_1 M_2}{k\left(M_1 + M_2\right)}} \tag{22.24}$$

The springs used in the contact interface are not checked for stability.

23. BOUNDARY AND LOADING CONDITIONS

23.1 Pressure Boundary Conditions

Consider pressure loadings on boundary ∂b_1 in Equation (2.4). To carry out the surface integration indicated by the integral

$$\int_{\partial b_1} N^t t \, ds \tag{23.1}$$

a Gaussian quadrature rule is used. To locate any point of the surface under consideration, a position vector, r, is defined:

$$r = f_1(\xi,\eta)i_1 + f_2((\xi,\eta)i_2 + f_3(\xi,\eta)i_3 \tag{23.2}$$

where

$$f_i(\xi,\eta) = \sum_{j=1}^{4} \phi_j x_i^j \tag{23.3}$$

and i_1, i_2, i_3 are unit vectors in the x_1, x_2, x_3 directions (see Figure 23.1).

Nodal quantities are interpolated over the four-node linear surface by the functions

$$\phi_i = \frac{1}{4}(1+\xi\xi_i)(1+\eta\eta_i) \tag{23.4}$$

so that the differential surface area ds may be written in terms of the curvilinear coordinates as

$$ds = |J| d\xi d\eta \tag{23.5}$$

where $|\mathrm{J}|$ is the surface Jacobian defined by

$$|\mathrm{J}| = \left| \frac{\partial r}{\partial \xi} \times \frac{\partial r}{\partial \eta} \right| = \left(EG - F^2 \right)^{1/2} \tag{23.6}$$

in which

$$\begin{aligned} E &= \frac{\partial r}{\partial \xi} \cdot \frac{\partial r}{\partial \xi} \\ F &= \frac{\partial r}{\partial \xi} \cdot \frac{\partial r}{\partial \eta} \\ G &= \frac{\partial r}{\partial \eta} \cdot \frac{\partial r}{\partial \eta} \end{aligned} \tag{23.7}$$

A unit normal vector n to the surface segment is given by

$$n = |\mathrm{J}|^{-1}\left(\frac{\partial r}{\partial \xi}\times\frac{\partial r}{\partial \eta}\right) \tag{23.8}$$

and the global components of the traction vector can now be written

$$t_i = n_i \sum_{j=1}^{4} \phi_j p^j \tag{23.9}$$

where p_j is the applied pressure at the jth node.

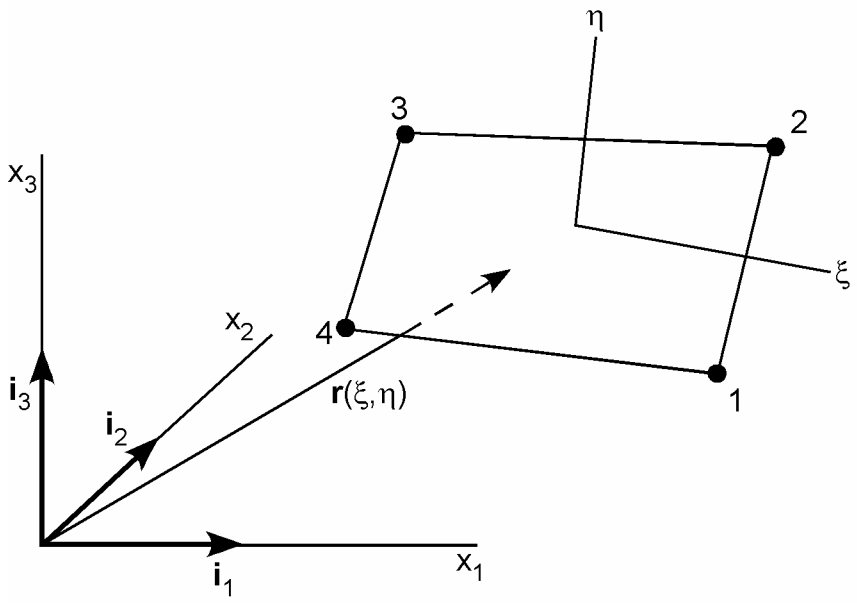


Figure 23.1. Parametric representation of a surface segment.

The surface integral for a segment is evaluated as:

$$\int_{-1}^{1}\int_{-1}^{1} N^t t\,|J|\,d\xi\,d\eta \tag{23.10}$$

One such integral is computed for each surface segment on which a pressure loading acts. Note that the Jacobians cancel when Equations (23.8) and (23.7) are put into Equation (23.10). Equation (23.10) is evaluated with one-point integration analogous to that employed in the volume integrals. The area of an element side is approximated by $4\,|J|$ where $|J| = |J\ \ (0,\ 0)|$.

23.2 Transmitting Boundaries

Transmitting boundaries are available only for problems that require the modeling of semi-infinite or infinite domains with solid elements and therefore are not available for beam or shell elements. Applications of this capability include problems in geomechanics and underwater structures.

The transmitting or silent boundary is defined by providing a complete list of boundary segments. In the approach used, discussed by Cohen and Jennings [1983] who in turn credit the method to Lysmer and Kuhlemeyer [1969], viscous normal shear stresses in Equation (23.11) are applied to the boundary segments:

$$\sigma_{normal} = -\rho c_d V_{normal} \tag{23.11a}$$

$$\sigma_{shear} = -\rho c_s V_{tan} \tag{23.11b}$$

where ρ, c_d, and c_s are the material density, dilatational wave speed, and the shear wave speed of the transmitting media respectively. The magnitude of these stresses is proportional to the particle velocities in the normal, V_{normal}, and tangential, V_{tan}, directions. The material associated with each transmitting segment is identified during initialization so that unique values of the constants ρ, c_d, and c_s can be defined automatically.

23.3 Kinematic Boundary Conditions

In this subsection, the kinematic constraints are briefly reviewed. LS-DYNA tracks reaction forces for each type of kinematic constraint and provides this information as output if requested. For the prescribed boundary conditions, the input energy is integrated and included in the external work.

23.3.1 Displacement Constraints

Translational and rotational boundary constraints are imposed either globally or locally by setting the constrained acceleration components to zero. If nodal single point constraints are employed, the constraints are imposed in a local system. The user defines the local system by specifying a vector u_1 in the direction of the local x-axis x_l, and a local in-plane vector v_l. After normalizing u_1, the local x_l, y_l and z_l axes are given by:

$$x_l = \frac{u_l}{\|u_l\|} \tag{23.12}$$

$$z_l = \frac{x_l \times v_l}{\|x_l \times v_l\|} \tag{23.13}$$

$$y_l = z_l \times x_l \tag{23.14}$$

A transformation matrix q is constructed to transform the acceleration components to the local system:

$$q = \begin{bmatrix} x_l^t \\ y_l^t \\ z_l^t \end{bmatrix} \tag{23.15}$$

and the nodal translational and rotational acceleration vectors a_I and $\dot{\omega}_I$, for node I are transformed to the local system:

$$a_{I_l} = q a_I \tag{23.16a}$$

$$\dot{\omega}_{I_l} = q\dot{\omega}_I \tag{23.16b}$$

and the constrained components are zeroed. The modified vectors are then transformed back to the global system:

$$a_I = q^t a_{I_l} \tag{23.17a}$$

$$\dot{\omega}_I = q^t \dot{\omega}_{I_l} \tag{23.17b}$$

23.3.2 Prescribed Displacements, Velocities, and Accelerations

Prescribed displacements, velocities, and accelerations are treated in a nearly identical way to displacement constraints. After imposing the zero displacement constraints, the prescribed values are imposed as velocities at time, $t^{n+1/2}$. The acceleration versus time curve is integrated or the displacement versus time curve is differentiated to generate the velocity versus time curve. The prescribed nodal components are then set.

23.4 Body Force Loads

Body force loads are used in many applications. For example, in structural analysis the base accelerations can be applied in the simulation of earthquake loadings, the gun firing of projectiles, and gravitational loads. The latter is often used with dynamic relaxation to initialize the internal forces before proceeding with the transient response calculation. In aircraft engine design the body forces are generated by the application of an angular velocity of the spinning structure. The generalized body force loads are available if only part of the structure is subjected to such loadings, e.g., a bird striking a spinning fan blade.

For base accelerations and gravity we can fix the base and apply the loading as part of the body force loads element by element according to Equation (22.18)

$$f_{ebody} = \int_{v_m} \rho \, \mathrm{N}^t \, N a_{base} dv = m_e a_{base} \tag{23.18}$$

where a_{base} is the base acceleration and m_e is the element (lumped) mass matrix.

24. TIME INTEGRATION

24.1 Background

Consider the single degree of freedom damped system in Figure 23.1.

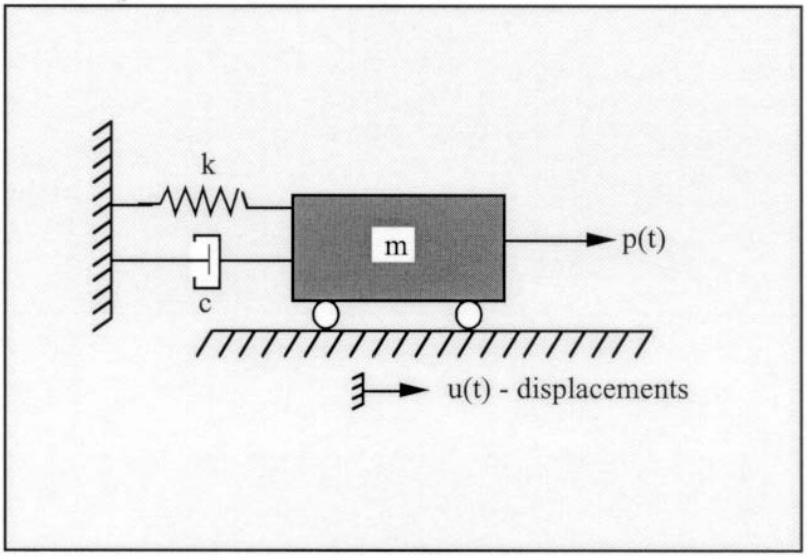


Figure 24.1. Single degree of freedom damped system.

Forces acting on mass m are shown in Figure 24.2.

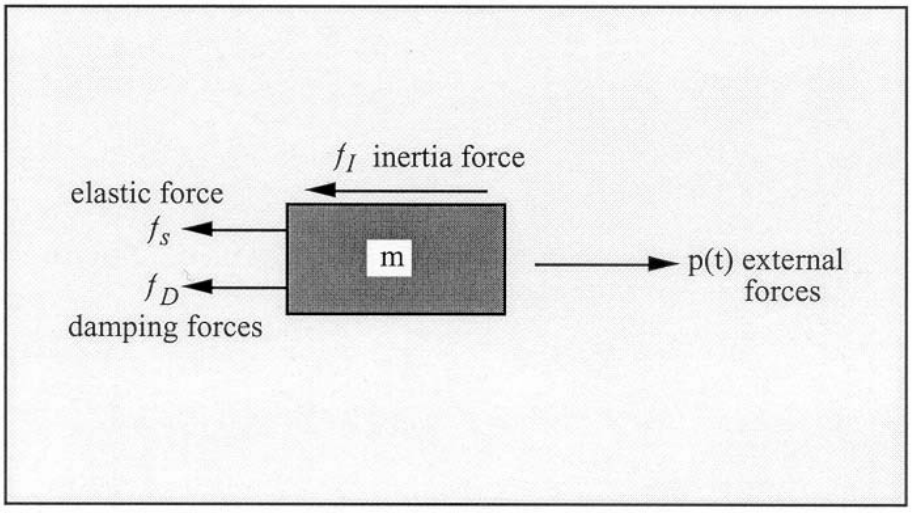


Figure 24.2. Forces acting on mass, m.

The equations of equilibrium are obtained from d'Alembert's principle

$$f_I + f_D + f_{\text{int}} = p(t) \tag{24.1}$$

$$f_I = m\ddot{u} \quad ; \quad \ddot{u} = \frac{d^2u}{dt^2} \quad \textit{acceleration}$$
$$f_D = c\dot{u} \quad ; \quad \dot{u} = \frac{du}{dt} \quad \textit{velocity} \qquad (24.2)$$
$$f_{\text{int}} = k \cdot u \quad ; \quad u \quad \textit{displacement}$$

where c is the damping coefficient, and k is the linear stiffness. For critical damping $c = c_{cr}$. The equations of motion for linear behavior lead to a linear ordinary differential equation, o.d.e.:

$$m\ddot{u} + c\dot{u} + ku = p(t) \qquad (24.3)$$

but for the nonlinear case the internal force varies as a nonlinear function of the displacement, leading to a nonlinear o.d.e.:

$$m\ddot{u} + c\dot{u} + f_{\text{int}}(u) = p(t) \qquad (24.4)$$

Analytical solutions of linear ordinary differential equations are available, so instead we consider the dynamic response of linear system subjected to a harmonic loading. It is convenient to define some commonly used terms:

Harmonic loading: $p(t) = p_0 \sin \varpi t$

Circular frequency: $\omega = \sqrt{\frac{k}{m}}$ for single degree of freedom

Natural frequency: $f = \frac{\omega}{2\pi} = \frac{1}{T} \quad T = \textit{period}$ (24.5)

Damping ratio: $\xi = \frac{c}{c_{cr}} = \frac{c}{2m\omega}$

Damped vibration frequency: $\omega_0 = \omega\sqrt{1-\xi^2}$

Applied load frequency: $\beta = \frac{\bar{\omega}}{\omega}$

The closed form solution is:

$$u(t) = u_0 \cos + \frac{\dot{u}_0}{\omega}\sin\omega t + \frac{p_0}{k}\frac{1}{1-\beta^2}\left(\sin\bar{\omega}t - \beta\sin\omega t\right) \qquad (24.6)$$

homogeneous solution *steady state* *transient*
particular solution

with the initial conditions:

u_0 = initial displacement
$\dot{u}_0$ = initial velocity
$\frac{p_0}{k}$ = static displacement

For nonlinear problems, only numerical solutions are possible. LS-DYNA uses the explicit central difference scheme to integrate the equations of motion.

24.2 The Central Difference Method

The semi-discrete equations of motion at time n are:

$$Ma^n = P^n - F^n + H^n \qquad (24.7)$$

where M is the diagonal mass matrix, P^n accounts for external and body force loads, F^n is the stress divergence vector, and H^n is the hourglass resistance. To advance to time t^{n+1}, we use central difference time integration:

$$a^n = M^{-1}\quad (P^n - F^n + H^n) \qquad (24.8)$$

$$v^{n+\frac{1}{2}} = v^{n-\frac{1}{2}} + a^n\Delta t^n \qquad (24.9)$$

$$u^{n+1} = u^n + v^{n+\frac{1}{2}}\Delta t^{n+\frac{1}{2}} \qquad (24.10)$$

where

$$\Delta t^{n+\frac{1}{2}} = \frac{\left(\Delta t^n + \Delta t^{n+1}\right)}{2} \qquad (24.11)$$

and v and u are the global nodal velocity and displacement vectors, respectively. We update the geometry by adding the displacement increments to the initial geometry:

$$x^{n+1} = x^0 + u^{n+1} \qquad (24.12)$$

We have found that, although more storage is required to store the displacement vector the results are much less sensitive to round-off error.

24.3 Stability of Central Difference Method

The stability of the central difference scheme is determined by looking at the stability of a linear system. The system of linear equations in uncoupled into the modal equations where the modal matrix of eigenvectors, ϕ, are normalized with respect to the mass and linear stiffness matrices K, and M, respectively, such that:

$$\begin{aligned} \phi^T M \phi &= I \\ \phi^T K \phi &= \omega^2 \end{aligned} \tag{24.13}$$

With this normalization, we obtain for viscous proportional damping the decoupling of the damping matrix, C:

$$\phi^T C \phi = 2\xi\omega \tag{24.14}$$

The equations of motion in the modal coordinates x are:

$$\ddot{x} + 2\xi\omega\dot{x} + \omega^2 x = \underbrace{\phi^T p}_{=Y} \tag{24.15}$$

With central differences we obtain for the velocity and acceleration:

$$\dot{x}_n = \frac{x_{n+1} - x_{n-1}}{2\Delta t} \tag{24.16}$$

$$\ddot{x}_n = \frac{x_{n+1} - 2x_n + x_{n-1}}{\Delta t^2} \tag{24.17}$$

Substituting $\dot{x}_n$ and $\ddot{x}_n$ into equation of motion at time t^n leads to:

$$x_{n+1} = \frac{2 - \omega^2 \Delta t^2}{1 + 2\xi\omega\Delta t^2} x_n - \frac{1 - 2\xi\omega\Delta t}{1 + 2\xi\omega\Delta t} x_{n-1} + \frac{\Delta t^2}{1 + 2\xi\omega\Delta t^2} Y_n \tag{24.18}$$

$$x_n = x_n \tag{24.19}$$

which in matrix form leads to

$$\begin{bmatrix} x_{n+1} \\ x_n \end{bmatrix} = \begin{bmatrix} \frac{2 - \omega^2 \Delta t^2}{1 + 2\xi\omega\Delta t} & -\frac{1 - 2\xi\omega\Delta t}{1 + 2\xi\omega\Delta t} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_n \\ x_{n-1} \end{bmatrix} + \begin{bmatrix} \frac{\Delta t^2}{1 + 2\xi\omega\Delta t^2} \\ 0 \end{bmatrix} Y_n \tag{24.20}$$

or

$$\hat{x}_{n+1} = \quad A \quad \hat{x}_n \quad + \quad L \quad Y_n \tag{24.21}$$

where, A is the time integration operator for discrete equations of motion. After m time steps with $L=0$ we obtain:

$$\hat{x}_m = A^m \hat{x}_0 \tag{24.22}$$

As m approaches infinity, A must remain bounded.
A spectral decomposition of A gives:

$$A^m = \left(P^T J P\right)^m = P^T J^m P \tag{24.23}$$

where, P, is the orthonormal matrix containing the eigenvectors of A, and J is the Jordan form with the eigenvalues on the diagonal. The spectral radius, $\rho(A)$, is the largest eigenvalue of $A=\max$ [diag. (J)]. We know that J^m, is bounded if and only if:

$$\left|\rho\left(A\right) \le 1\right| \tag{24.24}$$

Consider the eigenvalues of A for the undamped equation of motion

$$Det\left[\begin{vmatrix} 2-\omega^2\Delta t^2 & -1 \\ 1 & 0 \end{vmatrix} - \lambda \begin{vmatrix} 1 & -1 \\ 1 & 0 \end{vmatrix}\right] = 0 \tag{24.25}$$

$$-\left(2-\omega^2\Delta t^2-\lambda\right)\cdot\lambda+1=0 \tag{24.26}$$

$$\lambda = \frac{2-\omega^2\Delta t^2}{2} \pm \sqrt{\frac{\left(2-\omega^2\Delta t^2\right)^2}{4}-1} \tag{24.27}$$

The requirement that $\left|\lambda\right| \le 1$ leads to:

$$\Delta t \le \frac{2}{\omega_{\max}} \tag{24.28}$$

as the critical time step. For the damped equations of motion we obtain:

$$\Delta t \le \frac{2}{\omega_{\max}}\left(\sqrt{1+\xi^2}-\xi\right). \tag{24.29}$$

Thus, damping reduces the critical time step size. The time step size is bounded by the largest natural frequency of the structure which, in turn, is bounded by the highest frequency of any individual element in the finite element mesh.

24.4 Subcycling (Mixed Time Integration)

The time step size, Δt, is always limited by a single element in the finite element mesh. The idea behind subcycling is to sort elements based on their step size into groups whose step size is some even multiple of the smallest element step size, $2^{(n-1)}\Delta t$, for integer values of n greater than or equal to 1. For example, in Figure 24.1 the mesh on the right because of the thin row of elements is three times more expensive than the mesh on the left

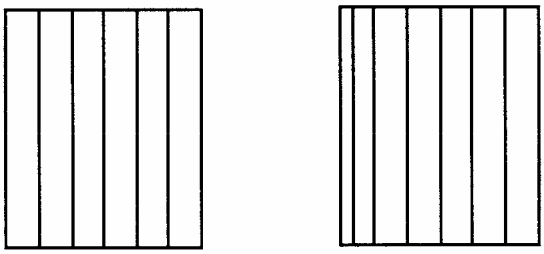


Figure 24.3. The right hand mesh is much more expensive to compute than the left hand due to the presence of the thinner elements.

The subcycling in LS-DYNA is based on the linear nodal interpoation partition subcycling algorithm of Belytschko, Yen, and Mullen [1979], and Belytschko [1980]. In their implementation the steps are:

1. Assign each node, i, a time step size, Δt_j, according to:

   $\Delta t_i = min \left(2/\omega_j \right)$ over all elements j, connected to node i

2. Assign each element, j, a time step size, Δt_j, according to:

   $\Delta t_j = min\,(\Delta t_i)$ over all nodes, i, of element, j

3. Group elements into blocks by time step size for vectorization.

In LS-DYNA we desire to use constant length vectors as much as possible even if it means updating the large elements incrementally with the small time step size. We have found that doing this decreases costs and stability is unaffected.

Hulbert and Hughes [1988] reviewed seven subcycling algorithms in which the partitioning as either node or element based. The major differences within the two subcycling methods lie in how elements along the interface between large and small elements are handled, a subject which is beyond the scope of this theoretical manual. Nevertheless, they concluded that three of the methods including the linear nodal interpolation method chosen for LS-DYNA, provide both stable and accurate solutions for the problems they studied. However, there was some concern about the lack of stability and accuracy proofs for any of these methods when applied to problems in structural mechanics.

The implementation of subcycling currently includes the following element classes and contact options:

- Solid elements
- Beam elements

- Shell elements
- Brick shell elements
- Penalty based contact algorithms.

but intentionally excludes discrete elements since these elements generally contribute insignificantly to the calculational costs. The interface stiffnesses used in the contact algorithms are based on the minimum value of the slave node or master segment stiffness and, consequently, the time step size determination for elements on either side of the interface is assumed to be decoupled; thus, scaling penalty values to larger values when subcycling is active can be a dangerous exercise indeed. Nodes that are included in constraint equations, rigid bodies, or in contact with rigid walls are always assigned the smallest time step sizes.

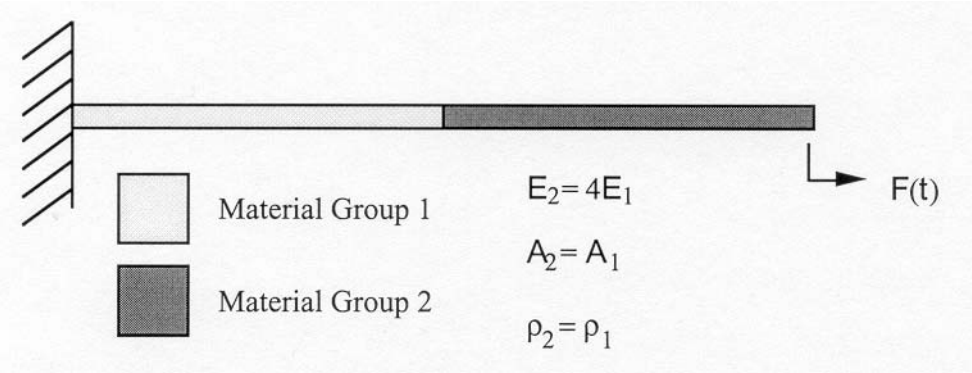


Figure 24.4. Subcycled beam problem from Hulbert and Hughes [1988].

To explain the implementation and the functioning of subcycling, we consider the beam shown in Figure 24.4 where the beam elements on the right (material group 2) have a Courant time step size exactly two times greater than the elements on the left. The nodes attached to material group 2 will be called minor cycle nodes and the rest, major cycle nodes. At time step n=mk all nodal displacements and element stresses are known, where m is the ratio between the largest and smallest time step size, k is the number of major time steps, and n is the number of minor time steps. In Figures 24.5 and 24.6, the update of the state variables to the next major time step k+1 is depicted. The stress state in the element on the material interface in group 1 is updated during the minor cycle as the displacement of the major cycle node shared by this element is assumed to vary linearly during the minor cycle update. This linear variation of the major cycle nodal displacements during the update of the element stresses improves accuracy and stablility.

In the timing study results in Table 24.1, fifty solid elements were included in each group for the beam depicted in Figure 24.4, and the ratio between E2 to E1 was varied from 1 to 128 giving a major time step size greater than 10 times the minor. Note that as the ratio between the major and minor time step sizes approaches infinity the reduction in cost should approach 50 percent for the subcycled case, and we see that this is indeed the case. The effect of subcycling for the more expensive fully integrated elements is greater as may be expected. The overhead of subcycling for the case where E1=E2 is relatively large. This provides some insight into why a decrease in speed is often observed when subcycling is active. For subcycling to have a significant effect, the ratio of the major to minor time step size should be large and the number of elements having the minor step size should be small. In crashworthiness applications the typical

mesh is very well planned and generated to have uniform time step sizes; consequently, subcycling will probably give a net increase in costs.

Case 1 ONE POINT INTEGRATION WITH ELASTIC MATERIAL MODEL	Number of cycles	cpu time(secs)
E2= E1	178 178	4.65 5.36 (+15.%)
E2=4E1	367 367	7.57 7.13 (-6.0%)
E2=16E1	714 715	12.17 10.18 (-20.%)
E2=64E1	1417 1419	23.24 16.39 (-29.%)
E2=128E1	2003 2004	31.89 22.37 (-30.%)

Case 2 EIGHT POINT INTEGRATION WITH ORTHOTROPIC MATERIAL MODEL	Number of cycles	cpu time(secs)
E2= E1	180 180	22.09 22.75 (+3.0%)
E2=4E1	369 369	42.91 34.20 (-20.%)
E2=16E1	718 719	81.49 54.75 (-33.%)
E2=64E1	1424 1424	159.2 97.04 (-39.%)
E2=128E1	2034 2028	226.8 135.5 (-40.%)

Table 24.1. Timing study showing effects of the ratio of the major to minor time step size.

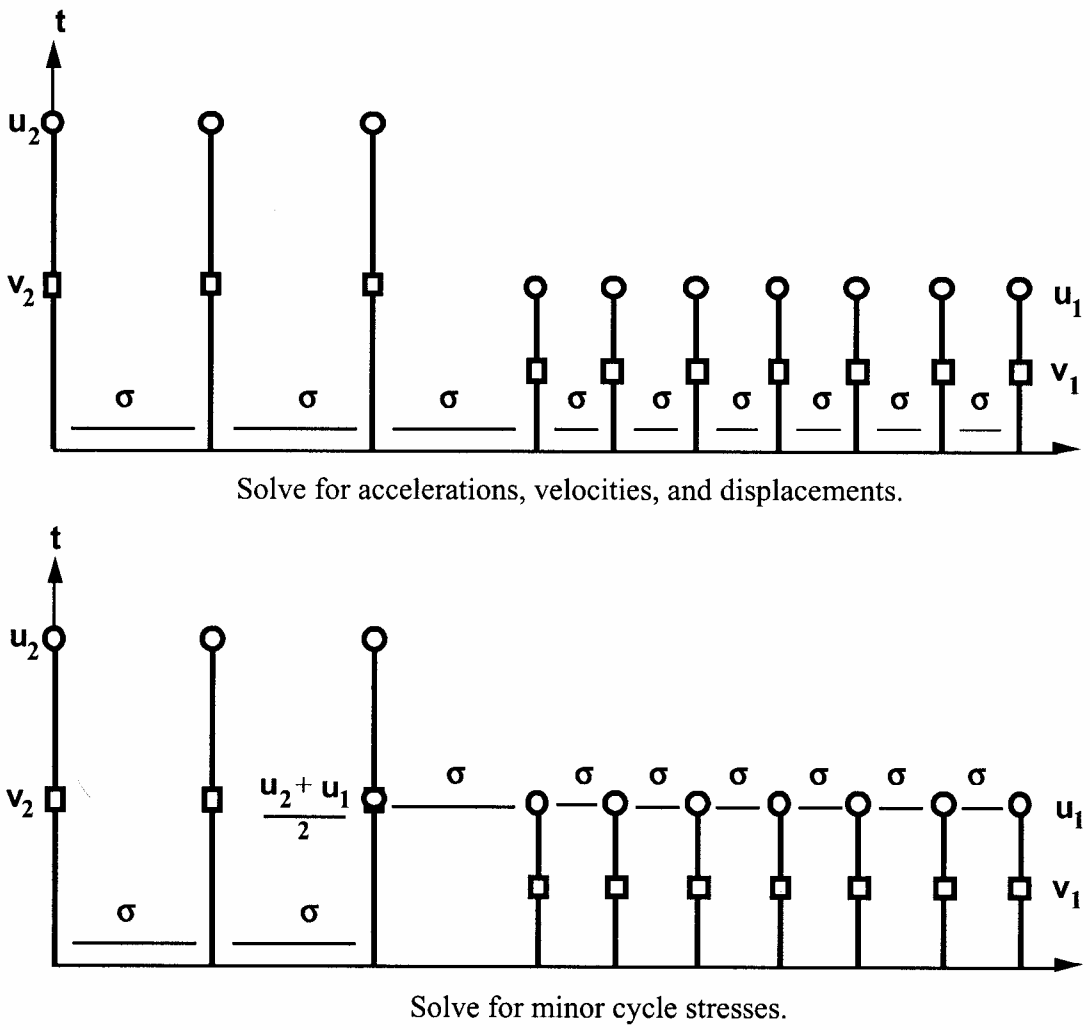


Figure 24.5. Timing diagram for subcycling algorithm based on linear nodal interpolations.

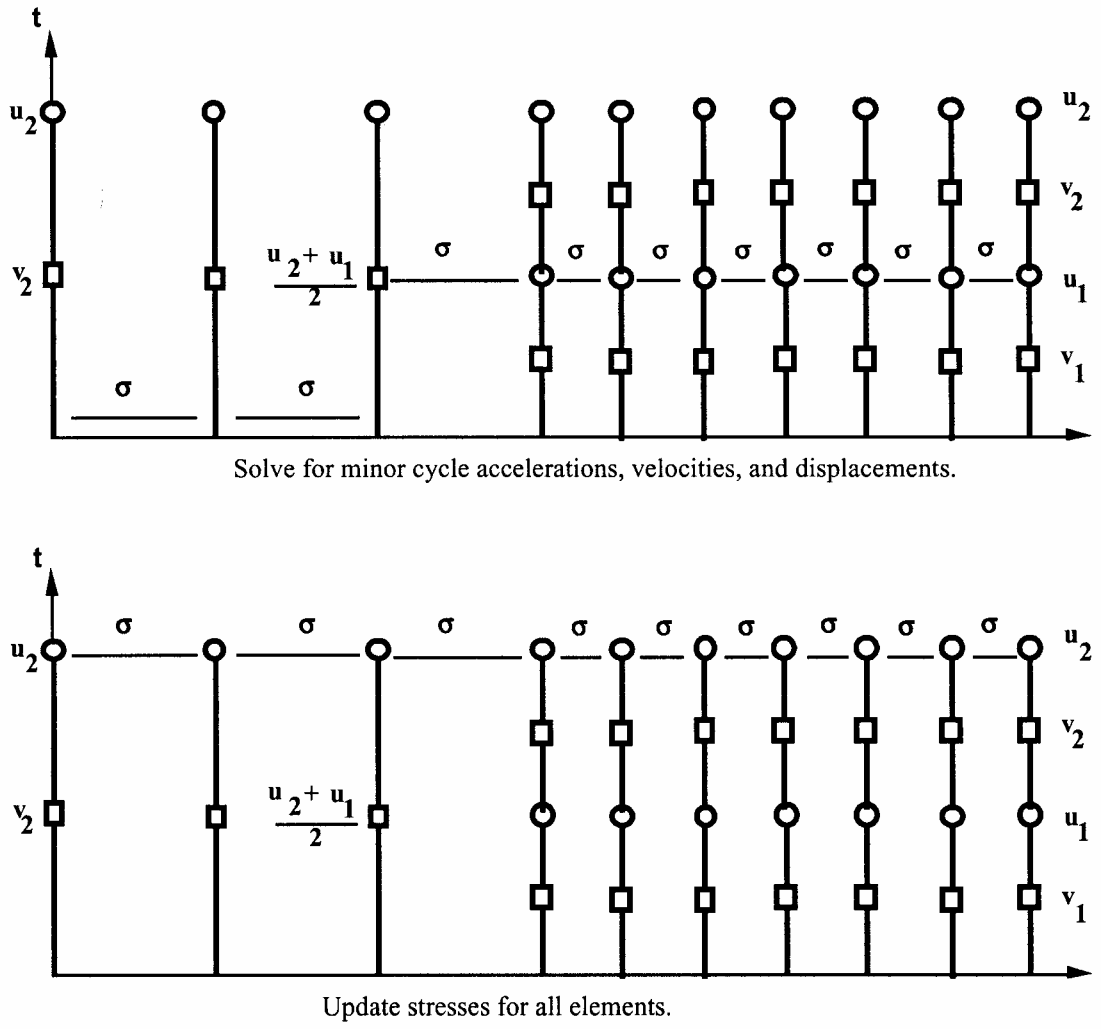


Figure 24.6. Timing diagram for subcycling algorithm based on linear nodal interpolations.

The impact of the subcycling implementation in the software has a very significant effect on the internal structure. The elements in LS-DYNA are now sorted three times

- By element number in ascending order.
- By material number for large vector blocks.
- By connectivity to insure disjointness for right hand side vectorization which is very important for efficiency.

Sorting by Δt, interact with the second and third sorts and can result in the creation of much smaller vector blocks and result in higher cost per element time step. During the simulation elements can continuously change in time step size and resorting may be required to maintain stability; consequently, we must check for this continuously. Sorting cost, though not high when spread over the entire calculation, can become a factor that results in higher overall cost if done

too frequently especially if the factor, m, is relatively small and the ratio of small to large elements is large.

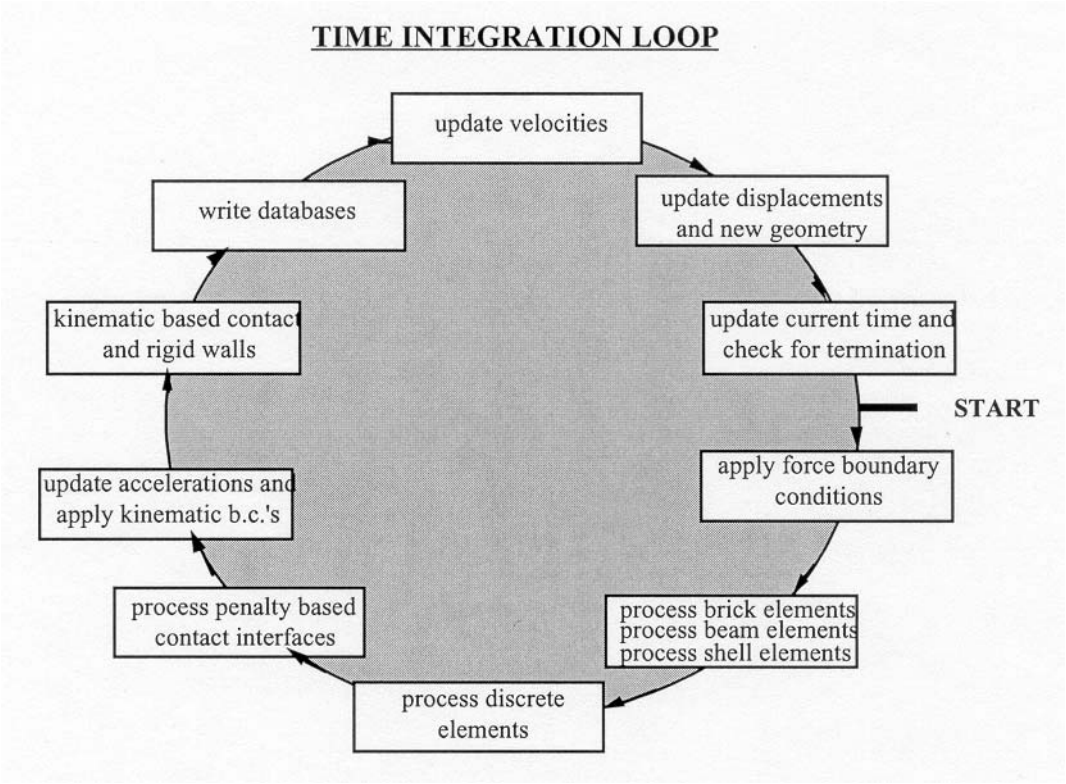


Figure 24.7. The time integration loop in LS-DYNA.

25. RIGID BODY DYNAMICS

A detailed discussion of the rigid body algorithm is presented by Benson and Hallquist [1986] and readers are referred to this publication for more information. The equations of motion for a rigid body are given by:

$$M\ddot{X}_i^{cm} = F_i^x \tag{25.1}$$

$$J_{ij}\dot{\omega}_j + e_{ijk}\omega_j J_{kl}\omega_l = F_i^\omega \tag{25.2}$$

where M is the diagonal mass matrix, J is the inertia tensor, X^{cm} is the location of the center of mass, ω is the angular velocity of the body, and F^x and F^ω are the generalized forces and torques. We can readily solve for the rigid body accelerations:

$$\ddot{X}_i^{cm} = \frac{F_i^x}{M} \tag{25.3}$$

$$\dot{\omega}_j = J_{ij}^{-1}\left[F_i^\omega - e_{ijk}\omega_j J_{kl}\omega_l\right] \tag{25.4}$$

There are three central issues associated with implementing Equations (25.1) and (25.2) in a structural dynamics program:

- calculating M and J from the mesh defining the body;
- calculating F^x and F^ω. This issue is especially critical when structures include rigid bodies contiguous with finite element.
- updating the displacements, velocities and inertia tensor in a manner that does not deform the rigid body.

A rigid body is defined using a finite element mesh by specifying that all of the elements in a region are rigid. In many of the equations that follow, summations are performed over all of the nodes associated with all of the elements in a rigid body, and these special summations are denoted $\sum_\alpha^{RB}$.

The rigid body mass is readily calculated from Equation (25.5):

$$M = \sum_{\alpha,\beta}^{RB} M_{\alpha i \beta i} \quad \text{(no sum on } i\text{)} \tag{25.5}$$

where in explicit analyses the mass matrix is diagonal. The most popular mass lumping procedures scale the rotation masses to increase the allowable integration step size under the central difference stability criterion. Scaling should not be performed on the mass matrix entries associated with the rigid body nodes because the rigid body elements do not affect the step size and scaling reduces the accuracy of the calculated inertia tensor. For thin shells, the

contributions to the inertia tensor from the rotational degrees of freedom are usually small and can be neglected. Thus, the inertia tensor is found by a nodal summation of the product of the point masses with their moment arms.

$$J_{ij} = e_{irs} e_{jnm} M_{\alpha r \beta n} \bar{x}_{\alpha s} \bar{x}_{\beta m} \tag{25.6}$$

where

$$\bar{x}_{\alpha i} = x_{\alpha i} - X_i^{cm} \tag{25.7}$$

The displacement of the rigid body is measured from its center of mass to eliminate the coupling between the translational and rotational momentum equations. Its location is initialized by calculating it from the mesh:

$$X_i^{cm} = \frac{\sum_{\alpha}^{RB} M_{\alpha i \beta i} x_{\beta i}}{M} \qquad \text{(no sum on } i\text{)} \tag{25.8}$$

The initial velocities of the nodes are readily calculated for a rigid body from:

$$\dot{x}_{\alpha i} = \dot{X}_i^{cm} + e_{ijk} \omega_j A_{kn}(\theta) \bar{x}_{\alpha j} \tag{25.9}$$

where A the transformation from the rotated reference configuration to the global coordinate frame, θ the measure of the rotation of the body, and ω the angular velocity of the body in the global coordinate frame. At the initial time, A is the identity transformation. The reference configuration is assumed to be co-aligned with the global reference frame. For arbitrary orientations of the body, the inertia tensor must be transformed each time step based on the incremental rotations using the standard rules of second-order tensors:

$$J_{ij}^{n+1} = A(\Delta\theta)_{ik} A(\Delta\theta)_{jm} J_{km}^{n} \tag{25.10}$$

where J^{n+1} is the inertia tensor components in the global frame. The transformation matrix A is not stored since the formulation is incremental. The forces and torques are found by summing over the nodes:

$$F_i^x = \sum_{\alpha}^{RB} f_{\alpha i} \tag{25.11}$$

$$F_i^{\omega} = e_{ijk} \bar{x}_{\alpha j} f_{\alpha k} \tag{25.12}$$

The summations are performed over all of nodes in the rigid body, including the nodes on the boundary between the rigid body and a nonlinear finite element mesh. The summation automatically accounts for all of the forces (concentrated loads, gravity, impact forces, surface traction, etc.) including the interface forces between the rigid body and any contiguous mesh. It

is the simplicity of the force and torque accumulations that makes rigid bodies so computationally attractive.

After calculating the rigid body accelerations from Equation (25.3) and (25.4), the rigid body velocities are updated as described in Section 24. The incremental rotation matrix of Equation (25.10) is calculated using the Hughes-Winget algorithm:

$$\Delta\theta_i^{(n+1)} \cong \Delta t^{n+1/2}\omega_i^{(n+1/2)} \tag{25.13}$$

$$A^{(n+1)}\left(\Delta\theta\right)_{ij} \cong \delta_{ij} + \left(\delta_{ik} - \frac{1}{2}\Delta S_{ik}\right)^{-1}\Delta S_{kj} = \delta_{ij} + \frac{1}{2D}\left(2\delta_{ik} + \Delta S_{ik}\right)\Delta S_{kj} \tag{25.14}$$

$$\Delta S_{ij} = e_{ikj}\Delta\theta_k^{(n+1)} \tag{25.15}$$

$$2D = 2 + \frac{1}{2}\left(\Delta\theta_1^2 + \Delta\theta_2^2 + \Delta\theta_3^2\right) \tag{25.16}$$

The coordinates of the nodes are incrementally updated

$$x_{\alpha i}^{(n+1)} = x_{\alpha i}^{(n)} + \left(X_i^{\,cm(n+1)} - X_i^{\,cm(n)}\right) + \left(A\left(\Delta\theta^{(n+1)}\right)_{ij} - \delta_{ij}\right)\overline{x}_{\alpha j}^{(n)} \tag{25.17}$$

where

$$\overline{x}^{(n)} = x^{(n)} - X^{\,cm(n)}$$

The velocities of the nodes are calculated by differencing the coordinates

$$\dot{x}_{\alpha i}^{(n+1/2)} = \frac{\left(x_{\alpha i}^{(n+1)} - x_{\alpha i}^{(n)}\right)}{\Delta t^{n+1/2}} \tag{25.18}$$

A direct integration of the rigid body accelerations into velocity and displacements is not used for two reasons: (1) calculating the rigid body accelerations of the nodes is more expensive than the current algorithm, and (2) the second-order accuracy of the central difference integration method would introduce distortion into the rigid bodies. Since the accelerations are not needed within the program, they are calculated by a post-processor using a difference scheme similar to the above.

25.1 Rigid Body Joints

The joints for the rigid bodies in LS-DYNA, see Figure 25.1, are implemented using the penalty method. Given a constraint equation $C\left(x_i, x_j\right) = 0$, for nodes i and j, the penalty function added to the Lagrangian of the system is $-1/2kC^2\left(x_i, x_j\right)$. The resulting nodal forces are:

$$f_i = -kC\left(x_i, x_j\right)\frac{\partial C\left(x_i, x_j\right)}{\partial x_i} \tag{25.19}$$

$$f_j = -kC\left(x_i, x_j\right)\frac{\partial C\left(x_i, x_j\right)}{\partial x_j} \tag{25.20}$$

The forces acting at the nodes have to convert into forces acting on the rigid bodies. Recall that velocities of a node i is related to the velocity of the center of mass of a rigid body by Equation (25.9). By using Equation (25.9) and virtual power arguments, it may be shown that the generalized forces are:

$$F_i^x = f_i \tag{25.21}$$

$$F_i^\omega = e_{ijk}\overline{x}_j f_k \tag{25.22}$$

which are the forces and moments about the center of mass.

The magnitude of the penalty stiffness k is chosen so that it does not control the stable time step size. For the central difference method, the stable time step Δt is restricted by the condition that,

$$\Delta t = \frac{2}{\Omega} \tag{25.23}$$

where Ω is the highest frequency in the system. The six vibrational frequencies associated with each rigid body are determined by solving their eigenvalue problems assuming $k=1$. For a body with m constraint equations, the linearized equations of the translational degrees of freedom are

$$M\ddot{X} + mkX = 0 \tag{25.24}$$

and the frequency is $\sqrt{mk/M}$ where M is the mass of the rigid body. The corresponding rotational equations are

$$J\ddot{\theta} + K\theta = 0 \tag{25.25}$$

J is the inertia tensor and K is the stiffness matrix for the moment contributions from the penalty constraints. The stiffness matrix is derived by noting that the moment contribution of a constraint may be approximated by

$$F^x = -kr_i \times \left(\theta \times r_i\right) \tag{25.26}$$

$$r_i = x_i - X^{cm} \tag{25.27}$$

and noting the identity,

$$A\times\left(B\times C\right)=\left|A\cdot C-A\otimes C\right|B \tag{25.28}$$

so that

$$K=\sum_{i=1}^{m}k\left[r_i\cdot r_i-r_i\otimes r_i\right] \tag{25.29}$$

The rotational frequencies are the roots of the equation $\det\left|K-\Omega^2 J\right|=0$, which is cubic in Ω^2. Defining the maximum frequency over all rigid bodies for k=1 as $\Omega_{\max}$, and introducing a time step scale factor $TSSF$, the equation for k is

$$k\le\left(\frac{2TSSF}{\Delta t\Omega_{\max}}\right)^2 \tag{25.30}$$

The joint constraints are defined in terms of the displacements of individual nodes. Regardless of whether the node belongs to a solid element or a structural element, only its translational degrees of freedom are used in the constraint equations.

A spherical joint is defined for nodes i and j by the three constraint equations,

$$x_{1i}-x_{1j}=0\quad x_{2i}-x_{2j}=0\quad x_{3i}-x_{3j}=0 \tag{25.31}$$

and a revolute joint, which requires five constraints, is defined by two spherical joints, for a total of six constraint equations. Since a penalty formulation is used, the redundancy in the joint constraint equations is unimportant. A cylindrical joint is defined by taking a revolute joint and eliminating the penalty forces along the direction defined by the two spherical joints. In a similar manner, a planar joint is defined by eliminating the penalty forces that are perpendicular to the two spherical joints.

The translational joint is a cylindrical joint that permits sliding along its axis, but not rotation. An additional pair of nodes is required off the axis to supply the additional constraint. The only force active between the extra nodes acts in the direction normal to the plane defined by the three pairs of nodes.

The universal joint is defined by four nodes. Let the nodes on one body be i and k, and the other body, j and l. Two of them, i and j, are used to define a spherical joint for the first three constraint equations. The fourth constraint equation is,

$$C\left(x_i,x_j,x_k,x_l\right)=\left(x_k-x_i\right)\cdot\left(x_i-x_j\right)=0 \tag{25.32}$$

and is differentiated to give the penalty forces $f_n=-kC\dfrac{\partial C}{\partial x_n}$, where n ranges over the four nodes numbers.

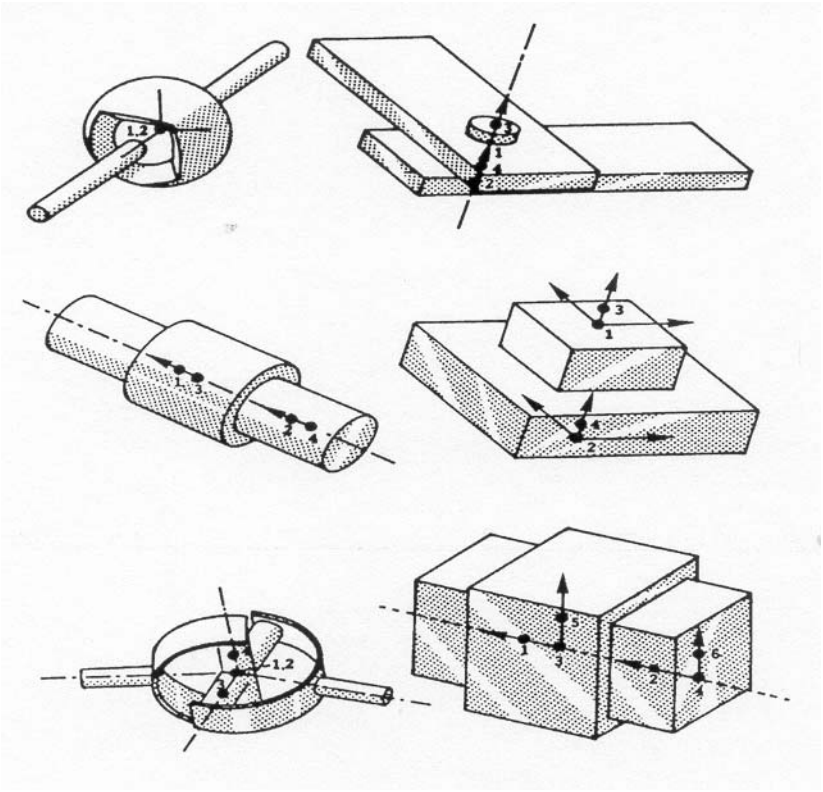


Figure 25.1. Rigid body joints in LS-DYNA.

25.2 Deformable to Rigid Material Switching

Occasionally in practice, long duration, large rigid body motions arise that are prohibitively expensive to simulate if the elements in the model are deformable. Such a case could occur possibly in automotive rollover where the time duration of the rollover would dominate the cost relative to the post impact response that occurs much later.

To permit such simulations to be efficiently handled a capability to switch a subset of materials from deformable to rigid and back to deformable is available. In practice the suspension system and tires would remain deformable A flag is set in the input to let LS-DYNA know that all materials in the model have the potential to become rigid bodies sometime during the calculation. When this flag is set a cost penalty is incurred on vector machines since the blocking of materials in the element loops will be based on the part ID rather than the material type ID. Normally this cost is insignificant relative to the cost reduction due to this unique feature.

For rigid body switching to work properly the choice of the shell element formulation is critical. The Hughes-Liu elements cannot currently be used for two reasons. First, since these elements compute the strains from the rotations of the nodal fiber vectors from one cycle to the next, the nodal fiber vectors should be updated with the rigid body motions and this is not done. Secondly, the stresses are stored in the global system as opposed to the co-rotational system. Therefore, the stresses would also need to be transformed with the rigid body motions or zeroed out. The co-rotational elements of Belytschko and co-workers do not reference nodal fibers for the strain computations and the stresses are stored in the co-rotational coordinate system which

eliminates the need for the transformations; consequently, these elements can be safely used. The membrane elements and airbag elements are closely related to the Belytschko shells and can be safely used with the switching options.

The beam elements have nodal triads that are used to track the nodal rotations and to calculate the deformation displacements from one cycle to the next. These nodal triads are updated every cycle with the rigid body rotations to prevent non-physical behavior when the rigid body is switched back to deformable. This applies to all beam element formulations in LS-DYNA. The Belytschko beam formulations are preferred for the switching options for like the shell elements, the Hughes-Liu beams keep the stresses in the global system. Truss elements like the membrane elements are trivially treated and pose no difficulties.

The brick elements store the stresses in the global system and upon switching the rigid material to deformable the element stresses are zeroed to eliminate spurious behavior.

The implementation addresses many potential problems and has worked well in practice. The current restrictions can be eliminated if the need arises and anyway should pose no insurmountable problems. We will continue to improve this capability if we find that it is becoming a popular option.

25.3 Rigid Body Welds

The weld capability in LS-DYNA is based on rigid body dynamics. Each weld is defined by a set of nodal points which moves rigidly with six degrees of freedom until a failure criteria is satisfied. Five weld options are implemented including:

- Spot weld.
- Fillet weld
- Butt weld
- Cross fillet weld
- General weld

Welds can fail three ways: by ductile failure which is based on the effective plastic strain, by brittle failure which is based on the force resultants acting on the rigid body weld, and by a failure time which is specified in the input. When effective plastic strain is used the weld fails when the nodal plastic strain exceeds the input value. A least squares algorithm is used to generate the nodal values of plastic strains at the nodes from the element integration point values. The plastic strain is integrated through the element and the average value is projected to the nodes with a least square fit. In the resultant based brittle failure the resultant forces and moments on each node of the weld are computed. These resultants are checked against a failure criterion which is expressed in terms of these resultants. The forces may be averaged over a user specified number of time steps to eliminate breakage due to spurious noise. After all nodes of a weld are released the rigid body is removed from the calculation.

Spotwelds are shown in Figure 25.2. Spotweld failure due to plastic straining occurs when the effective nodal plastic strain exceeds the input value, ε_{fail}^{p}. This option can model the tearing out of a spotweld from the sheet metal since the plasticity is in the material that surrounds the spotweld, not the spotweld itself. This option should only be used for the material models related to metallic plasticity and can result is slightly increased run times.

Brittle failure of the spotwelds occurs when:

$$\left(\frac{\max(f_n,0)}{S_n}\right)^n+\left(\frac{|f_s|}{S_s}\right)^m\ge 1$$

where f_n and f_s are the normal and shear interface force. Component f_n contributes for tensile values only. When the failure time, t_f, is reached the nodal rigid body becomes inactive and the constrained nodes may move freely. In Figure 25.2 the ordering of the nodes is shown for the 2 and 3 noded spotwelds. This order is with respect to the local coordinate system where the local z axis determines the tensile direction. The nodes in the spotweld may coincide but if they are offset the local system is not needed since the z-axis is automatically oriented based on the locations of node 1, the origin, and node 2. The failure of the 3 noded spotweld may occur gradually with first one node failing and later the second node may fail. For n noded spotwelds the failure is progressive starting with the outer nodes (1 and n) and then moving inward to nodes 2 and $n-1$. Progressive failure is necessary to preclude failures that would create new rigid bodies.

Ductile fillet weld failure, due to plastic straining, is treated identically to spotweld failure. Brittle failure of the fillet welds occurs when:

$$\beta\sqrt{\sigma_n^2+3\left(\tau_n^2+\tau_t^2\right)}\ge\sigma_f$$

where

σ_n	=	normal stress
τ_n	=	shear stress in direction of weld (local y)
τ_t	=	shear stress normal to weld (local x)
σ_f	=	failure stress
β	=	failure parameter

Component σ_n is nonzero for tensile values only. When the failure time, t_f, is reached the nodal rigid body becomes inactive and the constrained nodes may move freely. In Figure 25.3 the ordering of the nodes is shown for the 2 node and 3 node fillet welds. This order is with respect to the local coordinate system where the local z axis determines the tensile direction. The nodes in the fillet weld may coincide. The failure of the 3 node fillet weld may occur gradually with first one node failing and later the second node may fail.

In Figure 25.4 the butt weld is shown. Ductile butt weld failure, due to plastic straining, is treated identically to spotweld failure. Brittle failure of the butt welds occurs when:

$$\beta\sqrt{\sigma_n^2+3\left(\tau_n^2+\tau_t^2\right)}\ge\sigma_f$$

where

σ_n	=	normal stress

τ_n = shear stress in direction of weld (local y)

τ_t = shear stress normal to weld (local z)

σ_f = failure stress

β = failure parameter

Component σ_n is nonzero for tensile values only. When the failure time, t_f , is reached the nodal rigid body becomes inactive and the constrained nodes may move freely. The nodes in the butt weld may coincide.

The cross fillet weld and general weld are shown in Figures 25.5 and 25.6, respectively. The treatment of failure for these welds is based on the formulation for the fillet and butt welds.

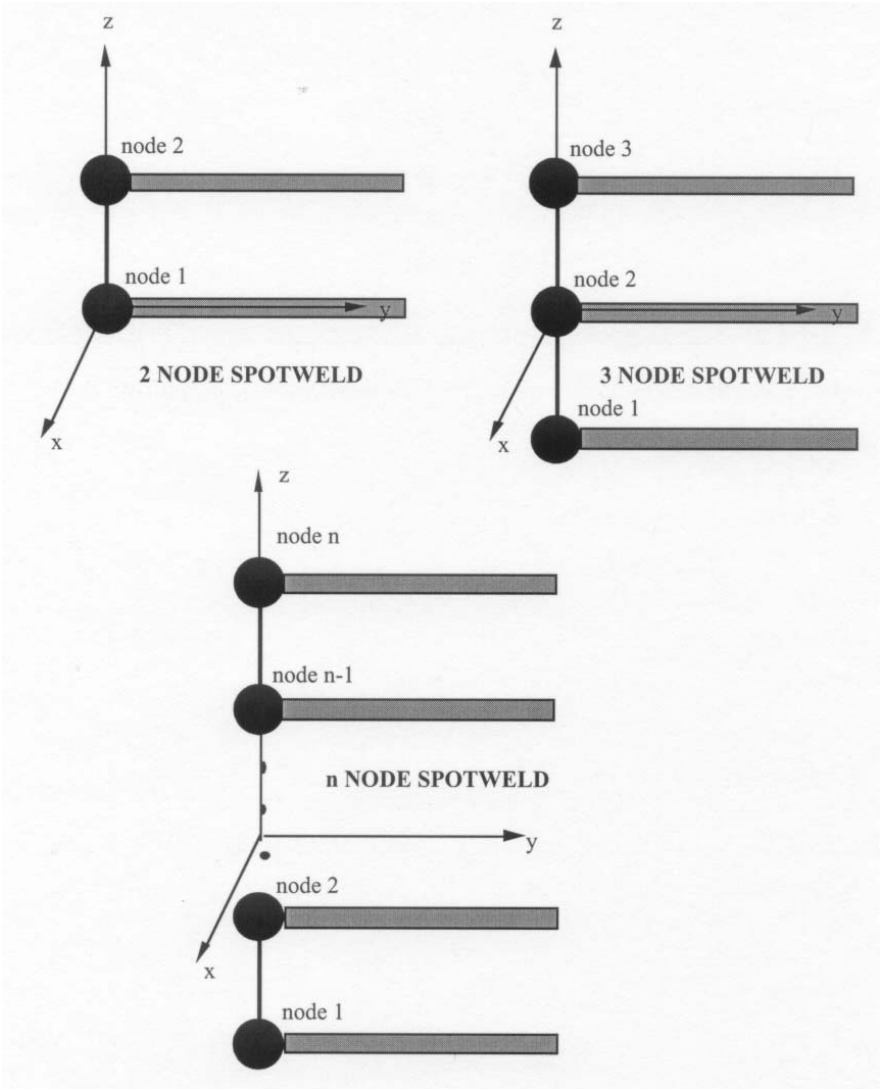


Figure 25.2. Nodal ordering and orientation of the local coordinate system is important for determining spotweld failure.

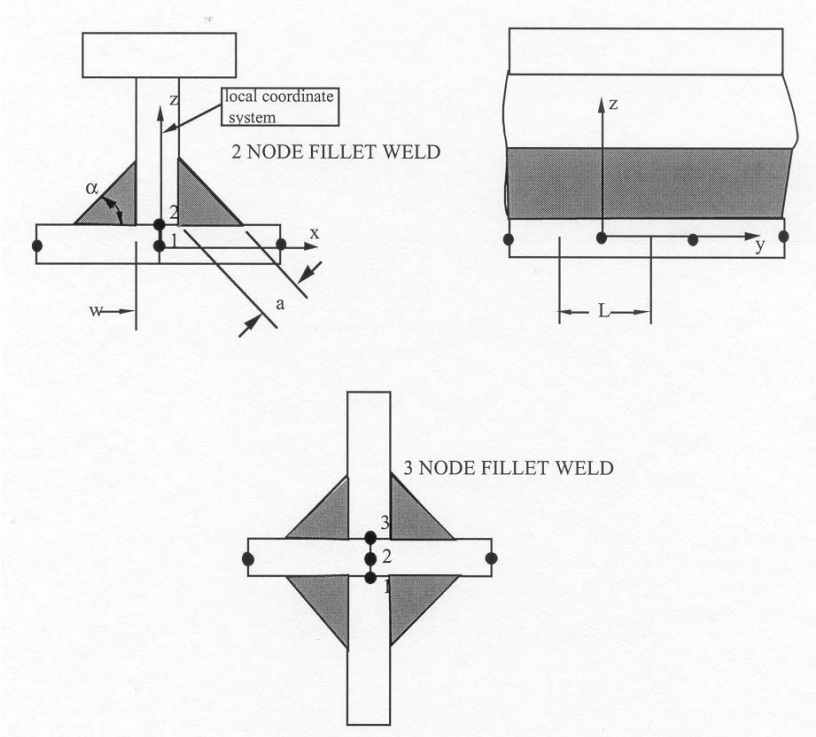


Figure 25.3. Nodal ordering and orientation of the local coordinate system is shown for fillet weld failure.

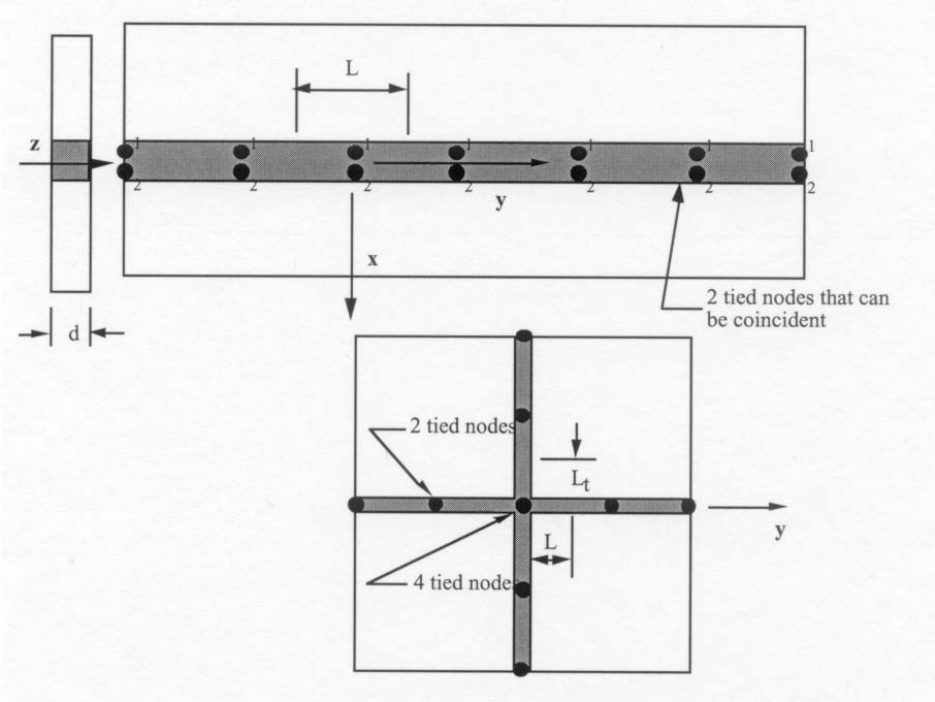


Figure 25.4. Orientation of the local coordinate system and nodal ordering is shown for butt weld failure.

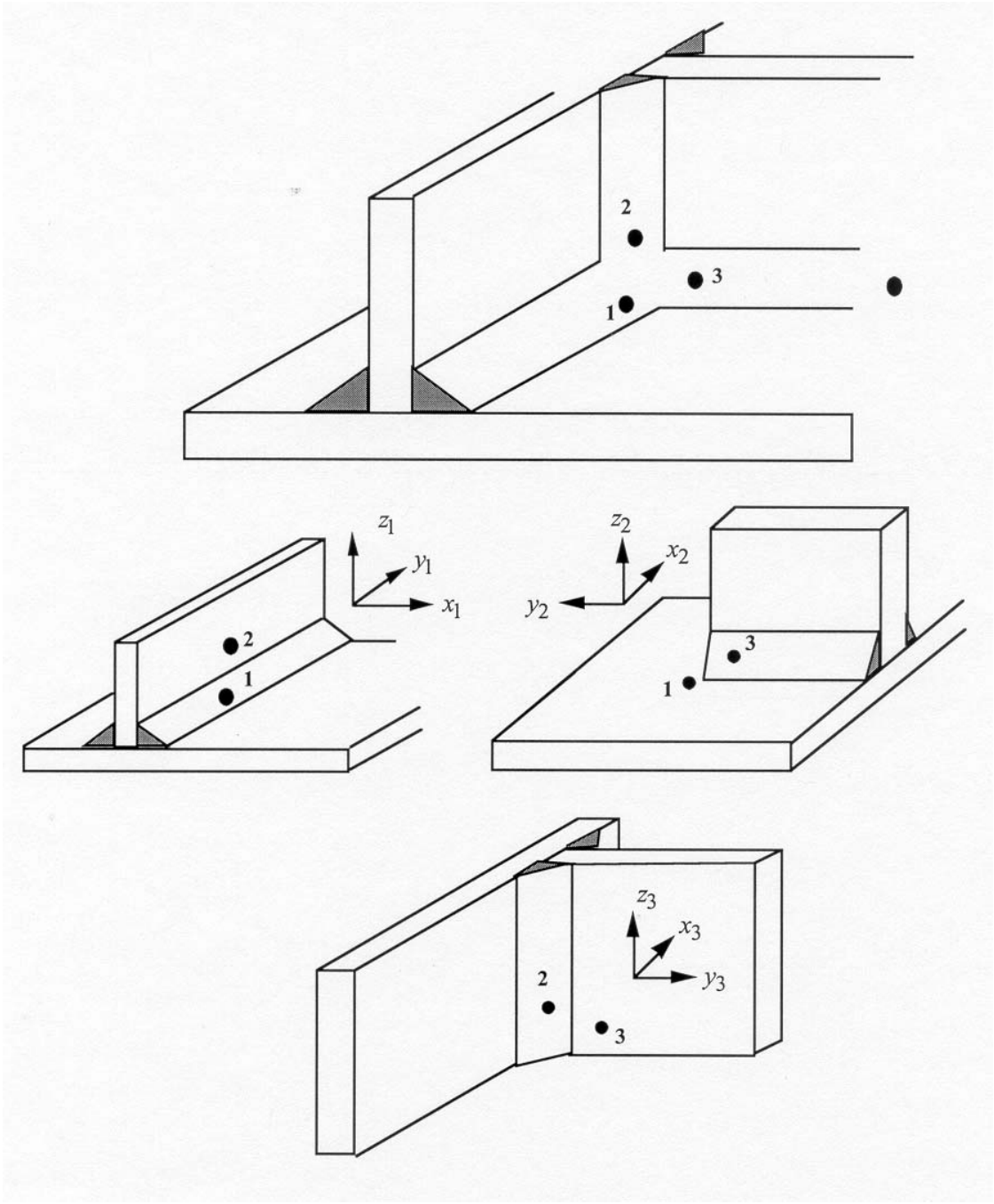


Figure 25.5. A simple cross fillet weld illustrates the required input. Here NFW=3 with nodal pairs (A=2, B=1), (A=3, B=1), and (A=3, B=2). The local coordinate axes are shown. These axes are fixed in the rigid body and are referenced to the local rigid body coordinate system which tracks the rigid body rotation.

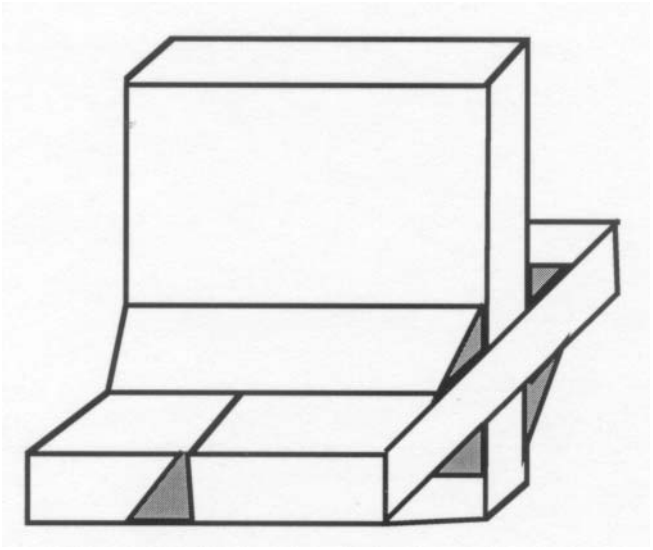


Figure 25.6. A general weld is a mixture of fillet and butt welds.

26. CONTACT-IMPACT ALGORITHM

26.1 Introduction

The treatment of sliding and impact along interfaces has always been an important capability in DYNA3D and more recently in LS-DYNA. Three distinct methods for handling this have been implemented, which we will refer to as the kinematic constraint method, the penalty method, and the distributed parameter method. Of these, the first approach is now used for tying interfaces. The relative merits of each approach are discussed below.

Interfaces can be defined in three dimensions by listing in arbitrary order all triangular and quadrilateral segments that comprise each side of the interface. One side of the interface is designated as the slave side, and the other is designated as the master side. Nodes lying in those surfaces are referred to as slave and master nodes, respectively. In the symmetric penalty method, this distinction is irrelevant, but in the other methods the slave nodes are constrained to slide on the master surface after impact and must remain on the master surface until a tensile force develops between the node and the surface.

Today, automatic contact definitions are commonly used. In this approach the slave and master surfaces are generated internally within LS-DYNA from the part ID's given for each surface. For automotive crash models it is quite common to include the entire vehicle in one single surface contact definition where the all the nodes and elements within the interface can interact.

26.2 Kinematic Constraint Method

The kinematic constraint method which uses the impact and release conditions of Hughes et al., [1976] was implemented first in DYNA2D [Hallquist 1976b] and finally extended to three dimensions in DYNA3D. Constraints are imposed on the global equations by a transformation of the nodal displacement components of the slave nodes along the contact interface. This transformation has the effect of eliminating the normal degree of freedom of nodes. To preserve the efficiency of the explicit time integration, the mass is lumped to the extent that only the global degrees of freedom of each master node are coupled. Impact and release conditions are imposed to insure momentum conservation. The release conditions are of academic interest and were quickly removed from the coding.

Problems arise with this method when the master surface zoning is finer than the slave surface zoning as shown in two dimensions in Figure 26.1. Here, certain master nodes can penetrate through the slave surface without resistance and create a kink in the slide line. Such kinks are relatively common with this formulation, and, when interface pressures are high, these kinks occur whether one or more quadrature points are used in the element integration. It may be argued, of course, that better zoning would minimize such problems; but for many problems that are of interest, good zoning in the initial configuration may be very poor zoning later. Such is the case, for example, when gaseous products of a high explosive gas expand against the surface of a structural member.

26.3 Penalty Method

The penalty method is used in the explicit programs DYNA2D and DYNA3D as well as in the implicit programs NIKE2D and NIKE3D. The method consists of placing normal interface springs between all penetrating nodes and the contact surface. With the exception of the spring stiffness matrix which must be assembled into the global stiffness matrix, the implicit and explicit treatments are similar. The NIKE2D/3D and DYNA2D/3D programs compute a

unique modulus for the element in which it resides. In our opinion, pre-empting user control over this critical parameter greatly increases the success of the method.

Quite in contrast to the nodal constraint method, the penalty method approach is found to excite little if any mesh hourglassing. This lack of noise is undoubtedly attributable to the symmetry of the approach. Momentum is exactly conserved without the necessity of imposing impact and release conditions. Furthermore, no special treatment of intersecting interfaces is required, greatly simplifying the implementation.

Currently three implementations of the penalty algorithm are available:

1. Standard Penalty Formulation
2. Soft Constraint Penalty Formulation, which has been implemented to treat contact between bodies with dissimilar material properties (e.g. steel-foam). Stiffness calculation and its update during the simulation differs from the Standard Penalty Formulation.
3. Segment-based Penalty Formulation, it is a powerful contact algorithm whose logic is a slave segment-master segment approach instead of a traditional slave node-master segment approach. This contact has proven very useful for airbag self-contact during inflation and complex contact conditions.

In the standard penalty formulation, the interface stiffness is chosen to be approximately the same order of magnitude as the stiffness of the interface element normal to the interface. Consequently the computed time step size is unaffected by the existence of the interfaces. However, if interface pressures become large, unacceptable penetration may occur. By scaling up the stiffness and scaling down the time step size, we may still solve such problems using the penalty approach. Since this increases the number of time steps and hence the cost, a sliding-only option has been implemented for treating explosive-structure interaction problems thereby avoiding use of the penalty approach. This latter option is based on a specialization of the third method described below.

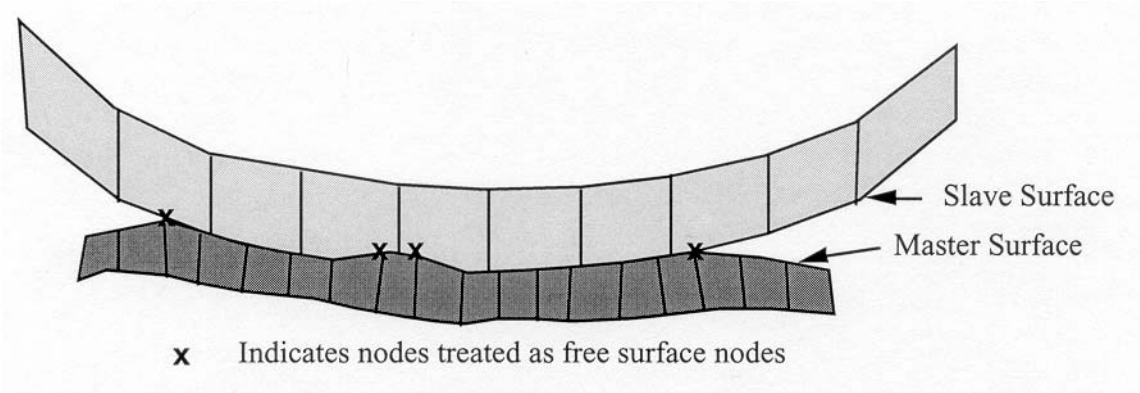


Figure 26.1. Nodes of the master slide surface designated with an “x” are treated as free surface nodes in the nodal constraint method.

26.4 Distributed Parameter Method

This method is used in DYNA2D, and a specialization of it is the sliding only option in DYNA3D. Motivation for this approach came from the TENSOR [Burton et. al., 1982] and HEMP [Wilkins 1964] programs which displayed fewer mesh instabilities than DYNA2D with the nodal constraint algorithm. The first DYNA2D implementation of this last algorithm is described in detail by Hallquist [1978]. Since this early publication, the method has been moderately improved but the major ideas remain the same.

In the distributed parameter formulation, one-half the slave element mass of each element in contact is distributed to the covered master surface area. Also, the internal stress in each element determines a pressure distribution for the master surface area that receives the mass. After completing this distribution of mass and pressure, we can update the acceleration of the master surface. Constraints are then imposed on slave node accelerations and velocities to insure their movement along the master surface. Unlike the finite difference hydro programs, we do not allow slave nodes to penetrate; therefore we avoid "put back on" logic. In another simplification, our calculation of the slave element relative volume ignores any intrusion of the master surfaces. The HEMP and TENSOR codes consider the master surface in this calculation.

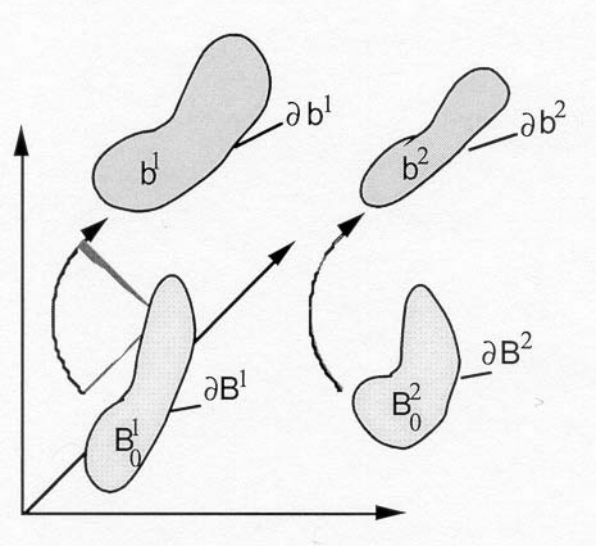


Figure 26.2. Reference and deformed configuration.

26.5 Preliminaries

Consider the time-dependent motion of two bodies occupying regions B^1 and B^2 in their undeformed configuration at time zero. Assume that the intersection

$$B^1 \cap B^2 = \emptyset \tag{26.1}$$

is satisfied. Let ∂B^1 and ∂B^2 denote the boundaries of B^1 and B^2, respectively. At some later time, these bodies occupy regions b^1 and b^2 bounded by ∂b^1 and ∂b^2 as shown in Figure 26.2. Because the deformed configurations cannot penetrate,

$$(b^1 - \partial b^1) \cap b^2 = \emptyset \tag{26.2}$$

As long as $(\partial b^1 \cap \partial b^2) = \emptyset$, the equations of motion remain uncoupled. In the foregoing and following equations, the right superscript α $(= 1, 2)$ denotes the body to which the quantity refers.

Before a detailed description of the theory is given, some additional statements should be made concerning the terminology. The surfaces ∂b^1 and ∂b^2 of the discretized bodies b^1 and b^2 become the master and slave surfaces respectively. Choice of the master and slave surfaces is arbitrary when the symmetric penalty treatment is employed. Otherwise, the more coarsely meshed surface should be chosen as the master surface unless there is a large difference in mass densities in which case the side corresponding to the material with the highest density is recommended. Nodal points that define ∂b^1 are called master nodes and nodes that define ∂b^2 are called slave nodes. When $(\partial b^1 \cap \partial b^2) \neq \emptyset$, the constraints are imposed to prevent penetration. Right superscripts are implied whenever a variable refers to either the master surface ∂b^1, or slave surface, ∂b^2; consequently, these superscripts are dropped in the development which follows.

26.6 Slave Search

The slave search is common to all interface algorithms implemented in DYNA3D. This search finds for each slave node its nearest point on the master surface. Lines drawn from a slave node to its nearest point will be perpendicular to the master surface, unless the point lies along the intersection of two master segments, where a segment is defined to be a 3- or 4-node element of a surface.

Consider a slave node, n_s, sliding on a piecewise smooth master surface and assume that a search of the master surface has located the master node, m_s, lying nearest to n_s. Figure 26.3 depicts a portion of a master surface with nodes m_s and n_s labeled. If m_s and n_s do not coincide, n_s can usually be shown to lie in a segment s_1 via the following tests:

$$\begin{aligned} (c_i \times s) \cdot (c_i \times c_{i+1}) &> 0 \\ (c_i \times s) \cdot (s \times c_{i+1}) &> 0 \end{aligned} \tag{26.3}$$

where vector c_i and c_{i+1} are along edges of s_1 and point outward from m_s. Vector $\boldsymbol{s}$ is the projection of the vector beginning at m_s, ending at n_s, and denoted by $\boldsymbol{g}$, onto the plane being examined (see Figure 26.4).

$$s = g - (g \cdot m)m \tag{26.4}$$

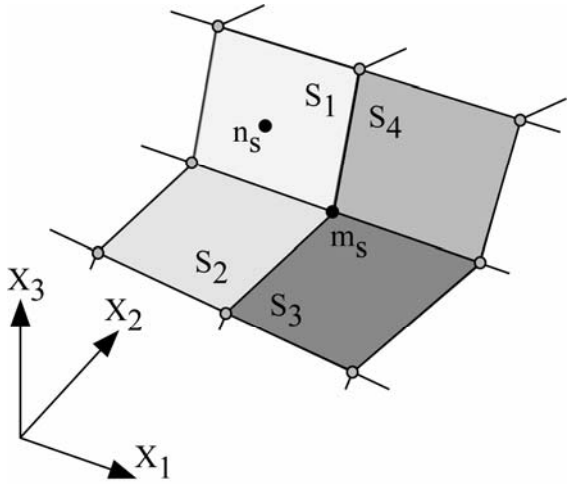


Figure 26.3. In this figure, four master segments can harbor slave node n_s given that m_s is the nearest master node.

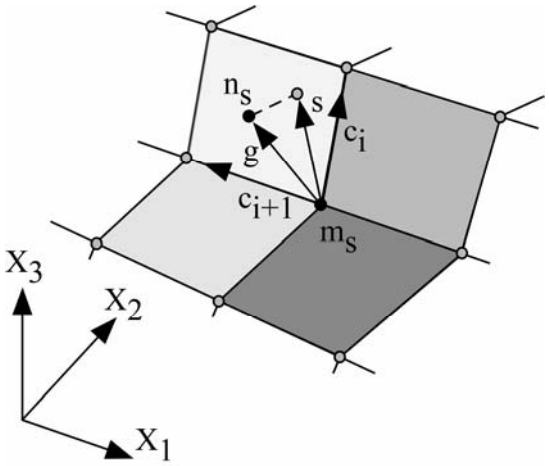


Figure 26.4. Projection of $\boldsymbol{g}$ onto master segment s_1

where for segment s_1

$$m = \frac{c_i \times c_{i+1}}{\left| c_i \times c_{i+1} \right|} \tag{26.5}$$

Since the sliding constraints keep n_s close but not necessarily on the master surface and since n_s may lie near or even on the intersection of two master segments, the inequalities of Equation (26.3) may be inconclusive, i.e., they may fail to be satisfied or more than one may give positive results. When this occurs n_s is assumed to lie along the intersection which yields the maximum value for the quantity

$$\frac{g \cdot c_i}{|c_i|} \quad i = 1, 2, 3, 4, \ldots \tag{26.6}$$

When the contact surface is made up of badly shaped elements, the segment apparently identified as containing the slave node actually may not, as shown in Figure 26.5.

Assume that a master segment has been located for slave node n_s and that n_s is not identified as lying on the intersection of two master segments. Then the identification of the contact point, defined as the point on the master segment which is nearest to n_s, becomes nontrivial. For each master surface segment, s_1 is given the parametric representation of Equation (23.2), repeated here for clarity:

$$r \quad f_1(\xi,\eta)i_1 \quad + \quad f_2(\xi,\eta)i_2 \quad + \quad f_3(\xi,\eta)i_3 \tag{23.2}$$

where

$$f_i(\xi,\eta) = \sum_{j=1}^{4} \phi_j x_i^j \tag{23.3}$$

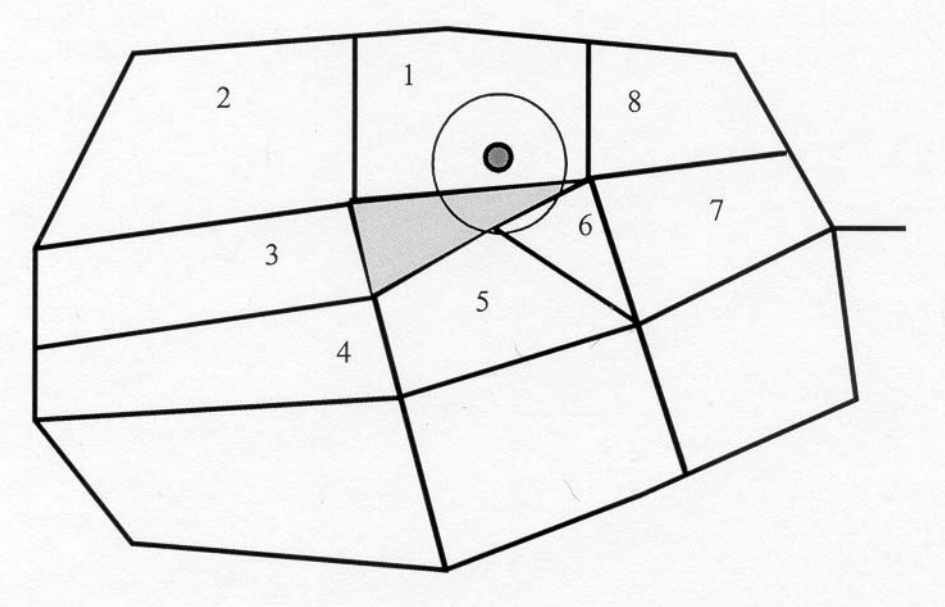


Figure 26.5. When the nearest node fails to contain the segment that harbors the slave node, segments numbered 1-8 are searched in the order shown.

Note that r_1 is at least once continuously differentiable and that

$$\frac{\partial r}{\partial \xi} \times \frac{\partial r}{\partial \eta} \neq 0 \tag{26.7}$$

Thus ***r*** represents a master segment that has a unique normal whose direction depends continuously on the points of s_1.

Let $\boldsymbol{t}$ be a position vector drawn to slave node n_s and assume that the master surface segment s_1 has been identified with n_s. The contact point coordinates (ξ_c, η_c) on s_1 must satisfy

$$\frac{\partial r}{\partial \xi}(\xi_c, \eta_c) \cdot \left[t - r(\xi_c, \eta_c) \right] = 0 \tag{26.8a}$$

$$\frac{\partial r}{\partial \eta}(\xi_c, \eta_c) \cdot \left[t - r(\xi_c, \eta_c) \right] = 0 \tag{26.8b}$$

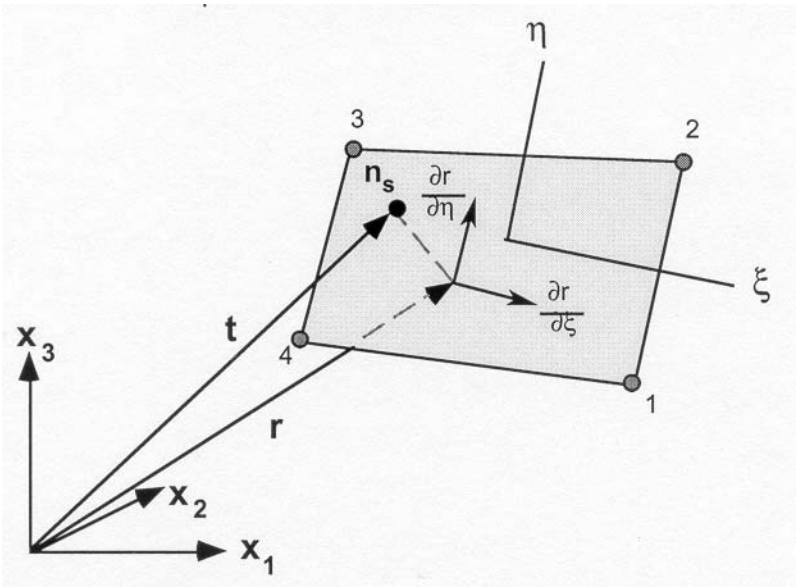


Figure 26.6. Location of contact point when n_s lies above master segment.

The physical problem is illustrated in Figure 26.6, which shows n_s lying above the master surface. Equations (26.8a) and (26.8b) are readily solved for ξ_c and η_c. One way to accomplish this is to solve Equation (26.8a) for ξ_c in terms of η_c, and substitute the results into Equation (26.8b). This yields a cubic equation in η_c which is presently solved numerically in LS-DYNA. In the near future, we hope to implement a closed form solution for the contact point.

The equations are solved numerically. When two nodes of a bilinear quadrilateral are collapsed into a single node for a triangle, the Jacobian of the minimization problem is singular at the collapsed node. Fortunately, there is an analytical solution for triangular segments since three points define a plane. Newton-Rap son iteration is a natural choice for solving these simple nonlinear equations. The method diverges with distorted elements unless the initial guess is accurate. An exact contact point calculation is critical in post-buckling calculations to prevent the solution from wandering away from the desired buckling mode.

Three iterations with a least-squares projection are used to generate an initial guess:

$$\xi_0 = 0, \qquad \eta_0 = 0, \tag{26.9}$$

$$\begin{bmatrix} r_{,\xi} \\ r_{,\eta} \end{bmatrix} \begin{bmatrix} r_{,\xi} & r_{,\eta} \end{bmatrix} \begin{Bmatrix} \Delta\xi \\ \Delta\eta \end{Bmatrix} = \begin{bmatrix} r_{,\xi} \\ r_{,\eta} \end{bmatrix} \left\{ r\left(\xi_{i,}\eta_i\right) - t \right\},$$

$$\xi_{i+1} = \xi_{i+1} + \Delta\xi\,, \qquad \eta_{i+1} = \eta_i + \Delta\eta$$

followed by the Newton-Raphson iterations which are limited to ten iterations, but which usually converges in four or less.

$$[H] \begin{Bmatrix} \Delta\xi \\ \Delta\eta \end{Bmatrix} = - \begin{Bmatrix} r_{,\xi} \\ r_{,\eta} \end{Bmatrix} \left\{ r\left(\xi_{i,}\eta_i\right) - t \right\}\,, \tag{26.10}$$

$$[H] = \begin{Bmatrix} r_{,\xi} \\ r_{,\eta} \end{Bmatrix} \begin{bmatrix} r_{,\xi} & r_{,\eta} \end{bmatrix} + \begin{bmatrix} 0 & r \cdot r_{,\xi\eta} \\ r \cdot r_{,\xi\eta} & 0 \end{bmatrix},$$

$$\xi_{i+1} = \xi_i + \Delta\xi\,, \quad \eta_{i+1} = \eta_i + \Delta\eta\,.$$

In concave regions, a slave node may have isoparametric coordinates that lie outside of the [-1, +1] range for all of the master segments, yet still have penetrated the surface. A simple strategy is used for handling this case, but it can fail. The contact segment for each node is saved every time step. If the slave node contact point defined in terms of the isoparametric coordinates of the segment, is just outside of the segment, and the node penetrated the isoparametric surface, and no other segment associated with the nearest neighbor satisfies the inequality test, then the contact point is assumed to occur on the edge of the segment. In effect, the definition of the master segments is extended so that they overlap by a small amount. In the hydrocode literature, this approach is similar to the slide line extensions used in two dimensions. This simple procedure works well for most cases, but it can fail in situations involving sharp concave corners.

26.7 Sliding With Closure and Separation

26.7.1 Standard Penalty Formulation

Because this is perhaps the most general and most used interface algorithm, we choose to discuss it first. In applying this penalty method, each slave node is checked for penetration through the master surface. If the slave node does not penetrate, nothing is done. If it does penetrate, an interface force is applied between the slave node and its contact point. The magnitude of this force is proportional to the amount of penetration. This may be thought of as the addition of an interface spring.

Penetration of the slave node n_s through the master segment which contains its contact point is indicated if

$$l = n_i \times [t - r(\xi_c, \eta_c)] < 0 \tag{26.11}$$

where

$$n_i = n_i(\xi_c, \eta_c)$$

is normal to the master segment at the contact point.

If slave node n_s has penetrated through master segment s_i, we add an interface force vector f_s:

$$f_s = -lk_i n_i \qquad \text{if } l < 0 \tag{26.12}$$

to the degrees of freedom corresponding to n_s and

$$f_m^i = \phi_i(\xi_c, \eta_c) f_s \qquad \text{if } l < 0 \tag{26.13}$$

to the four nodes ($i = 1,2,3,4$) that comprise master segment s_i. The stiffness factor k_i for master segment s_i is given in terms of the bulk modulus K_i, the volume V_i, and the face area A_i of the element that contains s_i as

$$k_i = \frac{f_{si} K_i A_i^2}{V_i} \tag{26.14a}$$

for brick elements and

$$k_i = \frac{f_{si} K_i A_i}{\max(\textit{shell diagonal})} \tag{26.14b}$$

for shell elements where f_{si} is a scale factor for the interface stiffness and is normally defaulted to .10. Larger values may cause instabilities unless the time step size is scaled back in the time step calculation.

In LS-DYNA, a number of options are available for setting the penalty stiffness value. This is often an issue since the materials in contact may have drastically different bulk modulii. The calculational choices are:

- Minimum of the master segment and slave node stiffness. (default)
- Use master segment stiffness
- Use slave node value
- Use slave node value, area or mass weighted.
- As above but inversely proportional to the shell thickness.

The default may sometimes fail due to an excessively small stiffness. When this occurs it is necessary to manually scale the interface stiffness. Care must be taken not to induce an instability when such scaling is performed. If the soft material also has a low density, it may be necessary to reduce the scale factor on the computed stable time step.

26.7.2 Soft Constraint Penalty Formulation

Very soft materials have an undesired effect on the contact stiffness, lowering its value and ultimately causing excessive penetration. An alternative to put a scale factor on the contact stiffness for SOFT=0 is to use a Soft Constraint Penalty Formulation.

The idea behind this option is to eliminate the excessive penetration by using a different formulation for the contact stiffness.

In addition to the master and slave contact stiffness, an additional stiffness is calculated, which is based on the stability (Courant's criterion) of the local system comprised of two masses (segments) connected by a spring. This is the stability contact stiffness k_{cs} and is calculated by:

$$k_{cs}(t) = 0.5 \cdot SOFSCL \cdot m^* \cdot \left(\frac{1}{\Delta t_c(t)} \right) \tag{26.15}$$

where SOFSCL on Optional Card A of *CONTROL_CONTACT is the scale factor for the Soft Constraint Penalty Formulation, m^* is a function of the mass of the slave node and of the master nodes. Δt_{cs} is set to the initial solution timestep. If the solution time step grows, Δt_c is reset to the current time step to prevent unstable behavior.

A comparative check against the contact stiffness calculated with the traditional penalty formulation, $k_{soft=0}$, and in general[1] the maximum stiffness between the two is taken,

$$k_{soft=1} = \max\left\{k_{cs,} k_{soft=0}\right\} \tag{26.16}$$

26.7.3 Segment-based Penalty Formulation

Segment based contact is a general purpose shell and solid element penalty type contact algorithm. Segment based contact uses a contact stiffness similar to the SOFT=1 stiffness option, but the details are quite different.

$$k_{cs}(t) = 0.5 \cdot SLSFAC \cdot \left\{ \begin{matrix} SFS \\ or \\ SFM \end{matrix} \right\} \left(\frac{m_1 m_2}{m_1 + m_2} \right) \left(\frac{1}{\Delta t_c(t)} \right)^2 \tag{26.17}$$

Segment masses are used rather than nodal masses. Segment mass is equal to the element mass for shell segments and half the element mass for solid element segments. Like the Soft Constraint Penalty Formulation, *dt* is set to the initial solution time step which is updated if the solution time step grows larger to prevent unstable behavior. However, it differs from SOFT=1 in how *dt* is updated. *dt* is updated only if the solution time step grows by more than 5%. This allows *dt* to remain constant in most cases, even if the solution time step slightly grows.

[1] Various contact algorithms can have slightly different implementations.

26.8 Recent Improvements in Surface-to-Surface Contact

A number of recent changes have been made in the surface-to-surface contact including contact searching, accounting for thickness, and contact damping. These changes have been implemented primarily to aid in the analysis of sheet metal forming problems.

26.8.1 Improvements to the Contact Searching

In metal forming applications, problems with the contact searching were found when the rigid body stamping dies were meshed with elements having very poor aspect ratios. The nearest node algorithm described above can break down since the nearest node is not necessarily anywhere near the segment that harbors the slave node as is assumed in Figure 26.5 (see Figure 26.7). Such distorted elements are commonly used in rigid bodies in order to define the geometry accurately.

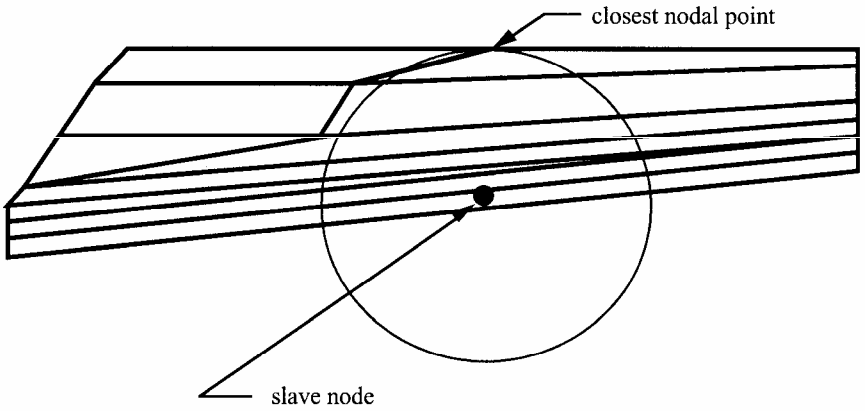


Figure 26.7. Failure to find the contact segment can be caused by poor aspect ratios in the finite element mesh.

To circumvent the problem caused by bad aspect ratios, an expanded searching procedure is used in which we attempt to locate the nearest segment rather than the nearest nodal point. We first sort the segments based on their centroids as shown in Figure 26.8 using a one-dimensional bucket sorting technique.

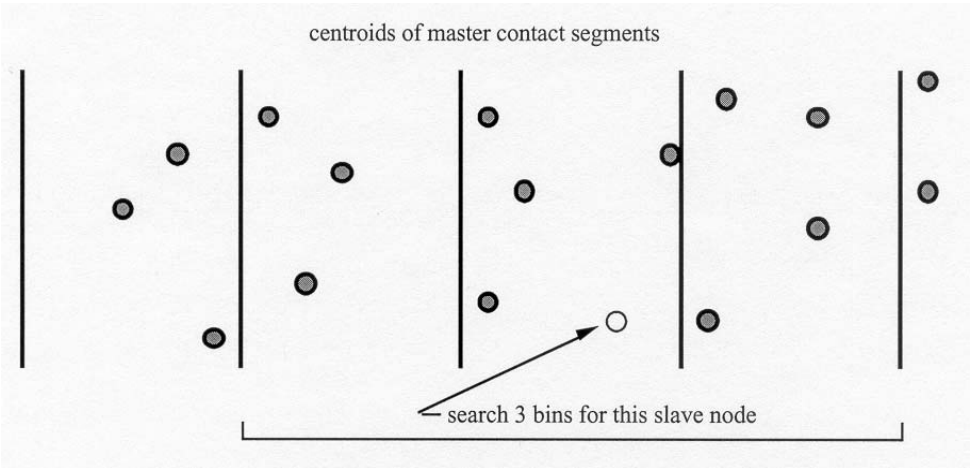


Figure 26.8 One-dimensional bucket sorting identifies the nearest segments for each slave node.

Once a list of possible candidates is identified for a slave node, it is necessary to locate the possible segments that contain the slave node of interest. For each quadrilateral segment, four points are constructed at the centroids of the four triangles each defined by 3 nodes as shown in Figure 26.9 where the black point is the centroid of the quadrilateral. These centroids are used to find the nearest point to the slave node and hence the nearest segment. The nodes of the three nearest segments are then examined to identify the three nearest nodes. Just one node from each segment is allowed to be a nearest node.

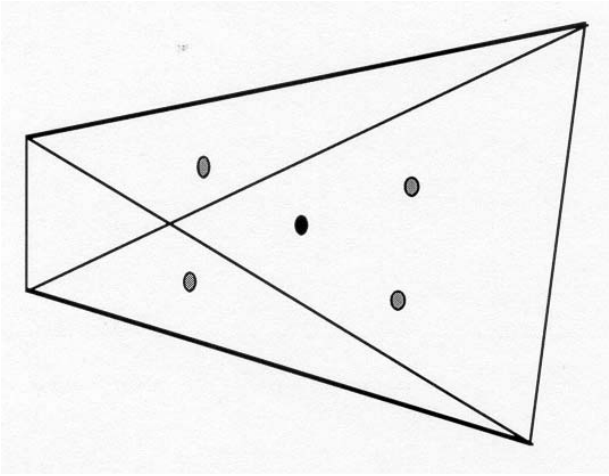


Figure 26.9. Interior points are constructed in the segments for determining the closest point to the slave node.

When the nearest segment fails to harbor the slave node, the adjacent segments are checked. The old algorithm checks the segments labeled 1-3 (Figure 26.10), which do not contain the slave node, and fails.

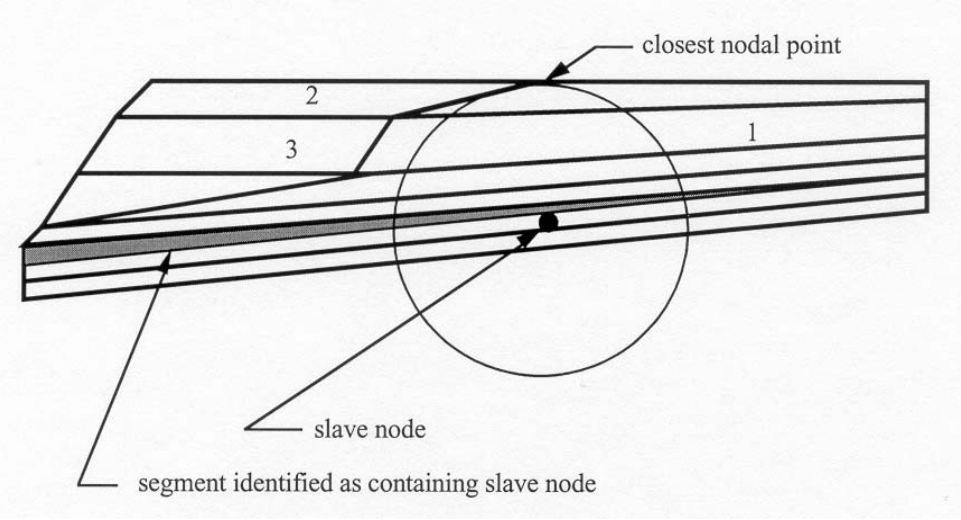


Figure 26.10. In case the stored segment fails to contain the node, the adjacent segments are checked.

26.8.2 Accounting For the Shell Thickness

Shell thickness effects are important when shell elements are used to model sheet metal. Unless thickness is considered in the contact, the effect of thinning on frictional interface stresses due to membrane stretching will be difficult to treat. In the treatment of thickness we project both the slave and master surfaces based on the mid-surface normal projection vectors as shown in Figure 26.11. The surfaces, therefore, must be offset by an amount equal to 1/2 their total thickness (Figure 26.12). This allows DYNA3D to check the node numbering of the segments automatically to ensure that the shells are properly oriented.

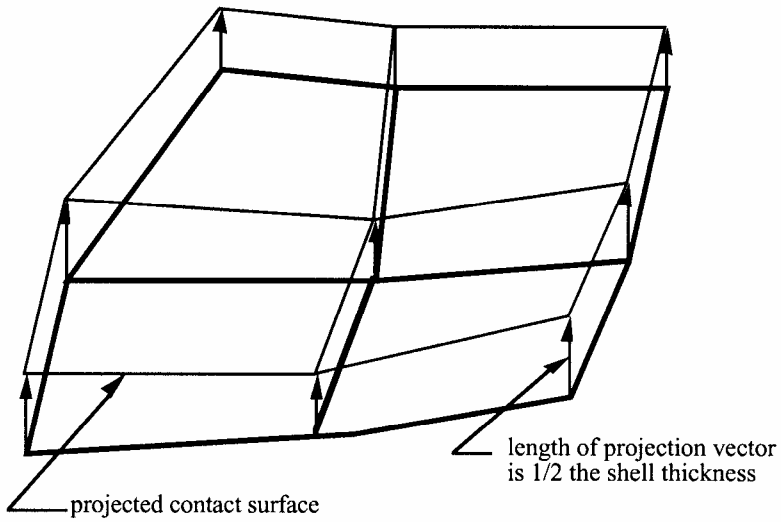


Figure 26.11. Contact surface is based on mid-surface normal projection vectors.

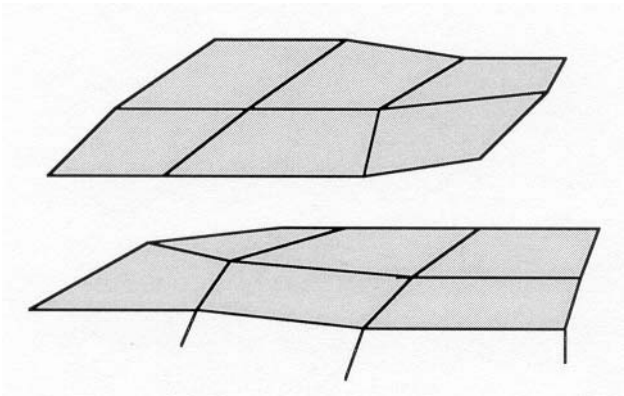


Figure 26.12. The slave and master surfaces must be offset in the input by one-half the total shell thickness. This also allows the segments to be oriented automatically.

Thickness changes in the contact are accounted for if and only if the shell thickness change option is flagged in the input. Each cycle, as the shell elements are processed, the nodal thicknesses are stored for use in the contact algorithms. The interface stiffness may change with thickness depending on the input options used.

Type 5 contact considers nodes interacting with a surface. This algorithm calls exactly the same subroutines as surface-to-surface but not symmetrically: i.e., the subroutines are called once, not twice. To account for the nodal thickness, the maximum shell thickness of any shell connected to the node is taken as the nodal thickness and is updated every cycle. The projection of the node is done normal to the contact surface as shown in Figure 26.13.

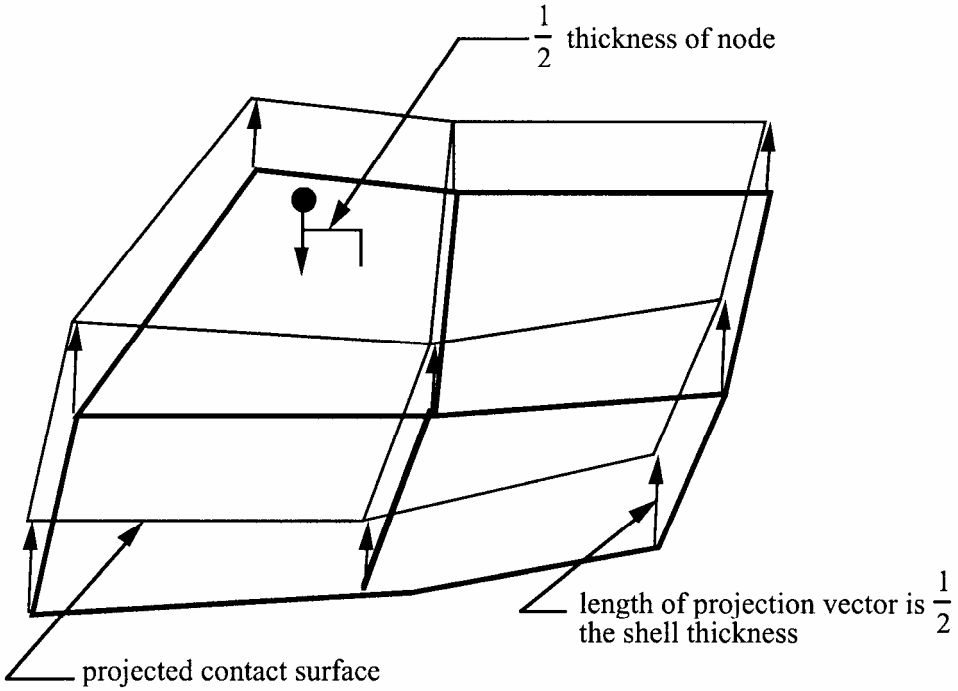


Figure 26.13. In a type 5 contact, thickness can also be taken into account.

26.8.3 Initial Contact Interpenetrations

The need to offset contact surfaces to account for the thickness of the shell elements contributes to initial contact interpenetrations. These interpenetrations can lead to severe numerical problems when execution begins so they should be corrected if LS-DYNA is to run successfully. Often an early growth of negative contact energy is one sign that initial interpenetrations exist. Currently, warning messages are printed to the terminal, the D3HSP file, and the MESSAG file to report interpenetrations of nodes through contact segments and the modifications to the geometry made by LS-DYNA to eliminate the interpenetrations. Sometimes such corrections simply move the problem elsewhere since it is very possible that the physical location of the shell mid-surface and possibly the shell thickness are incorrect. In the single surface contact algorithms any nodes still interpenetrating on the second time step are removed from the contact with a warning message.

In some geometry's interpenetrations cannot be detected since the contact node interpenetrates completely through the surface at the beginning of the calculation. This is illustrated in Figure 26.14. Another case contributing to initial interpenetrations occurs when the edge of a shell element is on the surface of a solid material as seen in Figure 26.15. Currently, shell edges are rounded with a radius equal to one-half the shell thickness.

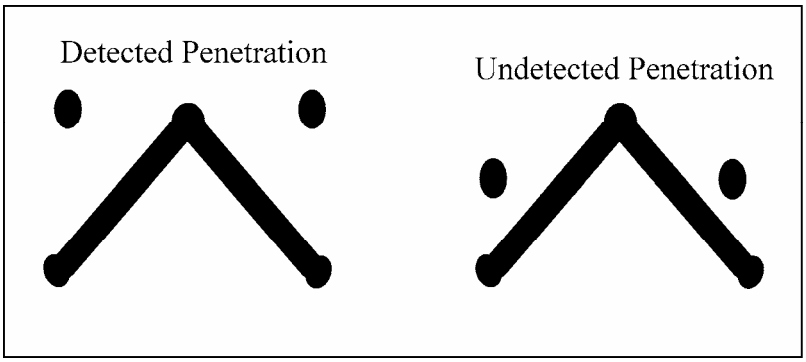


Figure 26.14. Undetected interpenetration. Such interpenetrations are frequently due to the use of coarse meshes.

To avoid problems with initial interpenetrations, the following recommendations should be considered:

- Adequately offset adjacent surfaces to account for part thickness during the mesh generation phase.
- Use consistently refined meshes on adjacent parts which have significant curvatures.
- Be very careful when defining thickness on shell and beam section definitions --especially for rigid bodies.
- Scale back part thickness if necessary. Scaling a 1.5mm thickness to .75mm should not cause problems but scaling to .075mm might. Alternatively, define a smaller contact thickness by part ID. Warning: if the part is too thin contact failure will probably occur
- Use spot welds instead of merged nodes to allow the shell mid surfaces to be offset.

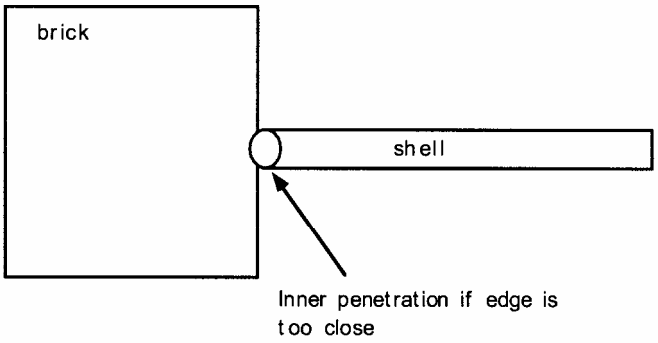


Figure 26.15. Undetected interpenetration due to rounding the edge of the shell element.

26.8.4 Contact Energy Calculation

Contact energy, $E_{contact}$, is incrementally updated from time n to time $n + 1$ for each contact interface as:

$$E_{contact}^{n+1} = E_{contact}^{n} + \left[\sum_{i=1}^{nsn} \Delta F_i^{slave} \times \Delta dist_i^{slave} + \sum_{i=1}^{nmn} \Delta F_i^{master} \times \Delta dist_i^{master} \right]^{n+\frac{1}{2}}$$

Where nsn is the number of slave nodes, nmn is the number of master nodes, ΔF_i^{slave} is the interface force between the ith slave node and the contact segment ΔF_i^{master} is the interface force between the ith master node and the contact segment, $\Delta dist_i^{slave}$ is the incremental distance the ith slave node has moved during the current time step, and $\Delta dist_i^{master}$ is the incremental distance the ith master node has moved during the current time step. In the absence of friction the slave and master side energies should be close in magnitude but opposite in sign. The sum, $E_{contact}$, should equal the stored energy. Large negative contact energy is usually caused by undetected penetrations. Contact energies are reported in the SLEOUT file. In the presence of friction and damping discussed below the interface energy can take on a substantial positive value especially if there is, in the case of friction, substantial sliding.

26.8.5 Contact Damping

Viscous contact damping has been added to all contact options including single surface contact. The original intent was to damp out oscillations normal to the contact surfaces during metal forming operations; however, it was later found to work effectively in removing high frequency noise in problems which involve impact. The input requires a damping value as a percentage of critical, $2m\omega$, where m is the mass and ω is the natural frequency. Letting k denote the interface stiffness, we compute the natural frequency for an interface slave node from Equation 26.15.

$$\omega = \sqrt{\frac{k\left(m_{slave} + m_{master}\right)}{m_{slave} m_{master}}} \quad m = \min\left\{m_{slave}, m_{master}\right\} \tag{26.18}$$

The master mass m_{master} is interpolated from the master nodes of the segment containing the slave node using the basis functions evaluated at the contact point of the slave node.

Force oscillations often occur as curved surfaces undergo relative motion. In these cases contact damping will eliminate the high frequency content in the contact reaction forces but will be unable to damp the lower frequency oscillations caused by nodes moving from segment to segment when there is a large angle change between the segments. This is shown in the hemispherical punch deep drawing in Figure 26.16. The reaction forces with and without contact damping in Figure 26.17 show only minor differences since the oscillations are not due to the dynamic effects of explicit integration. However, refining the mesh as shown in Figure 26.18 to include more elements around the die corner as in Figure 26.18 greatly reduces the oscillations as shown in Figure 26.19. This shows the importance of using an adequate mesh density in applications where significant relative motion is expected around sharp corners.

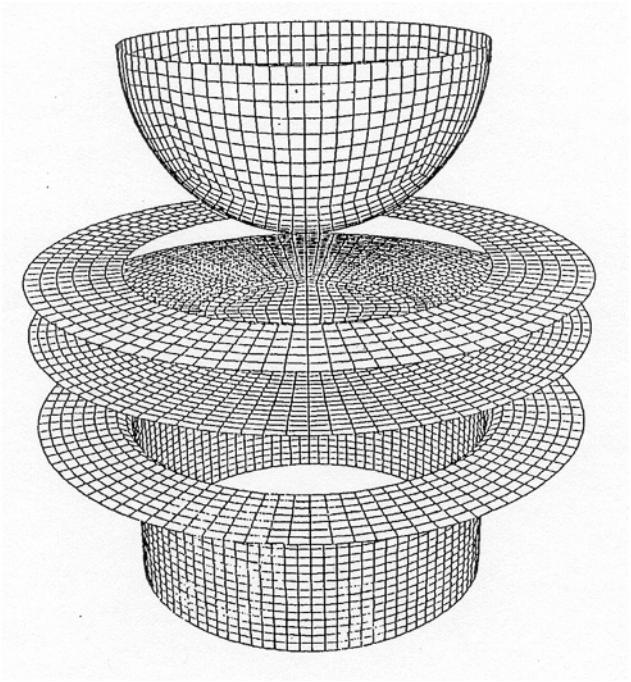


Figure 26.16. Hemispherical deep drawing problem.

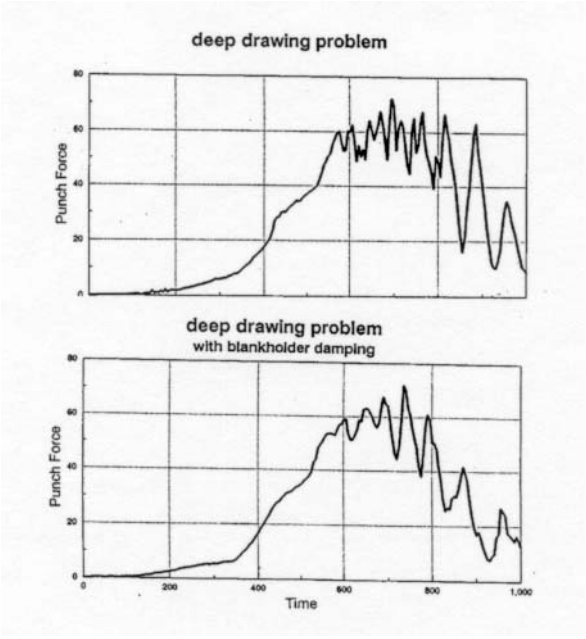


Figure 26.17. Reaction forces with and without contact damping.

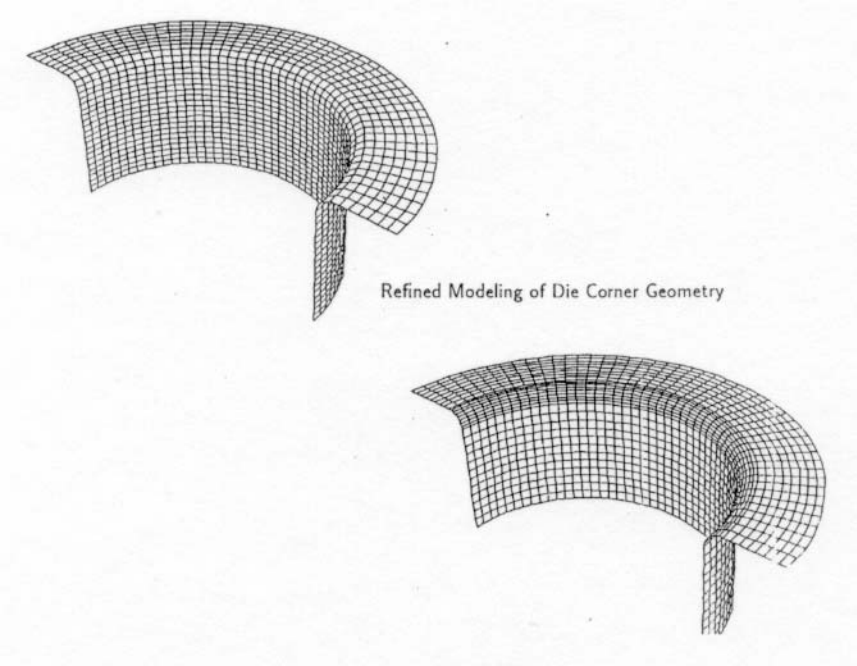


Figure 26.18. Refinement of die radius.

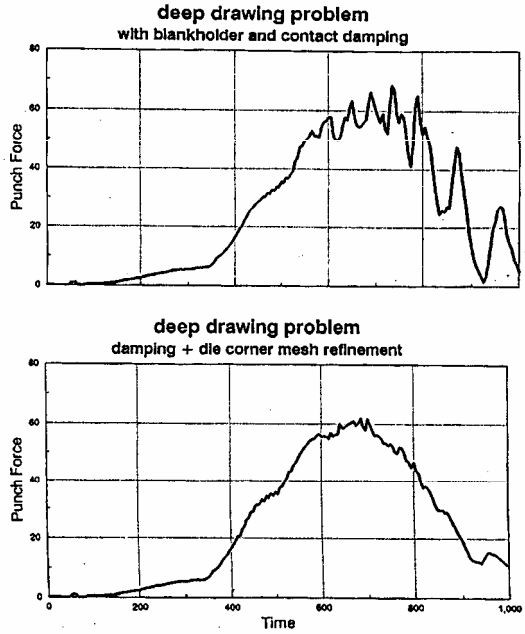


Figure 26.19. The oscillations are effectively eliminated by the mesh refinement.

26.8.6 Friction

Friction in LS-DYNA is based on a Coulomb formulation. Let f^* be the trial force, f_n the normal force, k the interface stiffness, μ the coefficient of friction, and f^n the frictional force at time n. The frictional algorithm, outlined below, uses the equivalent of an elastic plastic spring. The steps are as follows:

1. Compute the yield force, F_y:

$$F_y = \mu \left| f_n \right| \tag{26.19}$$

2. Compute the incremental movement of the slave node

$$\Delta e = \mathrm{r}^{n+1}\left(\xi_c^{n+1}, \eta_c^{n+1}\right) \quad - \mathrm{r}^{n+1}\left(\xi_c^{n}, \eta_c^{n}\right) \tag{26.20}$$

3. Update the interface force to a trial value:

$$f^* = f^n - k\Delta e \tag{26.21}$$

4. Check the yield condition:

$$f^{n+1} = f^* \qquad \text{if } \left| f^* \right| \le F_y \tag{26.22}$$

5. Scale the trial force if it is too large:

$$f^{n+1} = \frac{F_y f^*}{\left| f^* \right|} \quad \text{if } \left| f^* \right| > F_y \tag{26.23}$$

An exponential interpolation function smooths the transition between the static, μ_s, and dynamic, μ_d, coefficients of friction where v is the relative velocity between the slave node and the master segment:

$$\mu = \mu_d + \left(\mu_s - \mu_d\right) \ e^{-c|v|} \tag{26.24}$$

where

$$v = \frac{\Delta e}{\Delta t} \tag{26.25}$$

Δt is the time step size, and c is a decay constant.

The interface shear stress that develops as a result of Coulomb friction can be very large and in some cases may exceed the ability of the material to carry such a stress. We therefore allow another limit to be placed on the value of the tangential force:

$$f^{n+1} = \min\left(f_{Coulomb}^{n+1}, \kappa A_{master} \right) \tag{26.26}$$

where A_{master} is the area of the master segment and κ is the viscous coefficient. Since more than one node may contribute to the shear stress of a segment, we recognize that the stress may still in some cases exceed the limit κ.

Typical values of friction, see Table 26.1, can be found in Marks Engineering Handbook.

MATERIALS	STATIC	SLIDING
Hard steel on hard steel	0.78 (dry)	.08 (greasy), .42 (dry)
Mild steel on mild steel	0.74 (dry)	.10 (greasy), .57 (dry)
Aluminum on mild steel	0.61 (dry)	.47 (dry)
Aluminum on aluminum	1.05 (dry)	1.4 (dry)
Tires on pavement (40psi)	0.90 (dry)	.69(wet), .85(dry)

Table 26.1. Typical values of Coulomb Friction [Marks]

26.9 Tied Interfaces

Sudden transitions in zoning are permitted with the tied interfaces as shown in Figure 26.20 where two meshes of solid elements are joined. This feature can often decrease the amount of effort required to generate meshes since it reduces the need to match nodes across interfaces of merged parts.

Tied interfaces include four interface options of which three are in the Sliding Interface Definition Section in the LS-DYNA User's Manual. These are:

- Type 2 for tying surfaces with translational degrees of freedom.
- Type 6 for tying translational degrees of freedom of nodes to a surface
- Type 7 for tying both translational and rotational degrees of freedom of nodes

The fourth option is in the "Tie-Breaking Shell Definitions" Section of the user's manual and is meant as a way of tying edges of adjacent shells together. Unlike Type 7 this latter option does not require a surface definition, simply nodal lines, and includes a failure model based on plastic strain which can be turned off by setting the plastic failure strain to a high value. The first two options, which are equivalent in function but differ in the input definition, can be properly applied to nodes of elements which lack rotational degrees of freedom. The latter options must be used with element types that have rotational degrees of freedom defined at their nodes such as the shell and beam elements. One important application of Type 7 is that it allows edges of shells to be tied to shell surfaces. In such transitions the shell thickness is not considered.

Since the constraints are imposed only on the slave nodes, the more coarsely meshed side of the interface is recommended as the master surface. Ideally, each master node should coincide with a slave node to ensure complete displacement compatibility along the interface, but in practice this is often difficult if not impossible to achieve. In other words, master nodes that do not coincide with a slave node can interpenetrate through the slave surface.

Implementation of tied interface constraints is straightforward. Each time step we loop through the tied interfaces and update each one independently. First, we distribute the nodal forces and nodal mass of each slave node to the master nodes which define the segment containing the contact point, i.e., the increments in mass and forces

$$\Delta f_m^i = \phi_i(\xi_c, \eta_c) f_s \tag{26.27}$$

are added to the mass and force vector of the master surface. After the summation over all slave nodes is complete, we can compute the acceleration of the master surface. The acceleration of each slave node a_{i_s} is then interpolated from the master segment containing its contact points:

$$a_{i_s} = \sum_{j=1}^{4} \phi_j(\xi_c, \eta_c) a_i^j \tag{26.28}$$

Velocities and displacements are now updated normally.

The interpolated contact point, (ξ_c, η_c), for each slave node is computed once, since its relative position on the master segment is constant for the duration of the calculation. If the closest point projection of the slave node to the master surface is non-orthogonal, values of (ξ_c, η_c) greater than unity will be computed. To allow for slight errors in the mesh definition, the slave node is left unconstrained if the magnitude of the contact point exceeds 1.02. Great care should be exercised in setting up tied interfaces to ensure that the slave nodes are covered by master segments.

Conflicting constraints must be avoided. Care should be taken not to include nodes that are involved in a tied interfaces in another tied interface, in constraint sets such as nodal constraint sets, in linear constraint equations, and in spot welds. Furthermore, tied interfaces between rigid and deformable bodies are not permitted. LS-DYNA checks for conflicting constraints on nodal points and if such conflicts are found, the calculation will terminate with an error message identifying the conflict. Nodes in tied interfaces should not be included as slave nodes in rigid wall definitions since interactions with stonewalls will cause the constraints that were applied in the tied interface logic to be violated. We do not currently check for this latter condition is LS-DYNA.

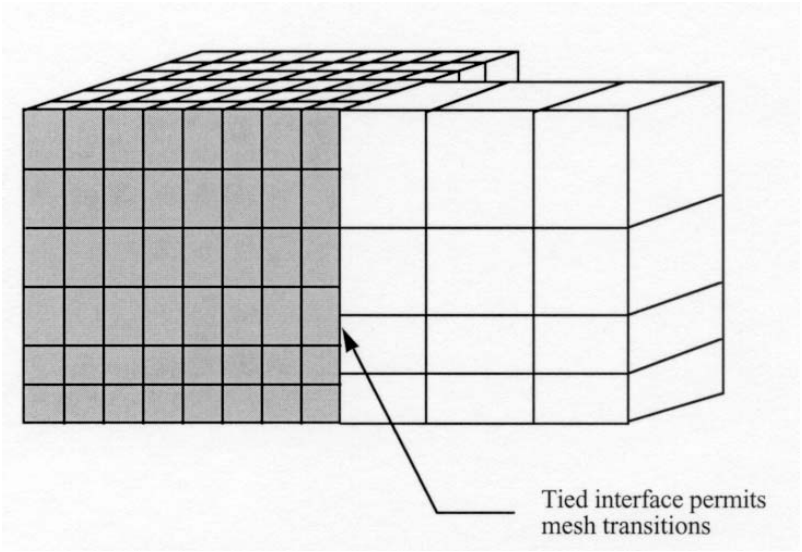


Figure 26.20. Tied interface used for a mesh transition.

Tied interfaces require coincident surfaces and for shell element this means that the mid-surfaces must be coincident. Consider Figure 26.21 where identical slave and master surfaces are

offset. In this case the tied constraints require that translational velocities of tied nodes be identical, i.e.,

$$v^s = v^m$$

Consequently, if the nodes are offset, rotations are not possible. The velocity of a tied slave node in Figure 26.21 should account for the segment rotation:

$$v^s = v^m - \hat{z}\, e_3 \times \omega$$

where $\hat{z}$ is the distance to the slave node, e_3 is the normal vector to the master surface at the contact point, and ω is the angular velocity. Since this is not the case in the tied interfaces logic, $\hat{z}$ must be of zero length.

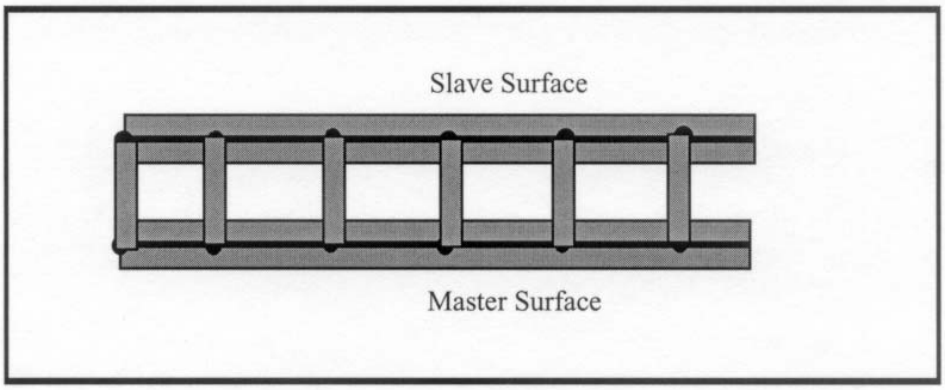


Figure 26.21. Offset tied interface.

LS-DYNA projects tied slave nodes back to the master surface if possible and prints warning messages for all projected offset nodes or nodes to far away to tie. This projection eliminates the problems with rotational constraints but creates other difficulties:

- Geometry is modified
- Tied interfaces must be excluded from automatic generation since tied surfaces cannot be mixed with automatic contact with thickness offsets.

An offset capability has been added to the tied interfaces which uses a penalty approach. The penalty approach removes the major limitations of the constraint formulation since with the offset option:

- Multiple tied interfaces cannot share common nodes.
- Rigid body nodes can be constrained.
- Tied interface nodes can have other constraints applied and can be subjected to prescribed motions.

26.10 Sliding-Only Interfaces

This option is seldom useful in structural calculations. Its chief usefulness is for treating interfaces where the gaseous detonation products of a high explosive act on a solid material. The present algorithm, though simple, has performed satisfactorily on a number of problems of this latter type. We briefly outline the approach here since the algorithm is still experimental and subject to change.

The method consists of five steps. In the first step, the mass per unit area (mass/area) and pressure are found at each node on the slave surface. Next, the contact point for each master node is found, and the slave mass/area and slave pressure at each master node is interpolated from the slave surface. In the third step, this pressure distribution is applied to the master surface to update its acceleration. In the fourth step, the normal component of the acceleration at each node on the master surface is scaled by its z-factor defined as the mass/area of the master surface at the master node divided by the sum of the mass/area of the slave surface at the master node. The last step consists of resetting the normal acceleration and velocity components of all slave nodes to ensure compatibility.

26.11 Bucket Sorting

Bucket sorting is now used extensively in both the surface to surface and single surface contact algorithms. Version 920 of LS-DYNA no longer contains one-dimensional sorting. Presently two separate but similar bucket sorts are in LS-DYNA. In the first and older method we attempt to find for each node the three nearest nodes. In the newer method which is systematically replacing the older method we locate the nearest segment.

The reasons for eliminating slave node tracking by incremental searching is illustrated in Figure 26.22 where surfaces are shown which cause the incremental searches to fail. In LS-DYNA tied interfaces are used extensively in many models creating what appears to the contact algorithms to be topologically disjoint regions. For robustness, our new algorithms account for such mesh transitions with only minor cost penalties. With bucket sorting incremental searches may still be used but for reliability they are used after contact is achieved. As contact is lost, the bucket sorting for the affected nodal points must resume.

In a direct search of a set of N nodes to determine the nearest node, the number of distance comparisons required is $N-1$. Since this comparison needs to be made for each node, the total number of comparisons is $N\,(N-1)$, with each of these comparisons requiring a distance calculation

$$1^2 = (x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 \qquad (26.29)$$

that uses eight mathematical operations. The cumulative effect of these mathematical operations for $N\,(N-1)$ compares can dominate the solution cost at less than 100 elements.

The idea behind a bucket sort is to perform some grouping of the nodes so that the sort operation need only calculate the distance of the nodes in the nearest groups. Consider the partitioning of the one-dimensional domain shown in Figure 26.23. With this partitioning the nearest node will either reside in the same bucket or in one of the two adjoining buckets. The number of distance calculations is now given by

$$\frac{3N}{a} - 1 \qquad (26.30)$$

where a is the number of buckets. The total number of distance comparisons for the entire one-dimensional surface is

$$N\left(\frac{3N}{a}-1\right) \tag{26.31}$$

Thus, if the number of buckets is greater than 3, then the bucket sort will require fewer distance comparisons than a direct sort. It is easy to show that the corresponding number of distance comparisons for two-dimensional and three-dimensional bucket sorts are given by

$$N\left(\frac{9N}{ab}-1\right) \quad \text{for 2D} \tag{26.32}$$

$$N\left(\frac{27N}{abc}-1\right) \quad \text{for 3D} \tag{26.33}$$

where b and c are the number of partitions along the additional dimension.

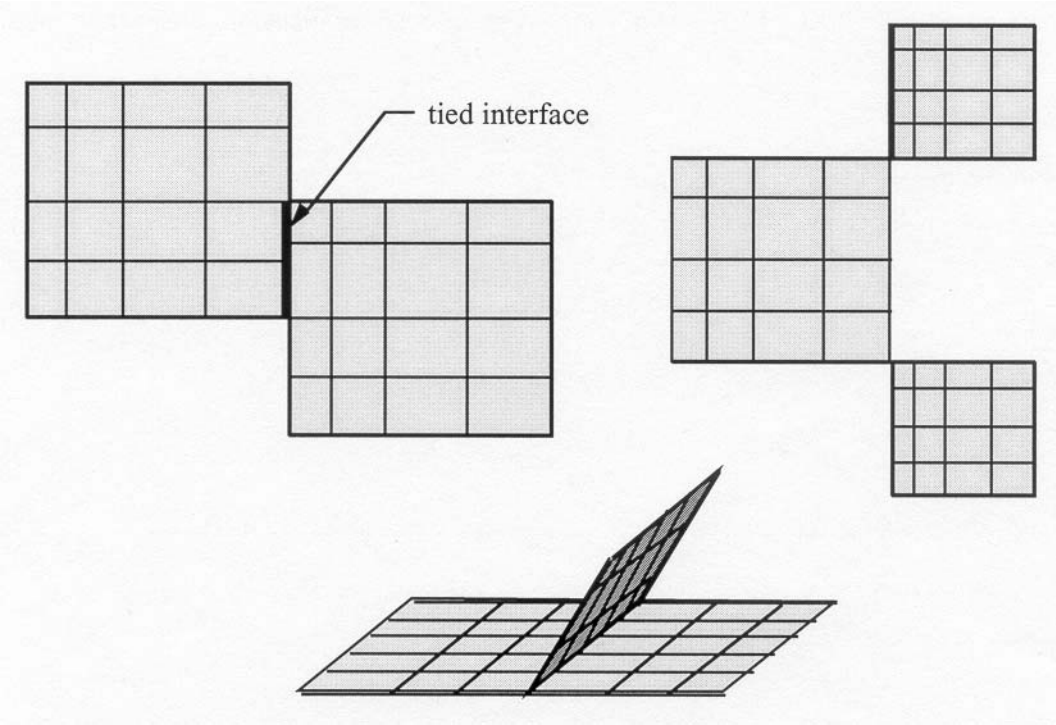


Figure 26.22. Incremental searching may fail on surfaces that are not simply connected. The new contact algorithm in LS-DYNA avoids incremental searching for nodal points that are not in contact and all these cases are considered.

The cost of the grouping operations, needed to form the buckets, is nearly linear with the number of nodes N. For typical LS-DYNA applications, the bucket sort is 100 to 1000 times

faster than the corresponding direct sort. However, the sort is still an expensive part of the contact algorithm, so that, to further minimize this cost, the sort is performed every ten or fifteen cycles and the nearest three nodes are stored. Typically, three to five percent of the calculational costs will be absorbed in the bucket sorting when most surface segments are included in the contact definition.

26.11.1 Bucket Sorting in TYPE 4 Single Surface Contact

We set the number of buckets in the x, y, and z coordinate directions to NX, NY, and NZ, respectively. Letting $LMAX$ represent the longest characteristic length (found by checking the length of the segment diagonals and taking a fraction thereof) over all segments in the contact definition, the number of buckets in each direction is given by

$$NX = \frac{x_{\max} - x_{\min}}{LMAX} \tag{26.34}$$

$$NY = \frac{y_{\max} - y_{\min}}{LMAX} \tag{26.35}$$

$$NZ = \frac{z_{\max} - z_{\min}}{LMAX} \tag{26.36}$$

where the coordinate pairs $(x_{\min}, x_{\max})$, $(y_{\min}, y_{\max})$, and $(z_{\min}, z_{\max})$ define the extent of the contact surface and are updated each time the bucket searching is performed. In order to dynamically allocate memory effectively with FORTRAN, we further restrict the number of buckets such that the total number of buckets does not exceed the number of nodes in the contact surface, NSN or 5000:

$$NX \cdot NY \cdot NZ \le MIN\ (NSN, 5000) \tag{26.37}$$

If the characteristic length, $LMAX$, is large due to an oversized contact segment or an instability leading to a node flying off into space, the bucket sorting can be slowed down considerably since the number of buckets will be reduced. In older versions of DYNA3D this led to the error termination message “More than 1000 nodes in bucket.”

The formulas given by Belytschko and Lin [1985] are used to find the bucket containing a node with coordinates (x, y, z). The bucket pointers are given by

$$PX = NX \cdot \frac{\left(x - x_{\min}\right)}{\left(x_{\max} - x_{\min}\right)} + 1 \tag{26.38}$$

$$PY = NY \cdot \frac{\left(y - y_{\min}\right)}{\left(y_{\max} - y_{\min}\right)} + 1 \tag{26.39}$$

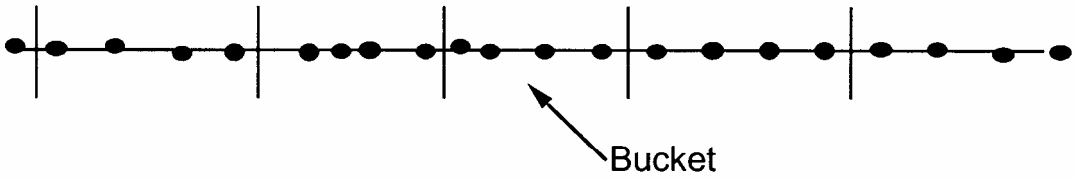
$$PZ = NZ \cdot \frac{\left(z - z_{\min}\right)}{\left(z_{\max} - z_{\min}\right)} + 1 \tag{26.40}$$

and are used to compute the bucket number given by

$$NB = PX + (PY - 1) \cdot PX + (PZ - 1) \cdot PX \cdot PY \qquad (26.41)$$

For each nodal point, k, in the contact surface we locate the three nearest neighboring nodes by searching all nodes in buckets from

$$\begin{aligned} &\text{MAX}(1,PX\ \ 1),\ \text{MIN}(NX,PX + 1) \\ &\text{MAX}(1,PY\ \ 1),\ \text{MIN}(NY,PY + 1) \\ &\text{MAX}(1,PZ\ \ 1),\ \text{MIN}(NZ,PZ + 1) \end{aligned} \qquad (26.42)$$



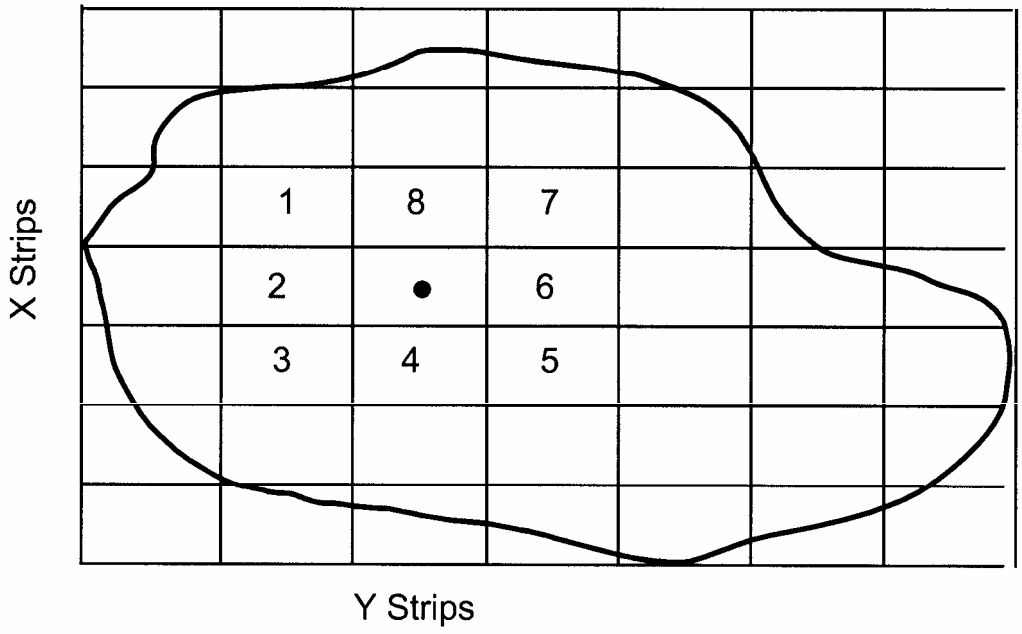


Figure 26.23. One- and two-dimensional bucket sorting.

A maximum of twenty-seven buckets are searched. Nodes that share a contact segment with k are not considered in this nodal search. By storing the three nearest nodes and rechecking these stored nodes every cycle to see if the nearest node has changed, we avoid performing the bucket sorting every cycle. Typically, sorting every five to fifteen cycles is adequate. Implicit in this approach is the assumption that a node will contact just one surface. For this reason the single surface contact (TYPE 4 in LS-DYNA) is not applicable to all problems. For example, in metal forming applications both surfaces of the workpiece are often in contact.

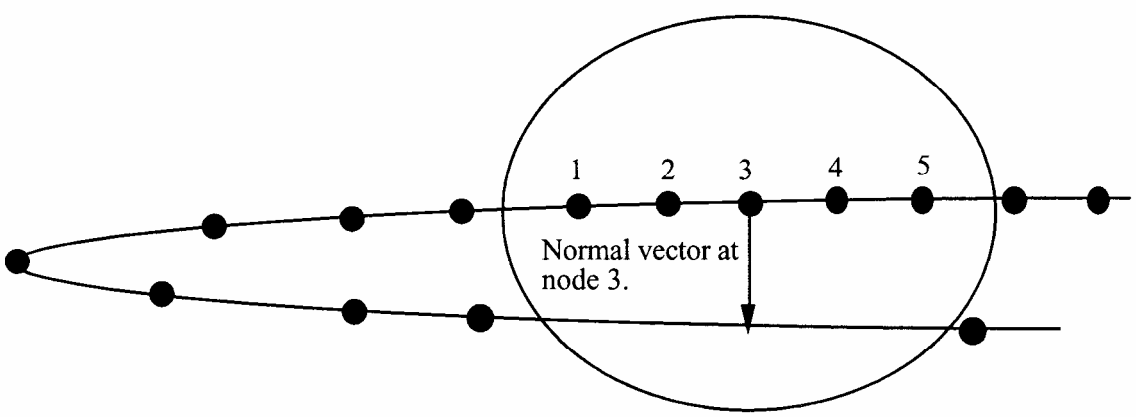


Figure 26.24. Nodes 2 and 4 share segments with node 3 and therefore the two nearest nodes are1 and 5. The nearest contact segment is not considered since its nodes are not members of the nearest node set.

The nearest contact segment to a given node, k, is defined to be the first segment encountered when moving in a direction normal to the surface away from k. A major deficiency with the nearest node search is depicted in Figure 26.24 where the nearest nodes are not even members of the nearest contact segment. Obviously, this would not be a problem for a more uniform mesh. To overcome this problem we have adopted segment based searching in both surface to surface and single surface contact.

26.11.2 Bucket Sorting in Surface to Surface and TYPE 13 Single Surface Contact

The procedure is roughly the same as before except we no longer base the bucket size on $LMAX$ which can result in as few as one bucket being generated. Rather, the product of the number of buckets in each direction always approaches NSN or 5000 whichever is smaller,

$$NX \cdot NY \cdot NZ \le MIN(NSN, 5000) \tag{26.43}$$

where the coordinate pairs $(x_{\min},\ x_{\max})$, $(y_{\min},\ y_{\max})$, and $(z_{\min},\ z_{\max})$ span the entire contact surface. In the new procedure we loop over the segments rather than the nodal points. For each segment we use a nested DO LOOP to loop through a subset of buckets from $IMIN$ to $IMAX$, $JMIN$ to $JMAX$, and to $KMAX$ where

$$IMIN = MIN(PX1, PX2, PX3, PX4)$$
$$IMAX = MAX(PX1, PX2, PX3, PX4)$$

$$
\begin{aligned}
JMIN &= MIN(PY1, PY2, PY3, PY4) \\
JMAX &= MAX\ (PY1, PY2, PY3, PY4)
\end{aligned}
\qquad (26.44)
$$

$$
\begin{aligned}
KMIN &= MIN(PZ1, PZ2, PZ3, PZ4) \\
KMAX &= MAX\ (PZ1, PZ2, PZ3, PZ4)
\end{aligned}
$$

and PXk, PYk, PZk are the bucket pointers for the kth node. Figure 26.25 shows a segment passing through a volume that has been partitioned into buckets.

We check the orthogonal distance of all nodes in the bucket subset from the segment. As each segment is processed, the minimum distance to a segment is determined for every node in the surface and the two nearest segments are stored. Therefore the required storage allocation is still deterministic. This would not be the case if we stored for each segment a list of nodes that could possibly contact the segment.

We have now determined for each node, k, in the contact surface the two nearest segments for contact. Having located these segments we permanently store the node on these segments which is nearest to node k. When checking for interpenetrating nodes we check the segments surrounding the node including the nearest segment since during the steps between bucket searches it is likely that the nearest segment may change. It is possible to bypass nodes that are already in contact and save some computer time; however, if multiple contacts per node are admissible then bypassing the search may lead to unacceptable errors.

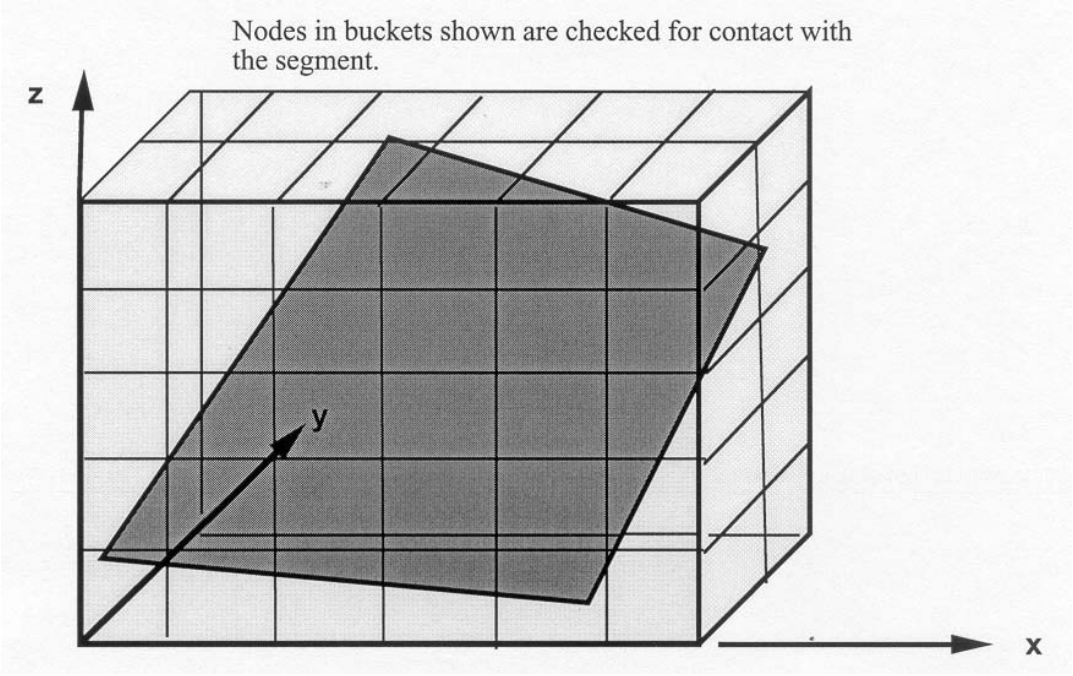


Figure 26.25. The orthogonal distance of each slave node contained in the box from the segment is determined. The box is subdivided into sixty buckets.

26.12 Single Surface Contact Algorithms in LS-DYNA

The single surface contact algorithms evolved from the surface to surface contact algorithms and the post contact searching follows the procedures employed for the surface to surface contact. Type 4 contact in LS-DYNA uses the following steps where *NSEG* is the number of contact segments and *NSN* is the number of nodes in the interface:

I. Loop through the contact segments from 1 to NSEG

1. Compute the normal segment vectors and accumulate an area weighted average at the nodal points to determine the normal vectors at the nodal points.

II. Loop through the slave nodes from 1 to NSN

1. Check all nearest nodes, stored from the bucket sort, and locate the node which is nearest.

2. Check to see if nearest node is within a penetration tolerance determined during the bucket sort, if not, proceed to the end of the loop.

3. For shell elements, determine if the nearest node is approaching the segment from the positive or negative side based on the right hand rule. Project both the node and the contact segment along the nodal normal vectors to account for the shell thickness.

4. Check for interpenetrating nodes and if a node has penetrated apply a nodal point force that is proportional to the penetration depth.

End of Loop

Of course, several obvious limitations of the above procedure exists. The normal vectors that are used to project the contact surface are meaningless for nodes along an intersection of two or more shell surfaces (Please see the sketch at the bottom of Figure 26.26). In this case the normal vector will be arbitrarily skewed depending on the choice of the numbering of the connectivities of the shells in the intersecting surfaces. Secondly, by considering the possibility of just one contact segment per node, metal forming problems cannot be handled within one contact definition. For example, if a workpiece is constrained between a die and a blankholder then at least some nodal points in the workpiece must necessarily be in contact with two segments-one in the die and the other in the workpiece. These two important limitations have motivated the development of the new bucket sorting procedure described above and the modified single surface contact procedure, type 13.

A major change in type 13 contact from type 4 is the elimination of the normal nodal vector projection by using the segment normal vector as shown in Figure 26.26.

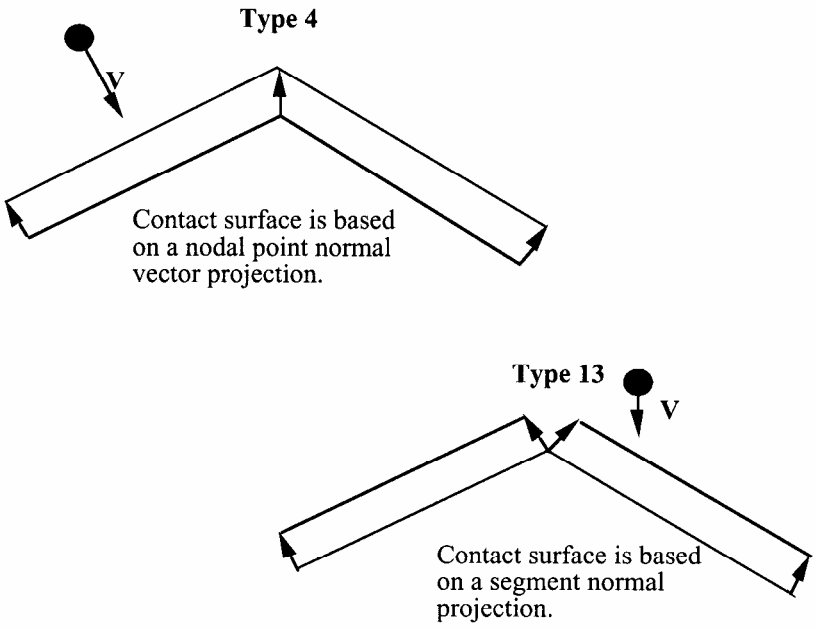


Figure 26.26. Projection of the contact surface for a node approaching from above is shown for types 4 and 13 contact.

Segment numbering within the contact surface is arbitrary when the segment normal is used greatly simplifying the model input generation. However, additional complexity is introduced since special handling of the nodal points is required at segment intersections where nodes may approach undetected as depicted in Figure 26.27a.

To overcome this limitation an additional logic that put cylindrical cap at segment intersections has been introduced in contact type 13 (and a3). See Figure 26.27b.

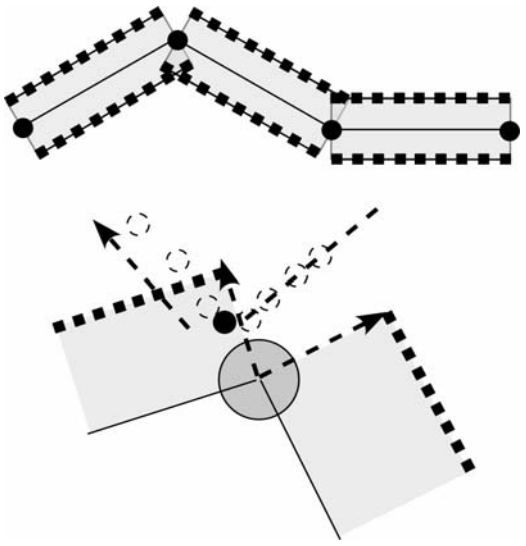


Figure 26.27a.

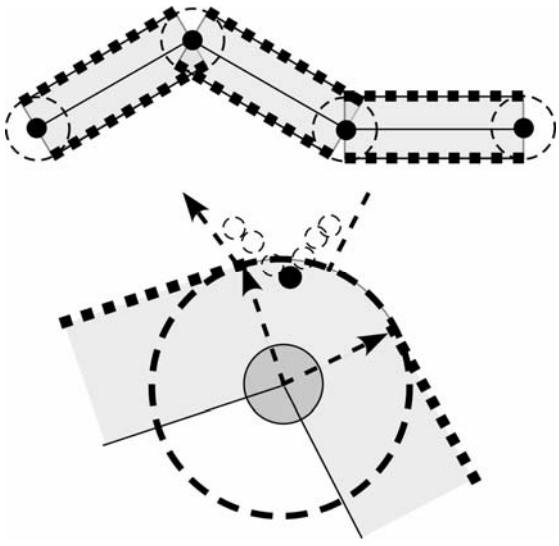


Figure 26.27b.

Assuming the segment based bucket sort has been completed and closest segments are known for all slave nodes then the procedure for processing the type 13 contact simplifies to:

I. Loop through the slave nodes from 1 to NSN

1. If node is in contact, check to see if the contact segment has changed and if so, then update the closest segment information and the orientation flag which remembers the side in contact. Since no segment orientation information is stored this flag may change as the node moves from segment to segment.

2. Check the closest segment to see if the node is in contact if not then proceed to the end of the loop. If the slave node or contact segment connectivity is a member of a shell element, project both the node and the contact segment along the segment normal vector to account for the shell thickness. A nodal thickness is stored for each node and a segment thickness is stored for each segment. A zero thickness is stored for solid elements. The thickness can be optionally updated to account for membrane thinning.

3. Check for interpenetrating nodes and if a node has penetrated apply a nodal point force that is proportional to the penetration depth.

End of Loop

Note that type 13 contact does not require the calculation of nodal normal vectors.

26.13 Surface to Surface Constraint Algorithm

The constraint algorithm that we implemented is based on the algorithm developed by Taylor and Flanagan [1989]. This involves a two-pass symmetric approach with a partitioning parameter, β, that is set between negative and positive unity where $\beta=1$ and $\beta=-1$ correspond to one way treatments with the master surface accumulating the mass and forces from the slave surface (for $\beta=1$) and visa versa (for $\beta=-1$). The searching algorithms are those used in the other contact algorithms for the surface to surface contact.

In this constraint approach the accelerations, velocities, and displacements are first updated to a trial configuration without accounting for interface interactions. After the update, a penetration force is computed for the slave node as a function of the penetration distance ΔL:

$$f_p = \frac{m_s \Delta L}{\Delta t^2} n \qquad (26.45)$$

where n is the normal vector to the master surface.

We desire that the response of the normal component of the slave node acceleration vector, a_s, of a slave node residing on master segment k be consistent with the motion of the master segment at its contact segment (s_c,t_c), i.e.,

$$a_s = \phi_1(s_c,t_c)a^1_{nk} + \phi_2(s_c,t_c)a^2_{nk} + \phi_3(s_c,t_c)a^3_{nk} + \phi_4(s_c,t_c)a^4_{nk} \qquad (26.46)$$

For each slave node in contact with and penetrating through the master surface in its trial configuration, its nodal mass and its penetration force given by Equation (26.43) is accumulated to a global master surface mass and force vector:

$$\left(m_k + \sum_s m_{ks} \right) a_{nk} = \sum_s f_{ks} \tag{26.47}$$

where

$$m_{ks} = \phi_k m_s$$

$$f_{ks} = \phi_k f_p \tag{26.48}$$

After solving Equation (26.45) for the acceleration vector, a_{nk}, we can obtain the acceleration correction for the slave node as

$$a_{ns} = a_s - \frac{f_p}{m_s} \tag{26.49}$$

The above process is repeated after reversing the master and slave definitions. In the final step the averaged final correction to the acceleration vector is found

$$a_n^{final} = \tfrac{1}{2}(1-\beta)\, a_n^{1st\ pass} + \tfrac{1}{2}(1+\beta)\, a_n^{2nd\ pass} \tag{26.50}$$

and used to compute the final acceleration at time $n+1$

$$a^{n+1} = a^{trial} + a_n^{final} \tag{26.51}$$

Friction, as described by Taylor and Flanagan [1989], is included in our implementation. Friction resists the relative tangential velocity of the slave node with respect to the master surface. This relative velocity if found by subtracting from the relative velocity:

$$v_r = v_s - \left(\phi_1 v_k^1 + \phi_2 v_k^2 + \phi_3 v_k^3 + \phi_4 v_k^4 \right) \tag{26.52}$$

the velocity component normal to the master segment:

$$v_t = v_r - (n \cdot v_r) n \tag{26.53}$$

A trial tangential force is computed that will cancel the tangential velocity

$$f^* = \frac{m_s v_t}{\Delta t} \tag{26.54}$$

where v_t is the magnitude of the tangential velocity vector

$$v_t = \sqrt{v_t \cdot v_t} \tag{26.55}$$

The magnitude of the tangential force is limited by the magnitude of the product of the Coulomb friction constant with the normal force defined as

$$f_n = m_s a_{ns} \cdot n \tag{26.56}$$

The limiting force is, therefore,

$$F_y = m|f_n| \tag{26.19}$$

and

$$f^{n+1} = f^* \qquad \text{if } |f^*| = F_y \tag{26.22}$$

$$f^{n+1} = \frac{F_y f^*}{|f^*|} \qquad \text{if } |f^*| > F_y \tag{26.23}$$

Therefore, using the above equations the modification to the tangential acceleration component of the slave node is given by

$$a_t = \min\left(\mu a_{nt} \cdot n, \frac{|v_s|}{\Delta t} \right) \tag{26.57}$$

which must act in the direction of the tangential vector defined as

$$n_t = \frac{v_t}{v_t} \tag{26.58}$$

The corrections to both the slave and master node acceleration components are:

$$a_{ts} = a_t n_t$$

$$a_{tk} = -\phi_k \frac{a_s m_s}{m_k} n_t \tag{26.59}$$

The above process is again repeated after reversing the master and slave definitions. In the final step the averaged final correction to the acceleration vector is found

$$a_t^{final} = \tfrac{1}{2}(1-\beta) a_t^{1st\ pass} + \tfrac{1}{2}(1+\beta) a_t^{2nd\ pass} \tag{26.60}$$

and is used to compute the final acceleration at time $n+1$

$$a^{n+1} = a^{trial} + a_n^{final} + a_t^{final} \tag{26.61}$$

A significant disadvantage of the constraint method relative to the penalty method appears if an interface node is subjected to additional constraints such as spot welds, constraint equations, tied interfaces, and rigid bodies. Rigid bodies can often be used with this contact algorithm if their motions are prescribed as is the case in metal forming. For the more general cases involving rigid bodies, the above equations are not directly applicable since the local nodal masses of rigid body nodes are usually meaningless. Subjecting the two sides of a shell surface to this constraint algorithm will also lead to erroneous results since an interface node cannot be constrained to move simultaneously on two mutually independent surfaces. In the latter case the constraint technique could be used on one side and the penalty method on the other.

The biggest advantage of the constraint algorithm is that interface nodes remain on or very close to the surfaces they are in contact with. Furthermore, elastic vibrations that can occur in penalty formulations are insignificant with the constraint technique. The problem related to finding good penalty constants for the contact are totally avoided by the latter approach. Having both methods available is possibly the best option of all.

26.14 Planar Rigid Boundaries

The rigid boundary represents the simplest contact problem and is therefore treated separately. As shown in Figure 26.28 the boundary is flat, finite or infinite in extent and is defined by an outward normal unit vector **n** with the origin of **n** at a corner point on the wall if the wall is finite or at an arbitrary point on the wall if the wall extends to infinity. The finite wall is rectangular with edges of length **L** and **M**. Unit vectors **l** and **m** lie along these edges. A subset of nodes is defined, usually boundary nodes of the calculational model, that are not allowed to penetrate. Let *k* represent one such boundary node and let r_k^{n+1} be the position vector from the origin of **n** to *k* after locally updating the coordinates. Each time step prior to globally updating the velocities and accelerations we check k to ensure that the nodes lies within the wall by checking that both inequalities are satisfied:

$$\begin{aligned} r_k^{n+1} \cdot l &\le L \\ r_k^{n+1} \cdot m &\le M \end{aligned} \tag{26.62}$$

This test is skipped for the infinite rigid wall. Assuming that the inequality is satisfied, we then check the penetration condition to see if k is penetrating through the wall,

$$r_k^{n+1} \cdot n < 0 \tag{26.63}$$

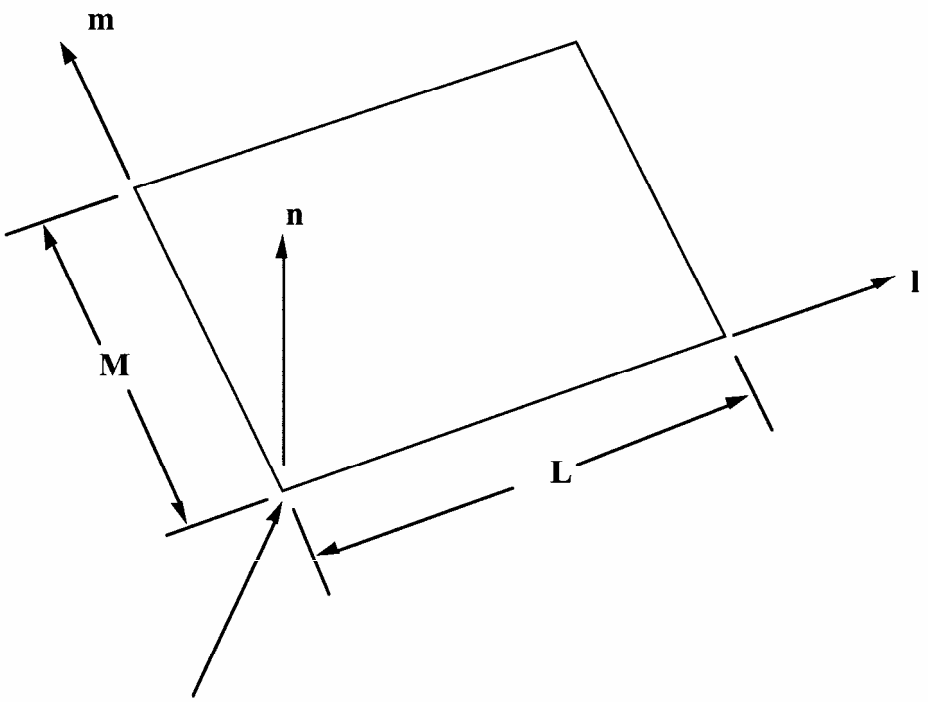


Figure 26.28. Vector **n** is normal to the stonewall. An optional vector **l** can be defined such that $m = n \times 1$. The extent of the stonewall is limited by defining **L** and **M**. A zero value for either of these lengths indicates that the stonewall is infinite in that direction.

and if so, the velocity and acceleration components normal to the wall are set to zero:

$$a^n_{k_{new}} = a^n_{k_{old}} - \left(a^n_{k_{old}} \cdot n\right) n$$

$$v^n_{k_{new}} = v^n_{k_{old}} - \left(v^n_{k_{old}} \cdot n\right) n \tag{26.64}$$

Here a_k and v_k are the nodal acceleration and velocity of node k, respectively. This procedure for stopping nodes represents a perfectly plastic impact resulting in an irreversible energy loss. The total energy dissipated is found by taking the difference between the total kinetic energy of all the nodal points slaved to the rigid wall before and after impact with the wall. This energy is computed and accumulated in LS-DYNA and is printed in the GLSTAT (global statistics) file.

The tangential motion of the boundary node may be unconstrained, fully constrained, or subjected to Coulomb friction while it is in contact with the rigid boundary.
Coulomb friction acts along a vector defined as:

$$n_t = \frac{v^n_{k_{new}}}{\sqrt{v^n_{k_{new}} \cdot v^n_{k_{new}}}} \tag{26.65}$$

The magnitude of the tangential force which is applied to oppose the motion is given as

$$f_t = \min\left(\frac{m_s \sqrt{v^n_{k_{new}} \cdot v^n_{k_{new}}}}{\Delta t}, \mu \left| f_n \right| \right) \tag{26.66}$$

i.e., the maximum value required to hold the node in the same relative position on the stonewall or the product of the coefficient of friction and the magnitude of the normal force which ever is less. In Equation (26.66), m_s is the mass of the slave node and f_n is the normal force.

26.15 Geometric Rigid Boundaries

The numerical treatment of geometric rigid walls is somewhat similar to that for the finite planar rigid walls. The geometric rigid walls can be subjected to a prescribed translational motion along an arbitrarily oriented vector; however, rotational motion is not permitted. As the geometric surface moves and contacts the structure, external work is generated which is integrated and added to the overall energy balance. In addition to the external work, plastic work also is generated as nodes contact the wall and assume the walls normal velocity at the point of contact. Contact can occur with any of the surfaces which enclose the volume. Currently four geometric shapes are available including the rectangular prism, the cylinder, flat surface, and sphere. These are shown in Figure 26.29.

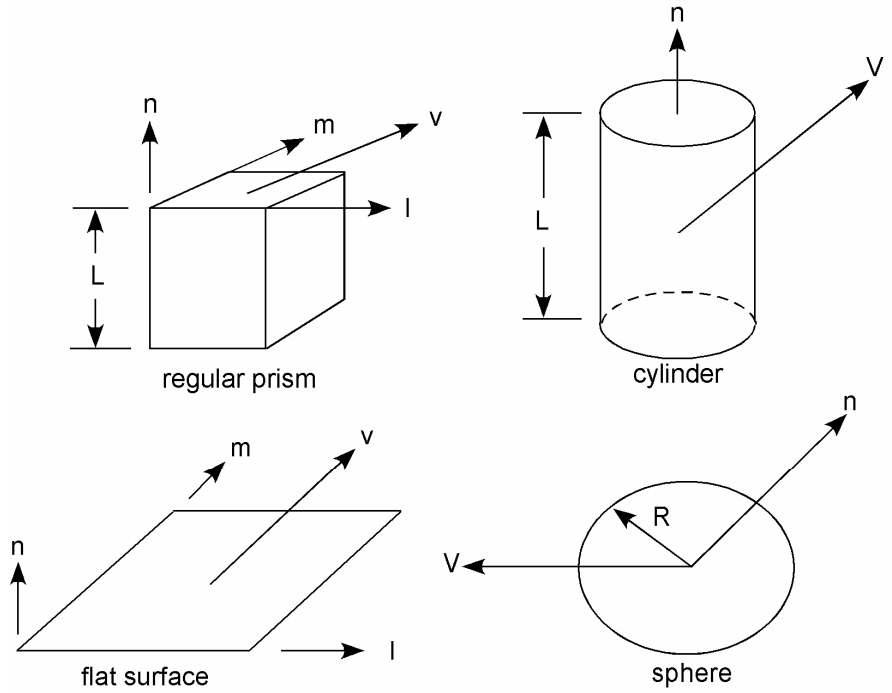


Figure 26.29. Vector **n** determines the orientation of the generalized stonewalls. For the prescribed motion options the wall can be moved in the direction **V** as shown.

26.16 VDA/IGES Contact

This cabability allows the user to read VDA/IGES surfaces directly into LS-DYNA for analysis as contact surfaces. No mesh generation is required, and the contact is performed against the analytic surface. LS-DYNA supports the VDA standard and an important subset of the IGES entities including:

- #100 Circle arc
- #102 Composite Curve
- #106 Copious data
- #110 Lines
- #112 Parametric polynomial curve
- #114 Parametric polynomial surface
- #116 Points
- #126 NURBS Curves
- #128 NURBS Surfaces
- #142 Curve on Parametric Surface
- #144 Trimmed Parametric Surfaces
- #402 form 7-group
- #406 form 15-associate name

First, the user must specify which VDA/IGES surfaces, faces, and groups should be attached to each material. This is done primarily through a special input file. Faces, surfaces, and groups from several different VDA/IGES input files can be combined into groups that later can be refered to by a user specified alias. For example, suppose a simple sheetmetal forming problem is going to be run. The user might have an input file that looks like this:

```
file punch.vda punch.bin {
  alias punch { grp001 }
}
file die.vda die.bin {
  alias part1 { fce001 sur002 }
  alias part2 { fce003 }
}
file die2.vda die2.bin {
  alias part3 { fce004 }
}
file holder.vda holder.bin {
  alias holder { sur001 sur002 }
}
alias die { part1 part2 part3 }
end
```

In this example, the user has specified that the punch will be made up of the group "grp001" from the file "punch.vda". The VDA file is converted to a binary file "punch.bin". If this simulation is ever rerun, the VDA input can be read directly from the binary file thereby significantly reducing startup time. The die in this example is made up of several surfaces and faces from 2 different VDA files. This format of input allows the user to combine any number of faces, surfaces, and

groups from any number of VDA files to define a single part. This single part name is then referenced within the LS-DYNA input file.

The contact algorithm works as follows. For the sake of simplicity, we will refer to one point as being slaved to a single part. Again, this part will in general be made up of several VDA surfaces and faces. First, the distance from the point to each VDA surface is computed and stored. For that surface which is nearest the point, several other parameters are stored such as the surface coordinates of the near point on the surface. Each time step of the calculation this information is updated. For the nearest surface the new near point is calculated. For all other surfaces the distance the point moves is subtracted from the distance to the surface. This continually gives a lower bound on the actual distance to each VDA surface. When this lower bound drops below the thickness of the point being tracked, the actual distance to the surface is recalculated. Actually, if the nearest surface is further away from the point than some distance, the near point on the surface is not tracked at all until the point comes close to some surface. These precautions result in the distance from the point to a surface having to be totally recomputed every few hundred timesteps, in exchange for not having to continually track the point on each surface.

To track the point on the nearest surface, a 2D form of Newton's method is used. The vector function to be solved specifies that the displacement vector from the surface to the point should be parallel to the surface normal vector. The surface tangent vectors are computed with respect to each of the two surface patch parameters, and the dot product taken with the displacement vector. See Figure 26.30 and Equation (26.66).

$$\left(p-q\right)\cdot\frac{\partial q}{\partial s}=0 \quad \textit{and} \quad \left(p-q\right)\cdot\frac{\partial q}{\partial t}=0 \tag{26.54}$$

This vector equation is then solved using Newton's method as in Equation (26.67).

$$q_{i+1}=q_i-\left(F'\right)^{-1}F \tag{26.67}$$

where

$$F\left(s,t\right)=\begin{pmatrix}\left(p-q\right)\cdot\dfrac{\partial q}{\partial s}\\ \left(p-q\right)\cdot\dfrac{\partial q}{\partial t}\end{pmatrix} \tag{26.68}$$

The convergence is damped in the sense that the surface point is not allowed to jump completely outside of a surface patch in one iteration. If the iteration point tries to leave a patch, it is placed in the neighboring patch, but on the adjoining boundary. This prevents the point from moving merely continuous (i.e., when the surface has a crease in it). Iteration continues until the maximum number of allowed iterations is reached, or a convergence tolerance is met. The convergence tolerance (as measured in the surface patch parameters) varies from patch to patch, and is based on the size and shape of the patch. The convergence criterion is set for a patch to ensure that the actual surface point has converged (in the spatial parameters x, y, and z) to some tolerance.

26.17 Simulated Draw Beads

The implementation of draw beads is based on elastic-plastic interface springs and nodes-to-surface contact. The area of the blank under the draw bead is taken as the master surface. The draw bead is defined by a *consecutive* list of nodes that lie along the draw bead. For straight draw beads only two nodes need to be defined, but for curved beads sufficient nodes must be used to define the curvature. The draw bead line is discretized into points that become the slave nodes to the master surface. The spacing of the points is determined by LS-DYNA such that several points lie within each master segment. This is illustrated in Figure 26.32. The dense distribution of point leads to a smooth draw bead force distribution which helps avoid exciting the zero energy (hourglass) modes within the shell elements in the workpiece. A three-dimensional bucket search is used for the contact searching to locate each point within a segment of the master surface.

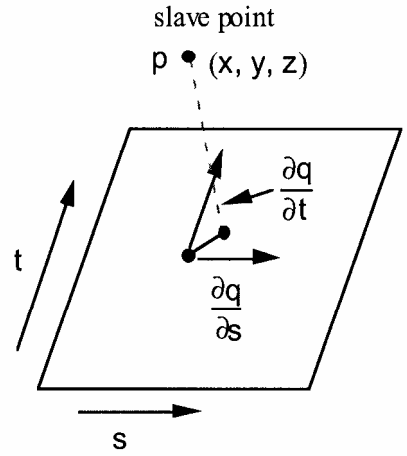


Figure 26.30. The geometry of the patch is a function of the parametric coordinates s and t.

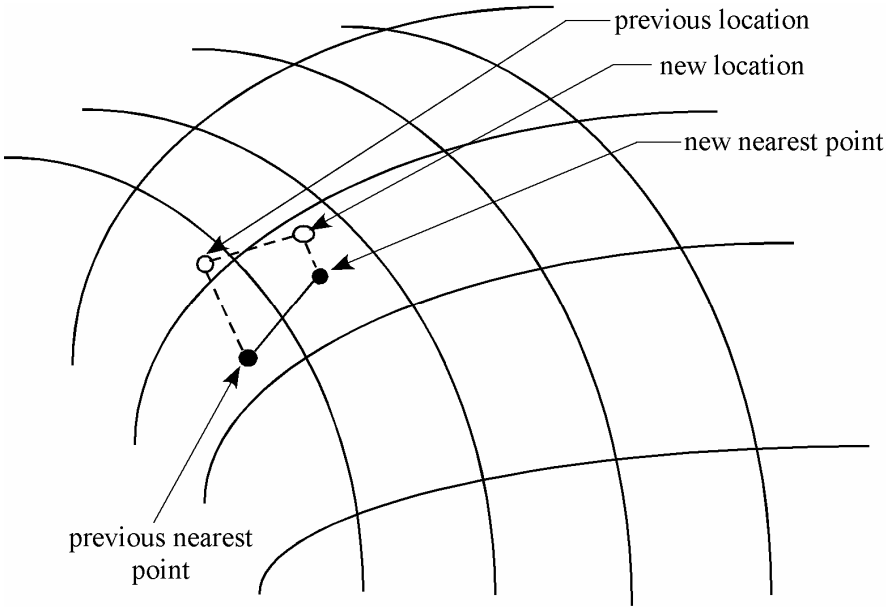


Figure 26.31. Newton iteration solves for the nearest point on the analytical surface.

The nodes defining the draw beads can be attached to rigid bodies by using the extra nodes for rigid body input option. When defining draw beads, care should be taken to limit the number of elements that are used in the master surface definition. If the entire blank is specified the CPU cost increases significantly and the memory requirements can become enormous. An automated draw bead box, which is defined by specifying the part ID for the workpiece and the node set ID for the draw bead, is available. The automated box option allows LS-DYNA determine the box dimensions. The size of this box is based on the extent of the blank and the largest element in the workpiece as shown if Figure 26.33.

The input for the draw beads requires a load curve giving the force due to the bending and unbending of the blank as it moves through the draw bead. The load curve may also include the effect of friction. However, the coulomb friction coefficients must be set to zero if the frictional component is included in the load curve. If the sign of the load curve ID is *positive* the load curve gives the retaining force per unit draw bead length as a function of displacement, δ. If the sign is *negative* the load curve defines the maximum retaining force versus the normalized position along the draw bead. This position varies from 0 (at the origin) to 1 (at the end) along the draw bead. See Figures 26.34 and 26.35.

When friction is active the frictional force component normal to the bead in the plane of the work piece is computed. Frictional forces tangent to the bead are not allowed.

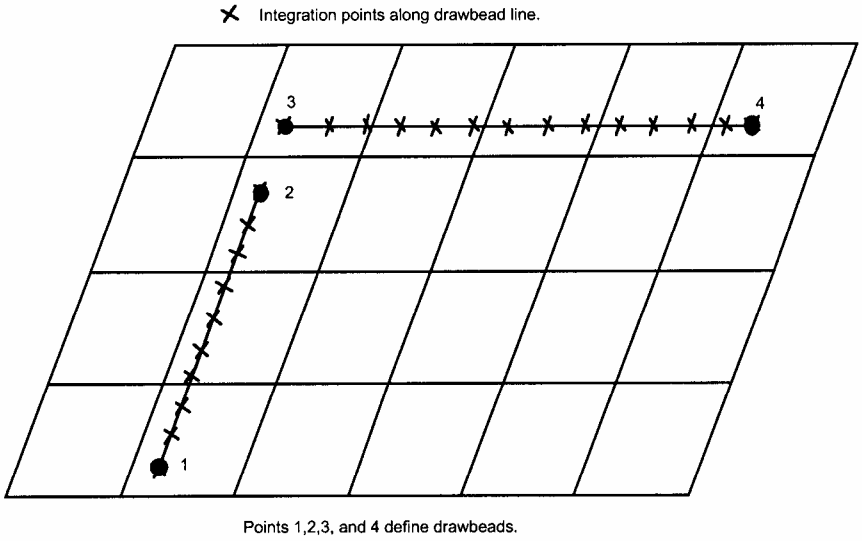


Figure 26.32. The drawbead contact provides a simple way of including drawbead behavior without the necessity of defining a finite element mesh for the drawbeads. Since the draw bead is straight, each bead is defined by only two nodes.

The second load curve gives the normal force per unit draw bead length as a function of displacement, δ. This force is due to bending the blank into the draw bead as the binder closes on the die and represents a limiting value. The normal force begins to develop when the distance between the die and binder is less than the draw bead depth. *As the binder and die close on the blank this force should diminish or reach a plateau.* This load curve was originally added to stabilize the calculation.

As the elements of the blank move under the draw bead, a plastic strain distribution develops through the shell thickness due to membrane stretching and bending. To account for this strain profile an optional load curve can be defined that gives the plastic strain versus the parametric coordinate through the shell thickness where the parametric coordinate is defined in the interval from –1 to 1. The value of the plastic strain at each through thickness integration point is interpolated from this curve. If the plastic strain at an integration point exceeds the value of the load curve at the time initialization occurs, the plastic strain at the point will remain unchanged. A scale factor that multiplies the shell thickness as the shell element moves under the draw bead can also be defined as a way of accounting for any thinning that may occur.

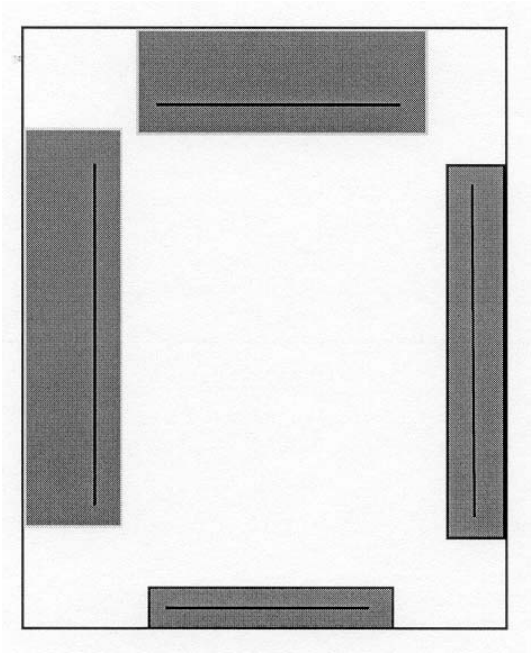


Figure 26.33. The draw bead box option automatically sizes the box around the draw bead. Any segments within the box are included as master segments in the contact definition.

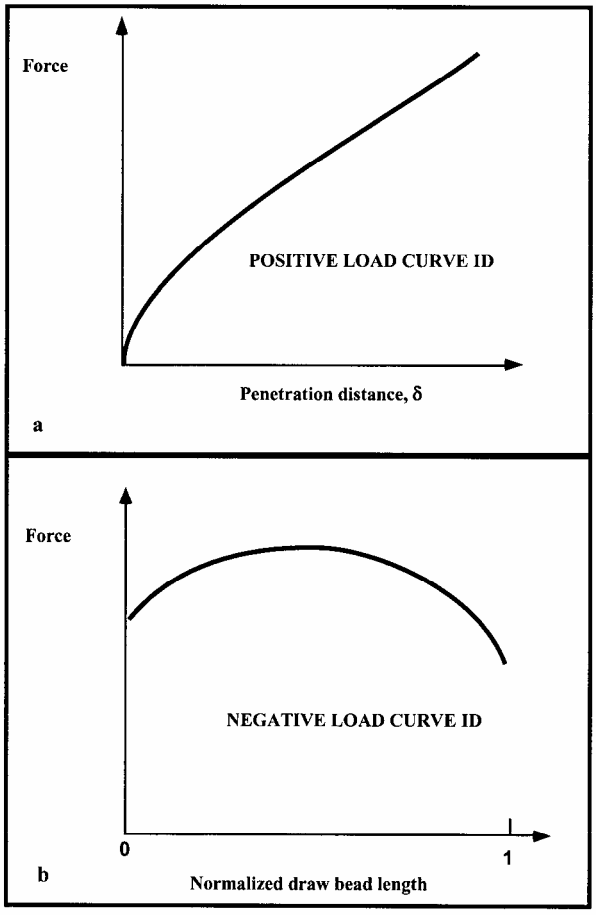


Figure 26.34. Draw bead contact model defines a resisting force as a function of draw bead displacement.

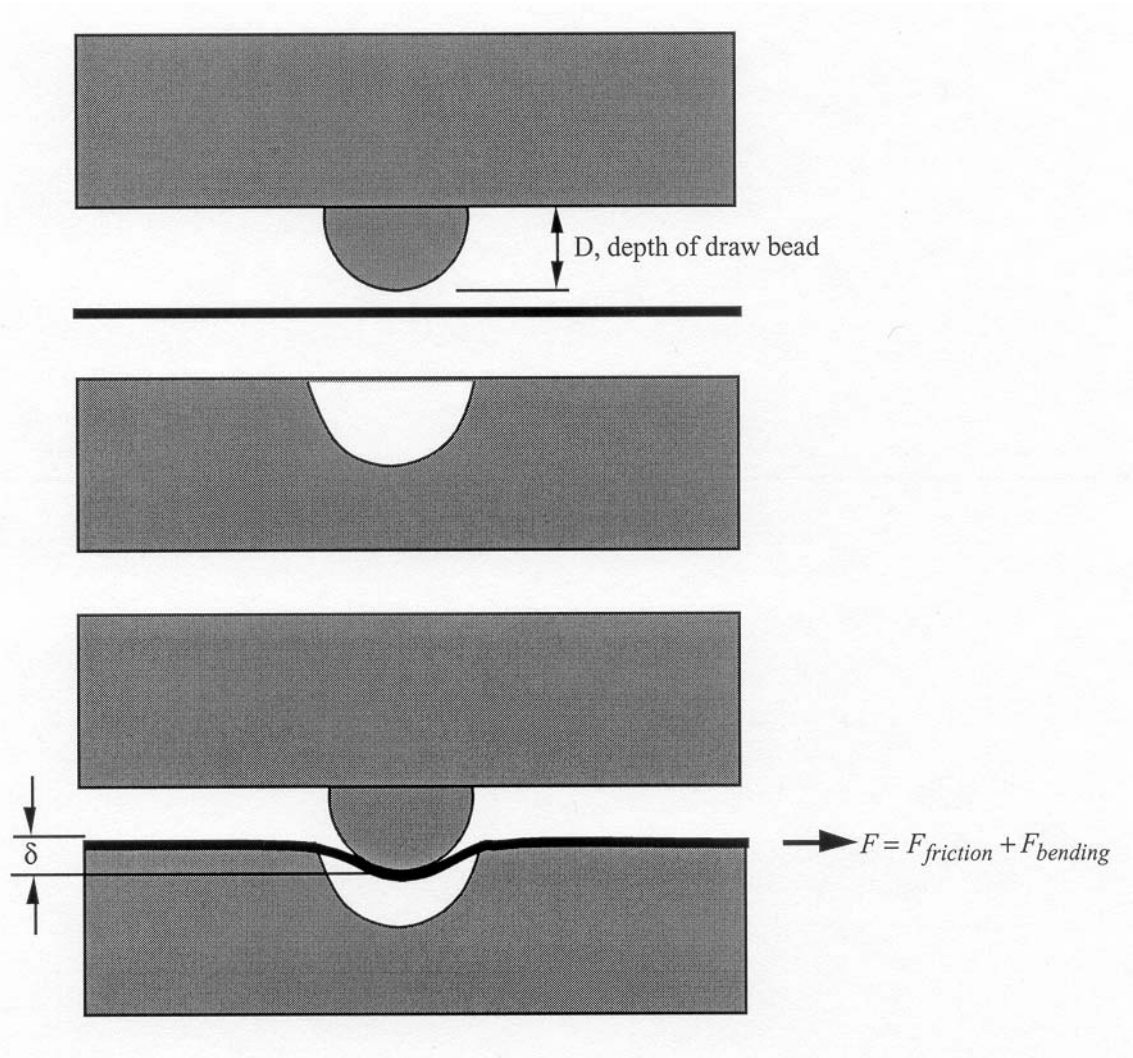


Figure 26.35. Draw bead contact model defines a resisting force as a function of draw bead displacement.

26.18 Edge to Edge Contact

Edge to edge contact can be important in some simulations. For example, if a fan blade breaks away from the hub in a jet turbine contact with the trailing blade will likely be along the edges of the blades. Edge to edge contact requires a special treatment since the nodal points do not make contact with the master segment which is the basis of the conventional contact treatments. Currently all automatic type contact possess edge-to-edge capabilities and therefore contact type 22 is only useful with those contact that do not possess this capability. All contact using the segment-based formulation have edge to edge capabilities.

The basis of single edge contact is the proven single surface formulation and the input is identical. The definition is by material ID. Edge determination is automatic. It is also possible to use a manual definition by listing line segments. The single edge contact is type 22 in the structured input or *CONTACT_SINGLE_EDGE in the keyword input.

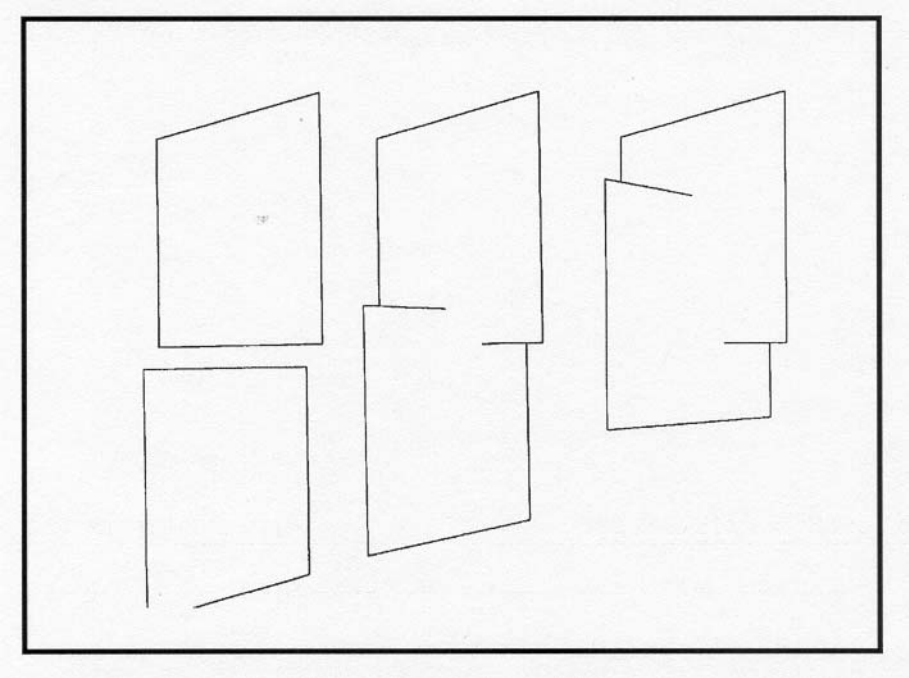


Figure 26.36. Contact between edges requires a special treatment since the nodes do not make contact.

This contact only considers edge to edge contact of the type illustrated in Figure 26.37. Here the tangent vectors to the plane of the shell and normal to the edge must point to each other for contact to be considered.

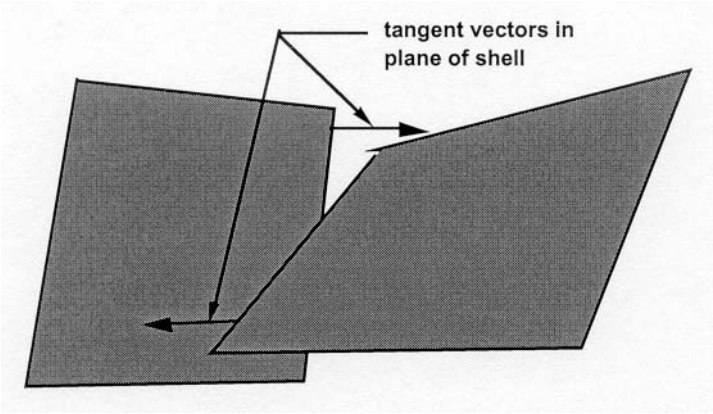


Figure 26.37. Single edge contact considers contact between two edges whose normals point towards each other.

26.19 Beam to Beam Contact

In the beam to beam contact the contact surface is assumed to be the surface of a cylinder as shown in Figure 26.38. The diameter of the contact cylinder is set equal to the square root of the area of the smallest rectangle that contains the cross section to avoid tracking the orientation of the beam within the contact algorithm. Contact is found by finding the intersection point between nearby beam elements and checking to see if their outer surfaces overlap as seen in Figure 26.39. If the surfaces overlap the contact force is computed and is applied to the nodal points of the interacting beam elements.

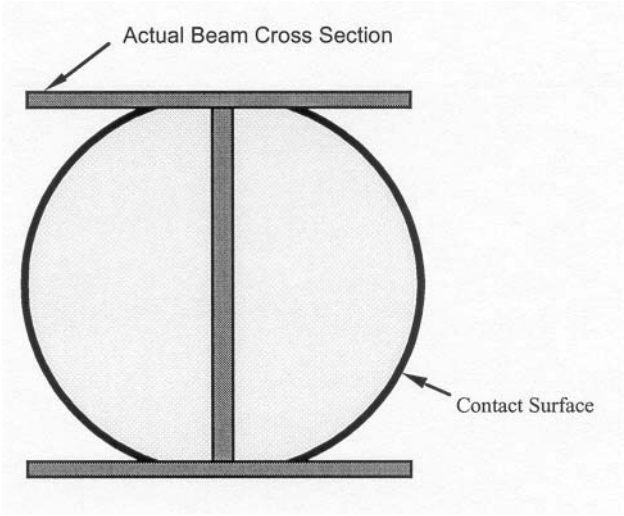


Figure 26.38. Beam contact surface approximation.

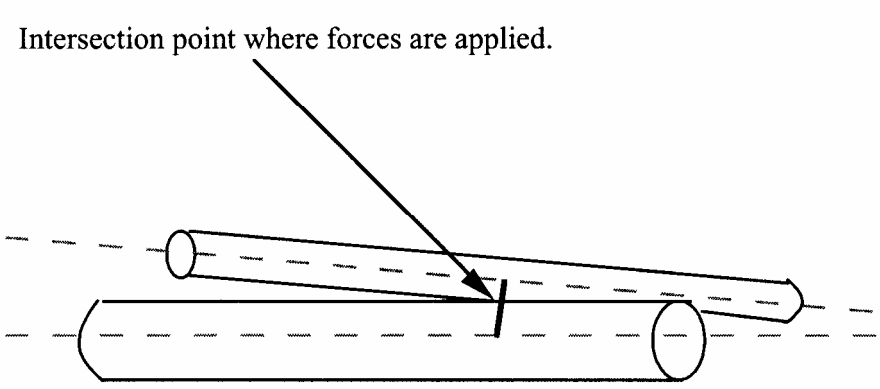


Figure 26.39. The forces are applied at the intersection point.

27. GEOMETRIC CONTACT ENTITIES

Contact algorithms in LS-DYNA currently can treat any arbitrarily shaped surface by representing the surface with a faceted mesh. Occupant modeling can be treated this way by using fine meshes to represent the head or knees. The generality of the faceted mesh contact suffers drawbacks when modeling occupants, however, due to storage requirements, computing costs, and mesh generation times. The geometric contact entities were added as an alternate method to model cases of curved rigid bodies impacting deformable surfaces. Much less storage is required and the computational cost decreases dramatically when compared to the more general contact.

Geometric contact entities are developed using a standard solids modeling approach. The geometric entity is defined by a scalar function $G(x,y,z)$. The solid is determined from the scalar function as follows:

$G(x,y,z) > 0$ The point (x,y,z) is outside the solid. (27.1)

$G(x,y,z) = 0$ The point (x,y,z) is on the surface of the solid. (27.2)

$G(x,y,z) < 0$ The point (x,y,z) is inside the solid. (27.3)

Thus, by a simple function evaluation, a node can be immediately determined to be outside the solid or in contact. Figure 27.1 illustrates this for a cylinder.

If the node is in contact with the solid, a restoring force must be applied to eliminate further penetration. A number of methods are available to do this such as Lagrange multipliers or momentum based methods. The penalty method was selected because it is the simplest and most efficient method. Also, in our applications the impact velocities are at a level where the penalty methods provide almost the identical answer as the exact solution.

Using the penalty method, the restoring force is proportional to the penetration distance into the solid and acts in the direction normal to the surface of the solid. Thus, the penetration distance and the normal vector must be determined. The surface normal vector is conveniently determined from the gradient of the scalar function.

$$\vec{N}(x,y,z) = \frac{\frac{\partial G}{\partial x} i + \frac{\partial G}{\partial y} j + \frac{\partial G}{\partial z} k}{\sqrt{\left(\frac{\partial G}{\partial x}\right)^2 + \left(\frac{\partial G}{\partial y}\right)^2 + \left(\frac{\partial G}{\partial z}\right)^2}} \tag{27.4}$$

for all (x,y,z) such that $G(x,y,z) = 0$. The definition of $G(x,y,z)$ guarantees that this vector faces in the outward direction. When penetration does occur, the function $G(x,y,z)$ will be slightly less than zero. For curved surfaces this will result in some errors in calculating the normal vector, because it is not evaluated exactly at the surface. In an implicit code, this would be important, however, the explicit time integration scheme in DYNA3D uses such a small time step that penetrations are negligible and the normal function can be evaluated directly at the slave node ignoring any penetration.

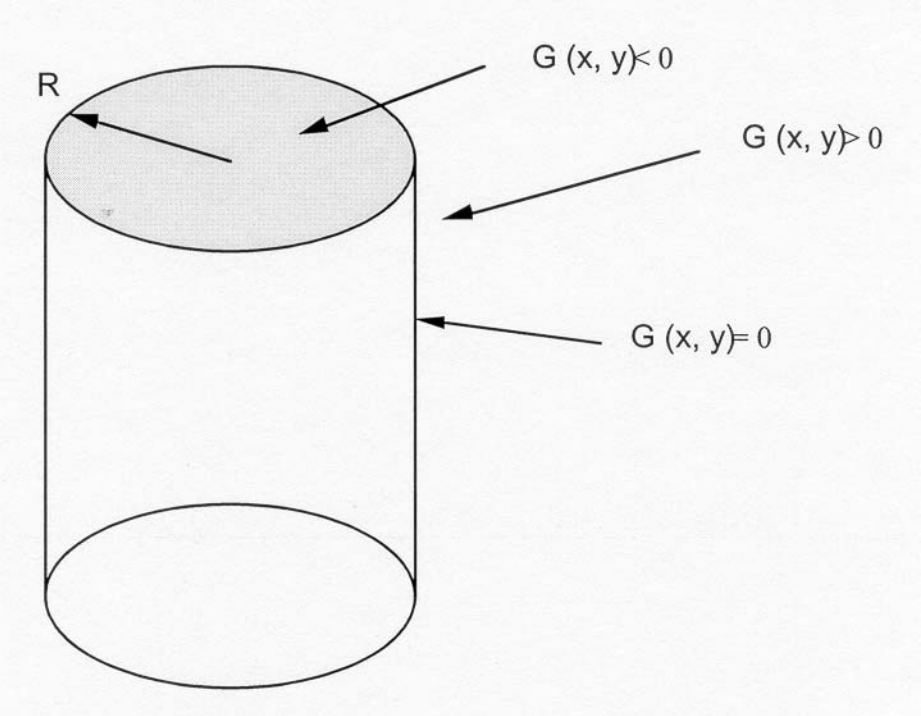


Figure 27.1. Determination of whether a node is interior or exterior to the cylindrical surface $G(x,y) = x^2 + y^2 - R^2$

The penetrations distance is the last item to be calculated. In general, the penetration distance, d, is determined by.

$$d = \left| \vec{X}_n - \vec{X}'_n \right| \tag{27.5}$$

where,

$\vec{X}_n$ is the location of node n and

$\vec{X}'_n$ is the nearest point on the surface of the solid.

To determine $\vec{X}'_n$, a line function is defined which passes through $\vec{X}_n$ and is normal to the surface of the solid:

$$\vec{L}(s) = \vec{X}_n + s\vec{N}\left(\vec{X}_n\right) \tag{27.6}$$

Substituting the line function into the definition of the Equation (27.2) surface of a solid body gives:

$$G\left(\vec{X}_n + s\vec{N}\left(\vec{X}_n\right)\right) = 0 \tag{27.7}$$

If Equation (27.7) has only one solution, this provides the parametric coordinates s which locates $\vec{X}_n'$. If Equation (27.7) has more than one root, then the root which minimizes Equation (27.5) locates the point $\vec{X}_n'$.

The penalty method defines the restoring forces as:

$$\vec{f} = pd\vec{N}\left(\vec{\bar{X}}_n'\right) \tag{27.8}$$

where p is a penalty factor and is effectively a spring constant. To minimize the penetration of the slave node into the solid, the constant p is set large, however, it should not be set so large that the Courant stability criteria is violated. This criteria for the slave node tells us that:

$$\Delta t \le \frac{2}{\omega_{\max}} = \frac{2}{\sqrt{\frac{K_n}{m_n}}} = 2\sqrt{\frac{m_n}{K_n}} \tag{27.9}$$

where, K_n is the stiffness of node n and m_n is the mass of node n.

The penalty factor, p, is determined by choosing a value which results in a penalty/slave mass oscillator which has a characteristic time step that is ten times larger than the Courant time step:

$$10\Delta t = 2\sqrt{\frac{m_n}{p_n}} \tag{27.10}$$

Solving for p_n gives:

$$p_n = \frac{4m_n}{\left(100\Delta t^2\right)} \tag{27.11}$$

Inclusion of any structural elements into the occupant model will typically result in very large stiffnesses due to the small time step and the $(1/\Delta t)^2$ term. Thus the method is highly effective even with impact velocities on the order of 1km/sec.

The scalar function $G\left(\vec{X}\right)$ is frequently more conveniently expressed as $g\left(\vec{x}\right)$ where, g is the function defined in local coordinates and $\vec{x}$ is the position in local coordinates. The local entity is related to the global coordinates by:

$$\vec{x} = [T]\left(\vec{X}_j - \vec{Q}_j\right) \tag{27.12}$$

where $\vec{Q}_j$ is the offset and $[T]$ is a rotation matrix. The solid scalar function and the penetration distance can be evaluated in either local or global coordinates with no difference to the results.

When working in local coordinates, the gradient of the local scalar function provides a normal vector which is in the local system and must be transformed into the global by:

$$\vec{N}\left(\vec{\bar{X}}\right)=\left[T\right]^{T}\vec{n}\left(\vec{x}\right) \tag{27.13}$$

An ellipsoid is defined by the function:

$$G\left(x,y,z\right)=\left(\frac{x}{a}\right)^{2}+\left(\frac{y}{b}\right)^{2}+\left(\frac{z}{c}\right)^{2}-1 \tag{27.14}$$

The gradient of G is $\frac{2x}{a^2}i+\frac{2y}{a^2}j+\frac{2z}{a^2}k$ and the normal vector is:

$$n\left(x,y,z\right)=\frac{\left(\frac{x}{a^2}i+\frac{y}{b^2}j+\frac{z}{c^2}k\right)}{\sqrt{\frac{x^2}{a^4}+\frac{y^2}{b^4}+\frac{z^2}{c^4}}} \tag{27.15}$$

Substituting Equations (27.6) and (27.14) into Equation (27.2) gives:

$$\left[\left(\frac{n_x}{a}\right)^2+\left(\frac{n_y}{b}\right)^2+\left(\frac{n_z}{c}\right)^2\right]s^2+2\left[\frac{n_x x_n}{a^2}+\frac{n_y y_n}{b^2}+\frac{n_z z_n}{c^2}\right]s$$
$$+\left[\left(\frac{x_n}{a}\right)^2+\left(\frac{y_n}{b}\right)^2+\left(\frac{z_n}{c}\right)^2-1\right]=0$$

Solving this quadratic equation for s provides the intercepts for the nearest point on the ellipsoid and the opposite point of the ellipsoid where the normal vector, $\vec{X}_n$, also points toward.

Currently, this method has been implemented for the case of an infinite plane, a cylinder, a sphere, and an ellipsoid with appropriate simplifications. The ellipsoid is intended to be used with rigid body dummy models. The methods are, however, quite general so that many more shapes could be implemented. A direct coupling to solids modeling packages should also be possible in the future.

28. NODAL CONSTRAINTS

In this section nodal constraints and linear constraint equations are described.

28.1 Nodal Constraint Sets

This option forces groups of nodes to move together with a common translational acceleration in either one or more degrees of freedom. The implementation is straightforward with the common acceleration defined by

$$a_{i_{common}} = \frac{\sum_{j}^{n} M_j a_i^j}{\sum_{j}^{n} M_j} \tag{28.1}$$

where n is the number of nodes, a_i^j is the acceleration of the jth constrained node in the ith direction, and $a_{i_{common}}$ is the common acceleration.

Nodal constraint sets eliminate rigid body rotations in the body that contains the node set and, therefore, must be applied very cautiously.

28.2 Linear Constraint Equations

Linear constraint equations of the form:

$$\sum_{k=1}^{n} C_k u_k = C_0 \tag{28.2}$$

can be defined where n is the number of constrained degrees of freedom, u_k is a constrained nodal displacement, and the C_k are user-defined coefficients. Unless LS-DYNA is initialized by linking to an implicit code to satisfy this equation at the beginning of the calculation, the constant C_0 is assumed to be zero. The first constrained degree of freedom is eliminated from the equations of motion:

$$u_1 = C_0 - \sum_{k=2}^{n} \frac{C_k}{C_1} u_k \tag{28.3}$$

Its velocities and accelerations are given by

$$\begin{aligned} \dot{u}_1 &= -\sum_{k=2}^{n} \frac{C_k}{C_1} \dot{u}_k \\ \ddot{u}_1 &= -\sum_{k=2}^{n} \frac{C_k}{C_1} \ddot{u}_k \end{aligned} \tag{28.4}$$

respectively. In the implementation a transformation matrix L is constructed relating the unconstrained u and constrained $u_{constrained}$ degrees of freedom. The constrained accelerations used in the above equation are given by:

$$u_{constrained} = \left[L^t M L \right]^{-1} L^t F \tag{28.5}$$

where M is the diagonal lumped mass matrix and F is the righthand side force vector. This requires the inversion of the condensed mass matrix which is equal in size to the number of constrained degrees of freedom minus one. The inverse of the condensed mass matrix is computed in the initialization phase and stored in core.

29. VECTORIZATION AND PARALLELIZATION

29.1 Vectorization

In 1978, when the author first vectorized DYNA3D on the CRAY-1, a four-fold increase in speed was attained. This increase was realized by recoding the solution phase to process vectors in place of scalars. It was necessary to process elements in groups rather than individually as had been done earlier on the CDC-7600 supercomputers.

Since vector registers are generally some multiple of 64 words, vector lengths of 64 or some multiple are appropriate. In LS-DYNA, groups of 128 elements or possibly some larger integer multiple of 64 are utilized. Larger groups give a marginally faster code, but can reduce computer time sharing efficiency because of increased core requirements. If elements within the group reference more than one material model, subgroups are formed for consecutive elements that reference the same model. LS-DYNA internally sorts elements by material to maximize vector lengths.

Conceptually, vectorization is straightforward. Each scalar operation that is normally executed once for one element, is repeated for each element in the group. This means that each scalar is replaced by an array, and the operation is put into a DO-loop. For example, the nodal force calculation for the hexahedron element appeared in a scalar version of DYNA3D as:

```
E11 = SGV1*PX1+SGV4*PY1+SGV6*PZ1
E21 = SGV2*PY1+SGV4*PX1+SGV5*PZ1
E31 = SGV3*PZ1+SGV6*PX1+SGV5*PY1
E12 = SGV1*PX2+SGV4*PY2+SGV6*PZ2
E22 = SGV2*PY2+SGV4*PX2+SGV5*PZ2
E32 = SGV3*PZ2+SGV6*PX2+SGV5*PY2
E13 = SGV1*PX3+SGV4*PY3+SGV6*PZ3
E23 = SGV2*PY3+SGV4*PX3+SGV5*PZ3
E33 = SGV3*PZ3+SGV6*PX3+SGV5*PY3
E14 = SGV1*PX4+SGV4*PY4+SGV6*PZ4
E24 = SGV2*PY4+SGV4*PX4+SGV5*PZ4
E34 = SGV3*PZ4+SGV6*PX4+SGV5*PY4
```

and in the vectorized version as:

```
      DO 110 I=LFT, LLT
      E11(I)  = SGV1(I)*PX1(I)+SGV4(I)*PY1(I)+SGV6(I)*PZ1(I)
      E21(I)  = SGV2(I)*PY1(I)+SGV4(I)*PX1(I)+SGV5(I)*PZ1(I)
      E31(I)  = SGV3(I)*PZ1(I)+SGV6(I)*PX1(I)+SGV5(I)*PY1(I)
      E12(I)  = SGV1(I)*PX2(I)+SGV4(I)*PY2(I)+SGV6(I)*PZ2(I)
      E22(I)  = SGV2(I)*PY2(I)+SGV4(I)*PX2(I)+SGV5(I)*PZ2(I)
      E32(I)  = SGV3(I)*PZ2(I)+SGV6(I)*PX2(I)+SGV5(I)*PY2(I)
      E13(I)  = SGV1(I)*PX3(I)+SGV4(I)*PY3(I)+SGV6(I)*PZ3(I)
      E23(I)  = SGV2(I)*PY3(I)+SGV4(I)*PX3(I)+SGV5(I)*PZ3(I)
      E33(I)  = SGV3(I)*PZ3(I)+SGV6(I)*PX3(I)+SGV5(I)*PY3(I)
      E14(I)  = SGV1(I)*PX4(I)+SGV4(I)*PY4(I)+SGV6(I)*PZ4(I)
      E24(I)  = SGV2(I)*PY4(I)+SGV4(I)*PX4(I)+SGV5(I)*PZ4(I)
110   E34(I)  = SGV3(I)*PZ4(I)+SGV6(I)*PX4(I)+SGV5(I)*PY4(I)
```

where $1 \le LFT \le LLT \le n$. Elements LFT to LLT inclusive use the same material model and n is an integer multiple of 64.

Gather operations are vectorized on most supercomputers. In the gather operation, variables needed for processing the element group are pulled from global arrays into local vectors. For example, the gather operation:

```
      DO 10 I      =     LFT, LLT
      X1(I)  =     X(1,IX1(I))
      Y1(I)  =     X(2,IX1(I))
      Z1(I)  =     X(3,IX1(I))
      VX1(I) =     V(1,IX1(I))
      VY1(I) =     V(2,IX1(I))
      VZ1(I) =     V(3,IX1(I))
      X2(I)  =     X(1,IX2(I))
      Y2(I)  =     X(2,IX2(I))
      Z2(I)  =     X(3,IX2(I))
      VX2(I) =     V(1,IX2(I))
      VY2(I) =     V(2,IX2(I))
      VZ2(I) =     V(3,IX2(I))
      X3(I)  =     X(1,IX3(I))
      X3(I)  =     X(2,IX3(I))
      X3(I)  =     X(3,IX3(I))
        X8(I)    =  X(1,IX8(I))
        Y8(I)    =  X(2,IX8(I))
        Z8(I)    =  X(3,IX8(I))
       VX8(I)    =  V(1,IX8(I))
       VY8(I)    =  V(2,IX8(I))
10     VZ8(I)    =  V(3,IX8(I))
```

initializes the nodal velocity and coordinate vector for each element in the subgroup LFT to LLT. In the scatter operation, element nodal forces are added to the global force vector. The force assembly does not vectorize unless special care is taken as described below.

In general, the element force assembly is given in FORTRAN by:

```
      DO 30 I=1,NODFRC
      DO 20 N=1,NUMNOD
      DO 10 L=LFT,LLT
      RHS(I,IX(N,L))=RHS(I,IX(N,L))+FORCE(I,N,L)
   10 CONTINUE
   20 CONTINUE
   30 CONTINUE
```

where NODFRC is the number of force components per node (3 for solid elements, 6 for shells), LFT and LLT span the number of elements in the vector block, NUMNOD is the number of nodes defining the element, FORCE contains the force components of the individual elements, and RHS is the global force vector. This loop does not vectorize since the possibility exists that more that one element may contribute force to the same node. FORTRAN vector compilers

recognize this and will vectorize only if directives are added to the source code. If all elements in the loop bounded by the limits LFT and LLT are disjoint, the compiler directives can be safely added. We therefore attempt to sort the elements as shown in Figure 28.1 to guarantee disjointness.

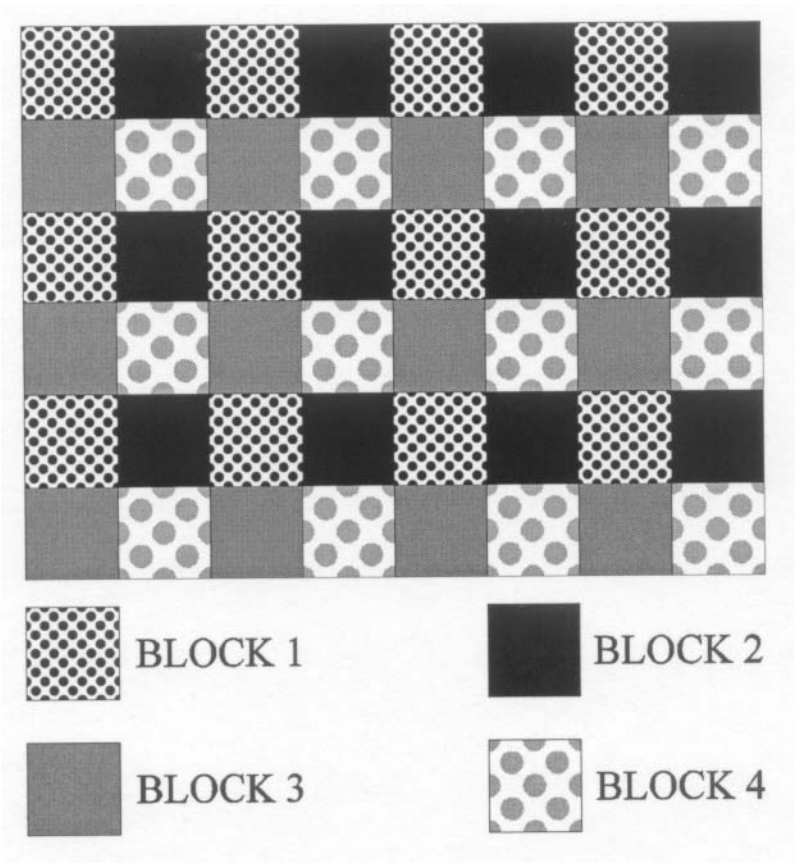


Figure 29.1. Group of 48 elements broken into 4 disjoint blocks.

The current implementation was strongly motivated by Benson [1989] and by work performed at General Motors [Ginsberg and Johnson 1988, Ginsberg and Katnik 1989], where it was shown that substantial improvements in execution speed could be realized by blocking the elements in the force assembly. Katnik implemented element sorting in a public domain version of DYNA3D for the Belytschko-Tsay shell element and added compiler directives to force vectorization of the scatter operations associated with the addition of element forces into the global force vector. The sorting was performed immediately after the elements were read in so that subsequent references to the stored element data were sequential. Benson performed the sorting in the element loops via indirect addressing. In LS-DYNA the published GM approach is taken.

Implementation of the vectorization of the scatter operations is implemented in for all elements including the solid, shell, membrane, beam, and truss elements. The sorting is completely transparent to the user.

29.2 Parallelization

In parallelization, the biggest hurdle is overcoming Amdahl's law for multitasking [Cray Research Inc. 1990]

$$S_m = \frac{1}{f_s + \frac{f_p}{N}}$$

where

S_m = maximum expected speedup from multitasking
N = number of processors available for parallel execution
f_p = fraction of a program that can execute in parallel
f_s = fraction of a program that is serial

Table 29.1 shows that to obtain a speed factor of four on eight processors it is necessary to have eighty-six percent of the job running in parallel. Obviously, to gain the highest speed factors the entire code must run in parallel.

LS-DYNA has been substantially written to function on all shared memory parallel machine architectures. Generally, shared memory parallel speed-ups of 5 on 8 processors are possible but this is affected by the machine characteristics. We have observed speeds of 5.6 on full car crash models on a machine of one manufacturer only to see a speed-up of 3.5 on a different machine of another manufacturer.

%	N=2	N=4	N=8	N=16	N=32	N=64	N=128	N=256
86.0%	1.75	2.82	4.04	5.16	5.99	6.52	6.82	6.98
90.0%	1.82	3.08	4.71	6.40	7.80	8.77	9.34	9.66
92.0%	1.85	3.23	5.13	7.27	9.20	10.60	11.47	11.96
94.0%	1.89	3.39	5.63	8.42	11.19	13.39	14.85	15.71
96.0%	1.92	3.57	6.25	10.00	14.29	18.18	21.05	22.86
98.0%	1.96	3.77	7.02	12.31	19.75	28.32	36.16	41.97
99.0%	1.98	3.88	7.48	13.91	24.43	39.26	56.39	72.11
99.2%	1.98	3.91	7.58	14.29	25.64	42.55	63.49	84.21
99.4%	1.99	39.3	7.68	14.68	26.98	46.44	72.64	101.19
99.6%	1.99	3.95	7.78	15.09	28.47	51.12	84.88	126.73
99.7%	1.99	3.96	7.84	15.31	29.28	53.83	92.69	145.04
99.8%	2.00	3.98	7.89	15.53	30.13	56.84	102.07	169.54
99.9%	2.00	3.99	7.94	15.76	31.04	60.21	113.58	203.98
100.0%	2.00	4.00	8.00	16.00	32.00	64.00	128.00	256.00

Table 29.1. Maximum theoretical speedup S_m, on N CPUs with parallelism [Cray Research Inc. 1990].

In the element loops element blocks with vector lengths of 64 or some multiple are assembled and sent to separate processors. All elements are processed in parallel. On the

average a speed factor of 7.8 has been attained in each element class corresponding to 99.7% parallelization.

A significant complication in parallelizing code is that a variable can sometimes be updated simultaneously by different processors with incorrect results. To force the processors to access and update the variable in a sequential manner, a GUARD compiler directive must be introduced. This results in an interruption to the parallel execution and can create a bottleneck. By sorting the data in the parallel groups or by allocating additional storage it is usually possible to eradicate the GUARDS from the coding. The effort may not be worth the gains in execution speed.

The element blocks are defined at the highest level and each processor updates the entire block including the right hand side force assembly. The user currently has two options: GUARD compiler directives prevent simultaneous updates of the RHS vector (recommended for single CPU processors or when running in a single CPU mode on a multi-processor), or assemble the right hand side in parallel and let LS-DYNA prevent conflicts between CPU's. This usually provides the highest speed and is recommended, i.e., no GUARDS.

When executing LS-DYNA in parallel, the order of operations will vary from run to run. This variation will lead to slightly different numerical results due to round-off errors. By the time the calculation reaches completion variations in nodal accelerations and sometimes even velocities are observable. These variations are independent of the precision and show up on both 32 and 64 bit machines. There is an option in LS-DYNA to use an ordered summation of the global right hand side force vector to eliminate numerical differences. To achieve this the element force vectors are stored. After leaving the element loop, the global force vector is assembled in the same order that occurs on one processor. The ordered summation option is slower and uses more memory than the default, but it leads to nearly identical, if not identical, results run to run.

Parallelization in LS-DYNA was initially done with vector machines as the target where the vector speed up is typically 10 times faster than scalar. On vector machines, therefore, vectorization comes first. If the problem is large enough then parallelization is automatic. If vector lengths are 128, for example, and if 256 beam elements are used only a factor of 2 in speed can be anticipated while processing beam elements. Large contact surfaces will effectively run in parallel, small surfaces having under 100 segments will not. The speed up in the contact subroutines has only registered 7 on 8 processors due to the presence of GUARD statements around the force assembly. Because real models often use many special options that will not even vectorize efficiently it is unlikely that more than 95% of a given problem will run in parallel on a shared memory parallel machine.

30. AIRBAGS

Additional information on the airbag modeling and comparisons with experimental data can be found in a report [Hallquist, Stillman, Hughes, and Tarver 1990] based on research sponsored by the Motor Vehicles Manufacturers Association (MVMA).

30.1 Control Volume Modeling

A direct approach for modeling the contents of the airbag would be to discretize the interior of the airbag using solid elements. The total volume and pressure-volume relationship of the airbag would then be the sum of all the elemental contributions. Although this direct approach could be applied in a straight forward manner to an inflated airbag, it would become very difficult to implement during the inflation phase of the airbag deployment. Additionally, as the model is refined, the solid elements would quickly overwhelm all other computational costs and make the numerical simulations prohibitively expensive.

An alternative approach for calculating the airbag volume, that is both applicable during the inflation phase and less computationally demanding, treats the airbag as a control volume. The control volume is defined as the volume enclosed by a surface. In the present case, the 'control surface' that defines the control volume is the surface modeled by shell or membrane elements comprising the airbag fabric material.

Because the evolution of the control surface is known, i.e., the position, orientation, and current surface area of the airbag fabric elements are computed and stored at each time step, we can take advantage of these properties of the control surface elements to calculate the control volume, i.e., the airbag volume. The area of the control surface can be related to the control volume through Green's Theorem

$$\iiint \phi \frac{\partial \psi}{\partial x} dxdydz = -\iiint \psi \frac{\partial \phi}{\partial x} dxdydz + \oint \phi \psi n_x d\Gamma \tag{30.1}$$

where the first two integrals are integrals over a closed volume, i.e., $dv = dxdydz$, the last integral is an integral over the surface enclosing the volume, and n_x is the direction cosine between the surface normal and the x direction (corresponding to the x-partial derivative); similar forms can be written for the other two directions. The two arbitrary functions ϕ and ψ need only be integrated over the volume and surface.

The integral form of the volume can be written as

$$V = \iiint dxdydz \tag{30.2}$$

Comparing the first of the volume integrals in Equation (30.1) to Equation (30.2), we can easily obtain the volume integral from Equation (30.1) by choosing for the two arbitrary functions

$$\phi = 1 \tag{30.3}$$

$$\psi = x_x \tag{30.4}$$

leading to

$$V = \iiint dxdydz = \oint xn_x d\Gamma \tag{30.5}$$

The surface integral in Equation (30.5) can be approximated by a summation over all the elements comprising the airbag, i.e.,

$$\oint xn_x d\Gamma \approx \sum_{i=1}^{N} \bar{x}_i n_{ix} A_i \tag{30.6}$$

where for each element i: $\bar{x}_i$ is the average x coordinate, n_{ix} is the direction cosine between the elements normal and the x direction, and A_i is the surface area of the element.

Although Equation (30.5) will provide the exact analytical volume for an arbitrary direction, i.e., any n, the numerical implementation of Equation (30.5), and its approximation Equation (30.6), has been found to produce slightly different volumes, differing by a few percent, depending on the choice of directions: if the integration direction is nearly parallel to a surface element, i.e., the direction cosine is nearly zero, numerical precision errors affect the volume calculation. The implementation uses as an integration direction, a direction that is parallel to the maximum principle moment of inertia of the surface. Numerical experiments have shown this choice of integration direction produces more accurate volumes than the coordinate or other principle inertia directions.

Because airbag models may contain holes, e.g., holes for inflation and deflation, and Green's Theorem only applies to closed surfaces, a special treatment is needed for calculating the volume of airbags with holes. This special treatment consists of the following:

- The n-sized polygon defining the hole is identified automatically, using edge locating algorithms in LS-DYNA.
- The n-sized polygon is projected onto a plane, i.e., it is assumed to be flat; this is a good approximation for typical airbag hole geometries. Planar symmetry should work with the control volume capability for one symmetry plane.
- The area of the flat n-sided polygon is calculated using Green's Theorem in two dimensions.
- The resulting holes are processed as another surface element in the airbag control volume calculation.

30.2 Equation of State Model

As explained above, at each time step in the calculation the current volume of the airbag is determined from the control volume calculation. The pressure in the airbag corresponding to the control volume is determined from an equation of state (EOS) that relates the pressure to the current gas density (volume) and the specific internal energy of the gas.

The equation of state used for the airbag simulations is the usual 'Gamma Law Gas Equation of State',

$$p = (k-1)\rho e \tag{30.7}$$

where p is the pressure, k k is a constant defined below, ρ ρ is the density, and e is the specific internal energy of the gas. The derivation of this equation of state is obtained from

thermodynamic considerations of the adiabatic expansion of an ideal gas. The incremental change in internal energy, dU , in n moles of an ideal gas due to an incremental increase in temperature, dT , at constant volume is given by

$$dU = nc_v dT \tag{30.8}$$

where c_v is the specific heat at constant volume. Using the ideal gas law we can relate the change in temperature to a change in the pressure and total volume, v, as

$$d(pv) = nRdT \tag{30.9}$$

where R is the universal gas constant. Solving the above for dT and substituting the result into Equation (30.8) gives

$$dU = \frac{c_v d(pv)}{R} = \frac{d(pv)}{(k-1)} \tag{30.10}$$

where we have used the relationship

$$R = c_p - c_v \tag{30.11}$$

and the notation

$$k = \frac{c_p}{c_v} \tag{30.12}$$

Equation (30.10) may be rewritten as

$$dU = \frac{\rho_0 v_0}{k-1} d\left(\frac{p}{\rho}\right) \tag{30.13}$$

and integrated to yield

$$e = \frac{U}{\rho_0 v_0} = \frac{p}{\rho(k-1)} \tag{30.14}$$

Solving for the pressure

$$p = (k - 1)\ \rho e \tag{30.15}$$

The equation of state and the control volume calculation can only be used to determine the pressure when the specific internal energy is also known. The evolution equation for the internal energy is obtained by assuming the change in internal energy is given by

$$dU = - pdv \tag{30.16}$$

where the minus sign is introduced to emphasize that the volume increment is negative when the gas is being compressed. This expression can be written in terms of the specific internal energy as

$$de = \frac{dU}{\rho_0 v_0} = -\frac{pdv}{\rho_0 v} \tag{30.17}$$

Next, we divide the above by the equation of state, Equation (30.15), to obtain

$$\frac{de}{e} = -\frac{\rho(k-1)dv}{\rho_0 v_0} = -\frac{(k-1)dv}{v} \tag{30.18}$$

which may be integrated to yield

$$ln\ e = (1 - k)\ ln\ V \tag{30.19}$$

or evaluating at two states and exponentiating both sides yields

$$e_2 = e_1\left(\frac{v_2}{v_1}\right)^{(1-k)} \tag{30.20}$$

The specific internal energy evolution equation, Equation (30.20), the equation of state, Equation (30.15), and the control volume calculation completely define the pressure-volume relation for an inflated airbag.

30.3 Airbag Inflation Model

Airbag inflation models have been used for many years in occupant simulation codes such as CAL3D [Fleck, 1981].

The inflation model we chose to implement in LS-DYNA is due to Wang and Nefske[1988] and more recent improvements to the model in LS-DYNA were suggested by Wang [1992]. In their development they consider the mass flow due to the vents and leakage through the bag. We assume that the mass flow rate and the temperature of the gas going into the bag from an inflator are provided as tabulated functions of time.

A pressure relation is defined:

$$Q = \frac{p_e}{p_2} \tag{30.21}$$

where p_e is the external pressure and p_2 is the internal pressure in the bag. A critical pressure relationship is defined as:

$$Q_{crit} = \left(\frac{2}{k+1} \right)^{k/k-1} \tag{30.22}$$

where k is the ratio of specific heats:

$$k = \frac{c_p}{c_v}$$

If

$$Q \leq Q_{crit} \quad \text{then} \quad Q = Q_{crit} \tag{30.23}$$

Wang and Nefske define the mass flow through the vents and leakage by

$$\dot{m}_{23} = C_{23} A_{23} \frac{p_2}{R\sqrt{T_2}} Q^{1/k} \sqrt{2g_c \left(\frac{kR}{k-1} \right) \left(1 - Q^{k-1/k} \right)}$$

and

$$\dot{m}'_{23} = C'_{23} A'_{23} \frac{p_2}{R\sqrt{T_2}} Q^{1/k} \sqrt{2g_c \left(\frac{kR}{k-1} \right) \left(1 - Q^{k-1/k} \right)} \tag{30.24}$$

where $C_{23}, A_{23}, C'_{23}, A'_{23}, R$ and g_c are the vent orifice coefficient, vent orifice area, the orifice coefficient for leakage, the area for leakage, the gas constant, and the gravitational conversion constant, respectively. The internal temperature of the airbag gas is denoted by T_2. We note that both A_{23} and A'_{23} can be defined as a function of pressure [Wang, 1992] or if they are input as zero they are computed within LS-DYNA. This latter option requires detailed modeling of the airbag with all holes included.

A uniform temperature and pressure is assumed; therefore, in terms of the total airbag volume V_2 and air mass, m_2, the perfect gas law is applied:

$$p_2 V = m_2 R T_2 \tag{30.25}$$

Solving for T_2:

$$T_2 = \frac{p_2 V}{m_2 R} \tag{30.26}$$

and substituting Equation (30.26) into equations (30.24), we arrive at the mass transient equation:

$$\dot{m}_{out} = \dot{m}_{23} + \dot{m}'_{23} = \mu\sqrt{2p_2\rho}\sqrt{\frac{k\left(Q^{\frac{2}{k}} - Q^{k+1/k}\right)}{k-1}} \tag{30.27}$$

where

ρ = density of airbag gas
μ = bag characterization parameter
$\dot{m}_{out}$ = total mass flow rate out of bag

In terms of the constants used by Wang and Nefske:

$$\mu = \sqrt{g_c}\left(C_{23}A_{23} + C'_{23}A'_{23}\right) \tag{30.28}$$

We solved these equations iteratively, via function evaluation. Convergence usually occurs in 2 to 3 iterations.

The mass flow rate and gas temperature are defined in load curves as a function of time. Using the mass flow rate we can easily compute the increase in internal energy:

$$\dot{E}_{in} = c_p\dot{m}_{in}T_{in} \tag{30.29}$$

where T_{in} is the temperature of the gas flowing into the airbag. Initializing the variables pressure, p, density, ρ, and energy, E, to their values at time n, we can begin the iterations loop to compute the new pressure, p^{n+1}, at time $n+1$.

$$\begin{aligned} p^{n+1/2} &= \frac{p^n + p^{n+1}}{2} \\ \rho^{n+1/2} &= \frac{\rho^n + \rho^{n+1}}{2} \\ E^{n+1/2} &= \frac{E^n + E^{n+1}}{2} \\ Q^{n+1/2} &= \max\left(\frac{p_e}{p_2^{n+\frac{1}{2}}}, Q_{crit}\right) \end{aligned} \tag{30.30}$$

The mass flow rate out of the bag, $\dot{m}_{out}$ can now be computed:

$$\dot{m}_{out}^{n+1/2} = \mu\sqrt{2p_2^{n+1/2}\rho^{n+1/2}}\sqrt{\frac{k\left(Q^{n+1/2\,2/k} - Q^{n+1/2\,k+1/k}\right)}{k-1}} \tag{30.31}$$

where

$$p_2^{n+1/2} = p^{n+1/2} + p_e \tag{30.32}$$

and the total mass updated:

$$\begin{aligned} m^{n+1} &= m^n + \Delta t\left(\dot{m}_{in}^{n+1/2} - m_{out}^{n+1/2}\right) \\ m^{n+1/2} &= \frac{m^n + m^{n+1}}{2} \end{aligned} \tag{30.33}$$

The energy exiting the airbag is given by:

$$\dot{E}_{out}^{n+1/2} = \dot{m}_{out}^{n+1/2}\frac{E^{n+1/2}}{m^{n+1/2}} \tag{30.34}$$

we can now compute our new energy at time n+1

$$E^{n+1} = E^n + \Delta t\left(\dot{E}_{in}^{n+1/2} - \dot{E}_{out}^{n+1/2}\right) - p^{n+1/2}\Delta V^{n+1/2} \tag{30.35}$$

where $\Delta V^{n+1/2}$ is the change in volume from time n to n+1. The new pressure can now be computed:

$$p^{n+1} = (k-1)\frac{E^{n+1}}{V^{n+1}} \tag{30.36}$$

which is the gamma-law (where $k = \gamma$) gas equation. This ends the iteration loop.

30.4 Wang's Hybrid Inflation Model

Wang's proposed hybrid inflator model [1995a, 1995b] provides the basis for the model in LS-DYNA. The first law of thermodynamics is used for an energy balance on the airbag control volume.

$$\frac{d}{dt}(mu)_{cv} = \sum \dot{m}_i h_i - \sum \dot{m}_o h_o - \dot{W}_{cv} - \dot{Q}_{cv} \tag{30.37}$$

where

$\frac{d}{dt}(mu)_{cv}$ = rate of change of airbag internal energy

$\sum m_i h_i$ = energy into airbag by mass flow (e.g., inflator)

$\sum m_o h_o$ = energy out of airbag by mass flow (e.g., vents)

$\dot{W}_{cv} = \int P d\dot{V}$ = work done by airbag expansion

$\dot{Q}_{cv}$ = energy out by heat transfer through airbag surface

The rate of change of internal energy, the left hand side of Equation (30.37), can be differentiated:

$$\frac{d}{dt}(mu) = \frac{dm}{dt}u + m\frac{du}{dt} = \frac{dm}{dt}u + m\frac{d}{dt}(c_v T) = \dot{m}u + m\dot{c}_v T + mc_v \frac{dT}{dt} \qquad (30.38)$$

where we have used the definition

$$u = c_v T \qquad (30.39)$$

Then, the energy equation can be re-written for the rate of change in temperature for the airbag

$$\frac{dT_{cv}}{dt} = \frac{\sum \dot{m}_i h_i - \sum \dot{m}_o h_o - \dot{W}_{cv} - \dot{Q}_{cv} - (\dot{m}u)_{cv} - (m\dot{c}_v T)_{cv}}{(mc_v)_{cv}} \qquad (30.40)$$

Temperature dependent heat capacities are used. The constant pressure molar heat capacity is taken as:

$$\bar{c}_p = \bar{a} + \bar{b}T \qquad (30.41)$$

and the constant volume molar heat capacity as:

$$\bar{c}_v = \bar{a} + \bar{b}T - \bar{r} \qquad (30.42)$$

where

$\bar{r}$ = gas constant = 8.314 J/gm-mole K
$\bar{a}$ = constant [J/gm-mole K]
$\bar{b}$ = constant [J/gm-mole K^2]

Mass based values are obtained by dividing the molar quantities by the molecular weight, M, of the gas

$$a = \frac{\bar{a}}{M} \qquad b = \frac{\bar{b}}{M} \qquad r = \frac{\bar{r}}{M} \qquad (30.43)$$

The constant pressure and volume specific heats are then given by

$$c_p = a + bT \qquad (30.44a)$$

$$c_v = a + bT - r \qquad (30.44b)$$

The specific enthalpy and internal energy becomes:

$$h = \int_0^T c_p dT = aT + \frac{bT^2}{2} \tag{30.45a}$$

$$u = \int_0^T c_v dT = aT + \frac{bT^2}{2} - rT \tag{30.45b}$$

For ideal gas mixtures the molecular weight is given as:

$$M = \frac{1}{\sum \frac{f_i}{M_i}} \tag{30.46}$$

and the constant pressure and volume specific heats as:

$$c_p = \sum f_i c_{p(i)} \tag{30.47a}$$

$$c_v = \sum f_i c_{v(i)} \tag{30.47b}$$

where

f_i = mass fraction of gas i

M_i = molecular weight of gas i

$c_{p(i)}$ = constant pressure specific heat of gas i

$c_{v(i)}$ = constant volume specific heat of gas i

The specific enthalpy and internal energy for an ideal gas mixture with temperature dependent heat capacity are

$$h = \int_0^T \sum f_i c_{p(i)} dT = \sum f_i \left(a_i T + \frac{b_i T^2}{2} \right) \tag{30.48}$$

$$u = \int_0^T \sum f_i c_{v(i)} dT = \sum f_i \left(a_i T + \frac{b_i T^2}{2} - r_i T \right) \tag{30.49}$$

The rate of change of temperature for the airbag is

$$\frac{dT_{cv}}{dt} = \frac{\sum \dot{m}_i h_i - \sum \dot{m}_o h_o - \dot{W}_{cv} - \dot{Q}_{cv} - (\dot{m}u)_{cv} - (m\dot{c}_v T)_{cv}}{(mc_v)_{cv}} \tag{30.50}$$

The energy in by mass flow becomes:

$$\sum \dot{m}_i h_i = \sum \dot{m}_i \left(a_i T_i + \frac{b_i T_i^2}{2} \right) \tag{30.51}$$

$\dot{m}_i$ is specified by an inflator mass inflow vs. time table
T_i is specified by an inflator temperature vs. time table
a, b are input constants for gas i

and the energy out by mass flow:

$$\sum \dot{m}_o h_o = \sum \dot{m}_o \left[\sum_{gases} f_i \left(a_i T_{cv} + \frac{b_i T_{cv}^2}{2} \right) \right] \tag{30.52}$$

The gas leaves the airbag at the control volume temperature T_{cv}. The mass flow rate out through vents and fabric leakage is calculated by the one dimensional isentropic flow equations per Wang and Nefske. The work done by the airbag expansion is given by:

$$\dot{W}_{cv} = \int P d\dot{V} \tag{30.53}$$

P is calculated by the equation of state for a perfect gas, $p = \rho RT$ and $\dot{V}$ is calculated by LS-DYNA

For the energy balance, we must compute the energy terms $(\dot{m}u)_{cv}$ and $(mc_v)_{cv}$. Conservation of mass leads to:

$$\begin{aligned} \dot{m}_{cv} &= \dot{m}_i - \dot{m}_o \\ m_{cv} &= \int \dot{m}_{cv} dt \end{aligned} \tag{30.54}$$

The internal energy is given by

$$u_{cv} = \sum f_i \left(a_i T_{cv} + \frac{b_i T_{cv}^2}{2} - r_i T_{cv} \right) \tag{30.55}$$

and the heat capacity at contact volume is:

$$(c_v)_{cv} = \sum f_i \left(a_i + b_i T_{cv} - r_i \right) \tag{30.56}$$

30.5 Constant Volume Tank Test

Constant volume tank tests are used to characterize inflators. The inflator is ignited within the tank and, as the propellant burns, gas is generated. The inflator temperature is assumed to be constant. From experimental measurements of the time history of the tank pressure it is straightforward to derive the mass flow rate, $\dot{m}$.

From energy conservation, where T_i and T_t are defined to be the temperature of the inflator and tank, respectively, we obtain:

$$c_p \dot{m} T_i = c_v \dot{m} T_t + c_v m \dot{T}_t \tag{30.57}$$

For a perfect gas under constant volume, $\dot{V} = 0$, hence,

$$\dot{p} V = \dot{m} R T_t + m R \dot{T}_t \tag{30.58}$$

and, finally, we obtain the desired mass flow rate:

$$\dot{m} = \frac{c_v \dot{p} V}{c_p R T_i} \tag{30.59}$$

31. DYNAMIC RELAXATION AND SYSTEM DAMPING

Dynamic relaxation allows LS-DYNA to approximate solutions to linear and nonlinear static or quasi-static processes. Control parameters must be selected with extreme care or bad results can be obtained. The current methods are not compatible with displacement or velocity boundary conditions, but various body loads, thermal loads, pressures, and nodal loads are allowed. The solutions to most nonlinear problems are path dependent, thus results obtained in the presence of dynamic oscillations may not be the same as for a nonlinear implicit code, and they may diverge from reality.

In LS-DYNA we have two methods of damping the solution. The first named "dynamic relaxation" is used in the beginning of the solution phase to obtain the initial stress and displacement field prior to beginning the analysis. The second is system damping which can be applied anytime during the solution phase either globally or on a material basis.

31.1 Dynamic Relaxation For Initialization

In this phase only a subset of the load curves is used to apply the static load which is flagged in the load curve section of the manual. The calculation begins and executes like a normal LS-DYNA calculation but with damping incorporated in the update of the displacement field.

Our development follows the work of Underwood [1986] and Papadrakakis [1981] with the starting point being the dynamic equilibrium equation, Equation (23.1) with the addition of a damping term, at time n:

$$\begin{aligned} Ma^n + Cv^n + Q^n(d) &= 0 \\ Q^n(d) &= F^n - P^n - H^n \end{aligned} \tag{31.1}$$

where we recall that M is the mass matrix, C is the damping matrix, n indicates the nth time step, a^n is the acceleration, v^n the velocity, and d is the displacement vector. With Δt as the fixed time increment we get for the central difference scheme:

$$v^{n+1/2} = \frac{\left(d^{n+1} - d^n\right)}{\Delta t} \quad ; \quad a^n = \frac{\left(v^{n+1/2} - v^{n-1/2}\right)}{\Delta t} \tag{31.2}$$

For v^n we can assume an averaged value

$$v^n = \frac{1}{2}\left(v^{n+1/2} + v^{n-1/2}\right) \tag{31.3}$$

and obtain

$$v^{n+1/2} = \left(\frac{1}{\Delta t}M + \frac{1}{2}C\right)^{-1}\left[\left(\frac{1}{\Delta t}M - \frac{1}{2}C\right)v^{n-1/2} - Q^n\right] \tag{31.4}$$

$$d^{n+1} = d^n + \Delta t v^{n+½} \tag{31.5}$$

In order to preserve the explicit form of the central difference integrator, M and C must be diagonal. For the dynamic relaxation scheme C has the form

$$C = c \cdot M \tag{31.6}$$

If Equation (31.6) is substituted into (31.4) the following form is achieved

$$v^{n+½} = \frac{2 - c \cdot \Delta t}{2 + c \cdot \Delta t} v^{n-½} + \frac{2\Delta t}{2 + c\Delta t} \cdot M^{-1} \cdot Q^n \tag{31.7}$$

Since M is diagonal, each solution vector component may be computed individually from

$$v_i^{n+½} = \frac{2 - c\Delta t}{2 + c\Delta t} v_i^{n-½} + \frac{2\Delta t}{2 + c\Delta t} \frac{Q_i^n}{m_i} \tag{31.8}$$

As a starting procedure it is suggested by Underwood

$$\begin{aligned} v^0 &= 0 \\ d^0 &= 0 \end{aligned} \tag{31.9}$$

Since the average value is used for v^n, which must be zero at the beginning for a quasi-static solution

$$v^{-½} = -v^{½} \tag{31.10}$$

thus the velocity at time $+½$ is

$$v^{½} = -\frac{\Delta t}{2} M^{-1} Q^o \tag{31.11}$$

A damping coefficient must now be selected to obtain convergence to the static solution in minimal time. The best estimate for damping values is based on the frequencies of the structure. One choice is to focus on an optimal damping parameter as suggested by Papadrakakis [1981]. Then dynamic relaxation is nothing else but a critically damped dynamic system

$$C = C_{cr} = 2\omega_{min} m \tag{31.12}$$

with m as modal mass. The problem is finding the dominant eigenvalue in the structure related to the "pseudo-dynamic" behavior of the structure. As the exact estimate would be rather costly and not fit into the explicit algorithm, an estimate must be used. Papadrakakis suggests

$$\lambda_D = \frac{\left\| d^{n+1} - d^n \right\|}{\left\| d^n - d^{n-1} \right\|} \tag{31.13}$$

When this quantity has converged to an almost constant value, the minimum eigenvalue of the structure can be estimated:

$$\omega_{min}^2 = -\frac{\left(\lambda_D^2 - \lambda_D \cdot \beta + \alpha\right)}{\lambda_D \cdot \gamma} \tag{31.14}$$

where

$$\alpha = \frac{2 - c\Delta t}{2 + c\Delta t}$$
$$\beta = \alpha + 1$$
$$\gamma = \frac{2\Delta t^2}{2 + c\Delta t}$$

The maximum eigenvalue determines the time step and is already known from the model

$$\omega_{\max}^2 = \frac{4.0}{\left(\Delta t\right)^2} \tag{31.15}$$

Now the automatic adjustment of the damping parameter closely follows the paper of Papadrakakis, checking the current convergence rate compared to the optimal convergence rate. If the ratio is reasonably close, then an update of the iteration parameters is performed.

$$c = \frac{4.0}{\Delta t} \frac{\sqrt{\omega_{\min}^2 \cdot \omega_{\max}^2}}{\left(\omega_{\min}^2 + \omega_{\max}^2\right)} \tag{31.16}$$

As is clearly visible from Equation (31.16) the value of highest frequency has always a rather high influence on the damping ratio. This results in a non-optimal damping ratio, if the solution is dominated by the response in a very low frequency compared to the highest frequency of the structure. This is typically the case in shell structures, when bending dominates the solution. It was our observation that the automatic choice following Papadrakakis results in very slow convergence for such structures, and this is also mentioned by Underwood for similar problems. The damping ratio should then be fully adjusted to the lowest frequency by hand by simply choosing a rather high damping ratio. An automatic adjustment for such cases is under preparation. For structures with dominant frequencies rather close to the highest frequency, convergence is really improved with the automatically adjusted parameter.

If the automated approach is not used then we apply the damping as

$$v^{n+\frac{1}{2}} = \eta v^{n-\frac{1}{2}} + a^n \Delta t \tag{31.17}$$

where η is an input damping factor (defaulted to .995). The factor, η, is equivalent to the corresponding factor in Equations (31.7- 31.8).

The relaxation process continues until a convergence criterion based on the global kinetic energy is met, i.e., convergence is assumed if

$$E_{ke} < cvtol \cdot E_{ke}^{\max} \tag{31.18}$$

where *cvtol* is the convergence tolerance (defaulted to .001). The kinetic energy excludes any rigid body component. Initial velocities assigned in the input are stored during the relaxation. Once convergence is attained the velocity field is initialized to the input values. A termination time for the dynamic relaxation phase may be included in the input and is recommended since if convergence fails, LS-DYNA will continue to execute indefinitely.

31.2 Mass Weighted Damping

With mass weighted damping, the Equation (23.2) is modified as:

$$a^n = M^{-1}\left(P^n - F^n + H^n - F_{damp}^n \right) \tag{31.19}$$

where

$$F_{damp}^n = D_s m v \tag{31.20}$$

As seen from Figure 31.1 and as discussed above the best damping constant for the system is usually the critical damping constant: Therefore,

$$D_s = 2\omega_{min} \tag{31.21}$$

is recommended.

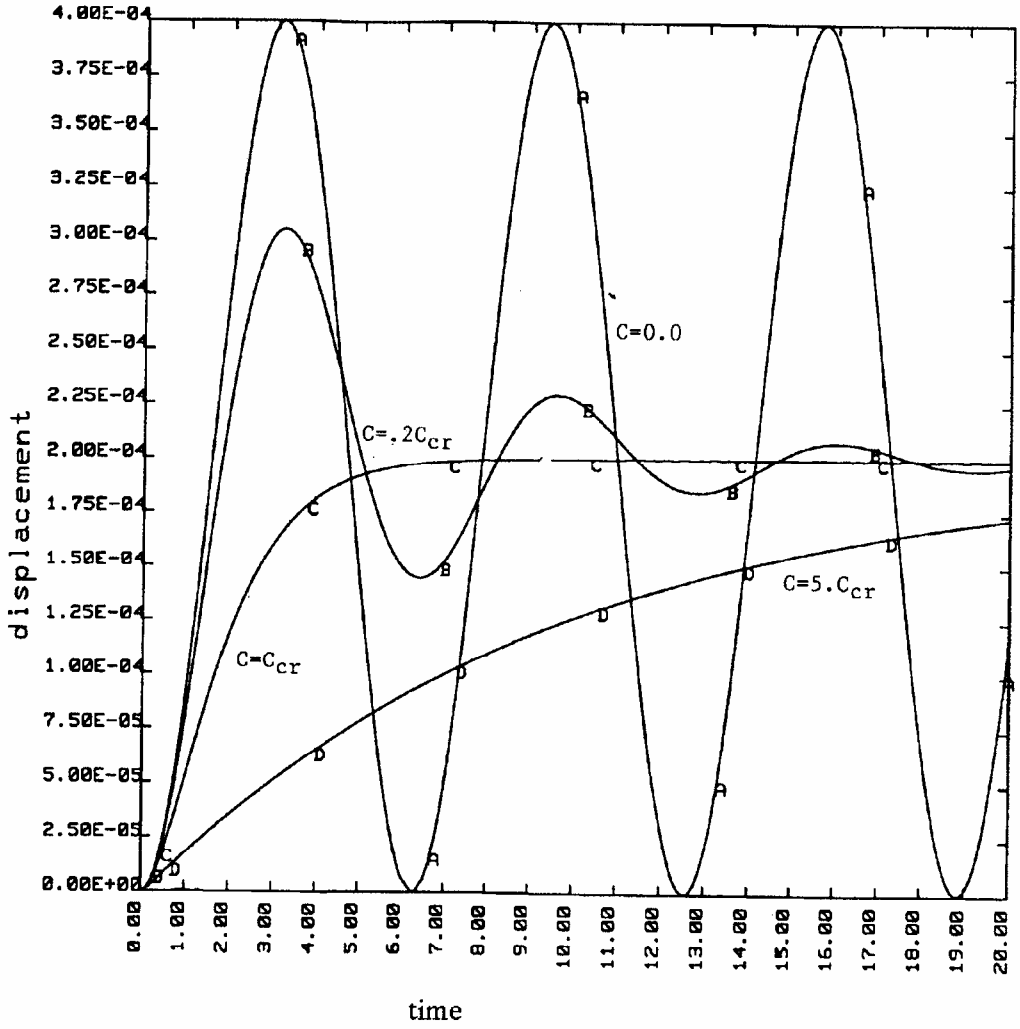


Figure 31.1. Displacement versus time curves with a variety of damping coefficients applied to a one degree-of-freedom oscillator.

31.3 Dynamic Relaxation—How Fast Does it Converge?

The number of cycles required to reduce the amplitude of the dynamic response by a factor of 10 can be approximated by [see Stone, Krieg, and Beisinger 1985]

$$ncycle = 1.15\frac{\omega_{\max}}{\omega_{\min}} \tag{31.22}$$

Structural problems which involve shell and beam elements can have a very large ratio and consequently very slow convergence.

32. HEAT TRANSFER

LS-DYNA can be used to solve for the steady state or transient temperature field on three-dimensional geometries. Material properties may be temperature dependent and either isotropic or orthotropic. A variety of time and temperature dependent boundary conditions can be specified including temperature, flux, convection, and radiation. The implementation of heat conduction into LS-DYNA is based on the work of Shapiro [1985].

32.1 Conduction of Heat in an Orthotropic Solid

The differential equations of conduction of heat in a three-dimensional continuum is given by

$$\rho c_p \frac{\partial \theta}{\partial t} = \left(k_{ij} \theta_{,j} \right)_{,i} + Q , \tag{32.1}$$

subject to the boundary conditions

$$\begin{aligned} \theta &= \theta_s \qquad \text{on } \Gamma_1 \\ k_{ij}\theta_{,j} n_i + \beta\theta &= \gamma \quad \text{on } \Gamma_2 \end{aligned} \tag{32.2}$$

and initial conditions at t_0:

$$\theta_\Gamma = \theta_0(x_i) \text{ **at** } t{=}t_0 . \tag{32.3}$$

where

$\theta = \theta(x_i, t)$	=	temperature
$x_i = x_i(t)$	=	coordinates as a function of time
$\rho = \rho(x_i)$	=	density
$c_p = c_p(x_i, \theta)$	=	specific heat
$k_{ij} = k_{ij}(x_i, \theta)$	=	thermal conductivity
$Q = Q(x_i, \theta)$	=	internal heat generation rate per unit volume Ω
θ_Γ	=	prescribed temperature on Γ_1
n_i	=	normal vector to Γ_2

Equations (32.1-32.3) represent the strong form of a boundary value problem to be solved for the temperature field within the solid.

DYNA3D employs essentially the same theory as TOPAZ [Shapiro 1985] in solving Equation (32.1) by the finite element method. Those interested in a more detailed description of the theory are referred to the TOPAZ User's Manual. Brick elements are integrated with a $2\times2\times2$ Gauss quadrature rule, with temperature dependence of the properties accounted for at the Gauss points. Time integration is performed using a generalized trapezoidal method shown by Hughes to be unconditionally stable for nonlinear problems. Newton's method is used to satisfy equilibrium in nonlinear problems.

The finite element method provides the following equations for the numerical solution of Equations (32.1-32.3).

$$\left[\frac{C_{n+\alpha}}{\Delta t}+\alpha H_{n+\alpha}\right]\{\theta_{n+1}-\theta_n\}=\{F_{n+\alpha}-H_{n+\alpha}\theta_n\} \tag{32.4}$$

where

$$[C]=\sum_e\left[C_{ij}^e\right]=\sum_e\int_{\Omega^e}N_i\ \rho c\ N_j d\Omega \tag{32.5}$$

$$[H]=\sum_e\left[H_{ij}^e\right]=\sum_e\left[\int_{\Omega^e}\nabla^T N_i K\nabla N_j d\Omega+\int_{\Gamma^e}N_i\beta N_j d\Gamma\right] \tag{32.6}$$

$$[F]=\sum_e\left[F_i^e\right]=\sum_e\left[\int_{\Omega^e}N_i q_g d\Omega+\int_{\Gamma^e}N_i\gamma d\Gamma\right] \tag{32.7}$$

The parameter α is taken to be in the interval [0,1]. Some well-known members of this α-family are

α	Method
0	forward difference; forward Euler
½	midpoint rule; Crank-Nicolson
⅔	Galerkin
1	backward difference, fully implicit

32.2 Thermal Boundary Conditions

Boundary conditions are represented by

$$k_x\frac{\partial\theta}{\partial x}n_x+k_y\frac{\partial\theta}{\partial y}n_y+k_z\frac{\partial\theta}{\partial z}n_z=\gamma\ -\beta\theta=\dot{q}'' \tag{32.8}$$

By convention, heat flow is positive in the direction of the surface outward normal vector. Surface definition is in accordance with the right hand rule. The outward normal vector points to the right as one progresses from node N_1 to N_2 to N_3 and finally to N_4. See Figure 32.1.

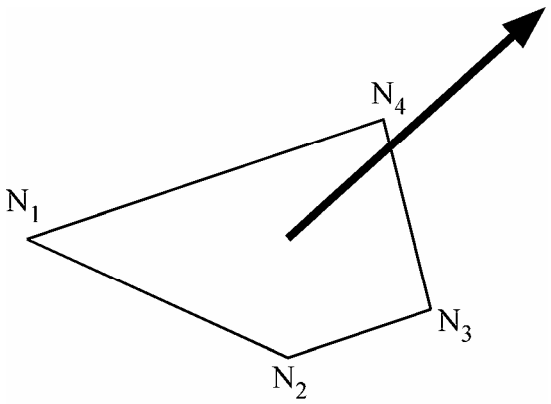


Figure 32.1.

Boundary conditions can be functions of temperature or time. More than one boundary condition can be specified over the same surface such as in a case of combined convection and radiation. For situations where it is desired to specify adiabatic (i.e., $\dot{q}'' = 0$) conditions, such as at an insulated surface or on a line of symmetry, no boundary condition need be specified. This is the default boundary condition in LS-DYNA.

Temperature boundary condition can be specified on any node whether on the physical boundary or not.

Flux, convection, and radiation boundary conditions are specified on element surface segments defined by 3 (triangular surface) or 4 nodes (quadrilateral surface). These boundary conditions can be specified on any finite element surface whether on the physical boundary or not.

1) Flux: Set $\dot{q}'' = q_f$ where q_f is defined at the node points comprising the flux boundary condition surface.

2) Convection: A convection boundary condition is calculated using $\dot{q}'' = h(T - T_\infty)$

where

h heat transfer coefficient.
$(T\text{-}T_\infty)$ temperature potential.

LS-DYNA evaluates h at the film temperature

$$T = \frac{1}{2}\left(T_{surf} + T_\infty\right) . \qquad (32.9)$$

3) Radiation: A radiation boundary condition is calculated using $\dot{q}'' = h_r(T - T_\infty)$, where $h_r = \sigma\varepsilon F$ is a radiant-heat-transfer coefficient.

32.3 Thermal Energy Balances

Various energy terms are printed and written into the plot file for post processing using the code LS-PREPOST. The energy terms are:

- change in material internal energy for time step,
- change in material internal energy from initial time,
- heat transfer rates on boundary condition surfaces,
- heat transfer rates on enclosure radiation surfaces,
- x, y, and z fluxes at all nodes.

32.4 Heat Generation

Volumetric heat generation rates may be specified by element, by material, or both (in which case the effect is additive). Volumetric heat generation rates can be a function of time or temperature.

32.5 Initial Conditions

Initial temperature conditions can be specified on the nodal data input cards or on the nodal temperature initial condition cards. If no temperatures are specified, the default is 0. For nonlinear steady state problems the temperature initial condition serves as a first guess for the equilibrium iterations.

32.6 Material Properties

Heat capacity and thermal conductivity may be functions of temperature. Since the density and heat capacity appear only as a product in the governing equations, the temperature dependence of the density may be included in the temperature dependence of the heat capacity. Material properties are evaluated at the element Gauss point temperature or average element temperature.

The thermal conductivity may be either isotropic or orthotropic. For an orthotropic material, the three material axes (x_1', x_2', x_3') are orthogonal and the thermal conductivity tensor K' is diagonal.

The thermal conductivity tensor K in the global coordinate system is related by

$$K_{ij} = K'_{ij}\beta_{mi}\beta_{nj}$$

where

$$\beta_{ij} = \cos\left(x_i', x_j\right)$$

32.7 Nonlinear Analysis

In a nonlinear problem, C, H, and F are functions of temperature. Newton's method is used to transform equation 32.4 into an alternate form which contains temperature derivatives of C, H, and F (i.e., the tangent matrix). Iterations are required to solve this alternate form.

In a steady state nonlinear problem, an initial guess should be made of the final temperature distribution and included in the input file as an initial condition. If your guess is good, a considerable savings in computation time is achieved.

32.8 Units

Any consistent set of units with the governing equation may be used. Examples are:

Quantity	Units		
temperature	K	C	F
space	m	cm	ft
time	s	s	hr
density	kg/m^3	g/cm^3	Lb_m/ft^3
heat capacity	J/kg k	cal/g c	Btu/Lb_mF
thermal conductivity	W/m K	cal/s cm C	Btu/hr ft F
thermal generation	W/M^3	cal/s cm^3	Btu/hr ft^3
heat flux	W/m^2	cal/s cm^2	Btu/hr ft^2

33. ADAPTIVITY

LS-DYNA includes an h-adaptive method for the shell elements. In an h-adaptive method, the elements are subdivided into smaller elements wherever an error indicator shows that subdivision of the elements will provide improved accuracy. An example of an adaptive calculation on a thin wall square cross section beam is shown in Figure 33.1. In Figures 33.2 through 33.4 a simple metal stamping simulation is shown [also see Galbraith, Finn, et. al., 1991]. In the following, the methodologies used in the h-adaptive method in LS-DYNA are described. The objective of the adaptive process used in LS-DYNA is to obtain the greatest accuracy for a given set of computational resources. The user sets the initial mesh and the maximum level of adaptivity, and the program subdivides those elements in which the error indicator is the largest. Although this does not provide control on the error of the solution, it makes it possible to obtain a solution of comparable accuracy with fewer elements, and, hence, less computational resources, than with a fixed mesh.

LS-DYNA uses an h-adaptive process, where parts of the mesh are selectively refined during the course of the solution procedure. The methodology used is based on Belytschko, Wong, and Plaskacz [1989]. In the former, elements were also fused or combined when it was felt that they were no longer needed. It was found that the implementation of fusing procedures for general meshes, such as occur in typical applications of commercial programs, is too complex, so only fission is included. Adaptivity in LS-DYNA can be restricted to specific groups of shell elements. Elements that fall in this group are said to be in the active adaptivity domain.

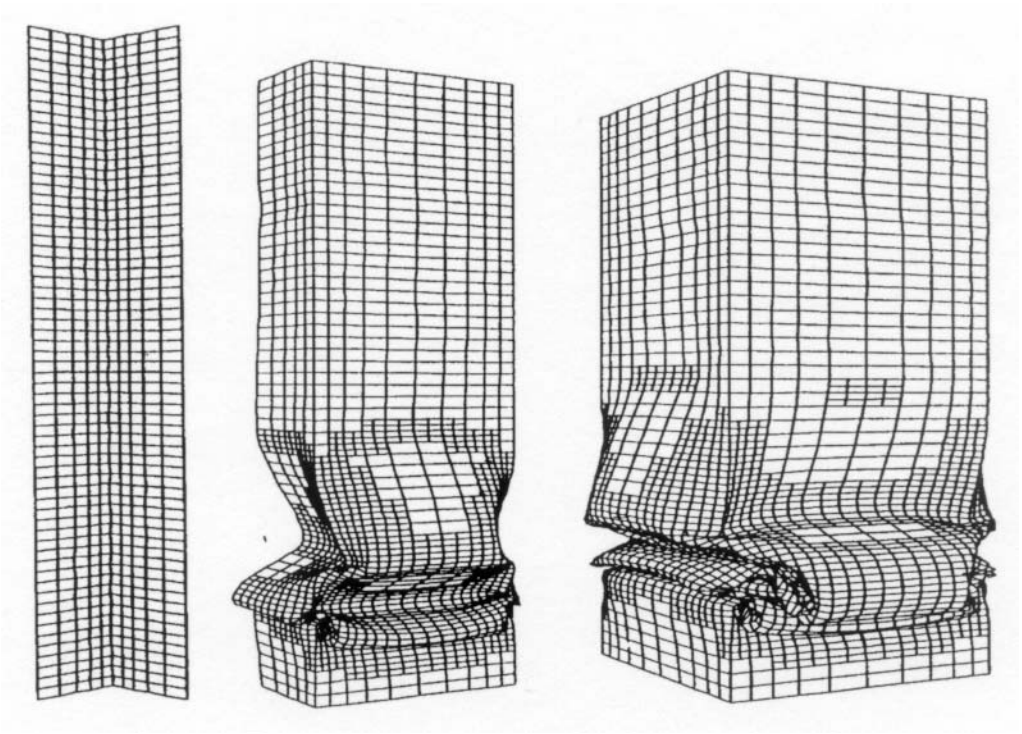


Figure 33.1. One level adaptive calculation on a square cross section beam.

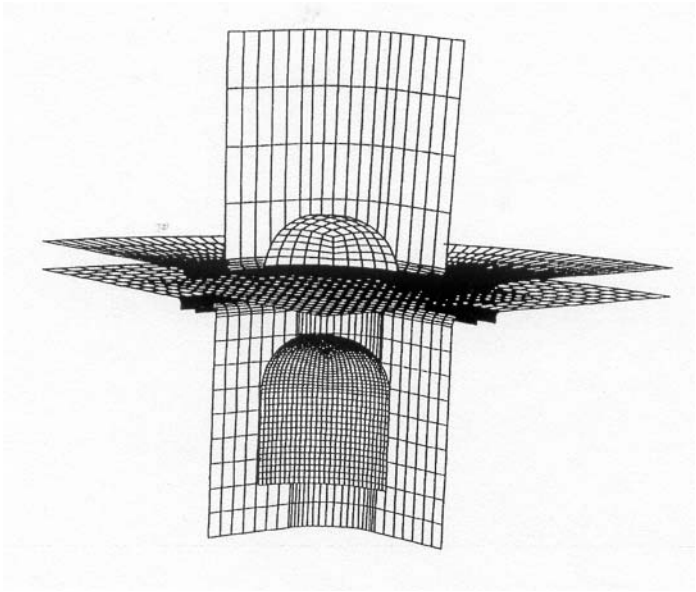


Figure 33.2. Aluminum blank with 400 shells in blank and four rigid tools.

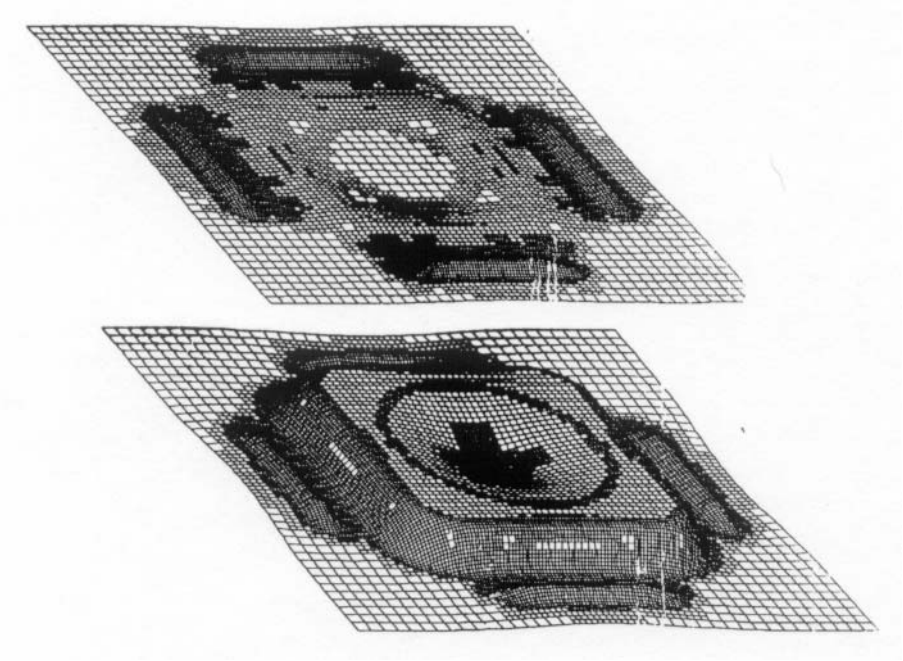


Figure 33.3. Adaptive calculations using two adaptive levels.

In the h-adaptive process, elements are subdivided into smaller elements where more accuracy is needed; this process is called fission. The elements involved in the fission process are subdivided into elements with sides h/2, where h is the characteristic size of the original elements. This is illustrated in Figure 33.5 for a quadrilateral element. In fission, each quadrilateral is subdivided into four quadrilaterals (as indicated in Figure 33.2) by using the mid-points of the sides and the centroid of the element to generate four new quadrilaterals.

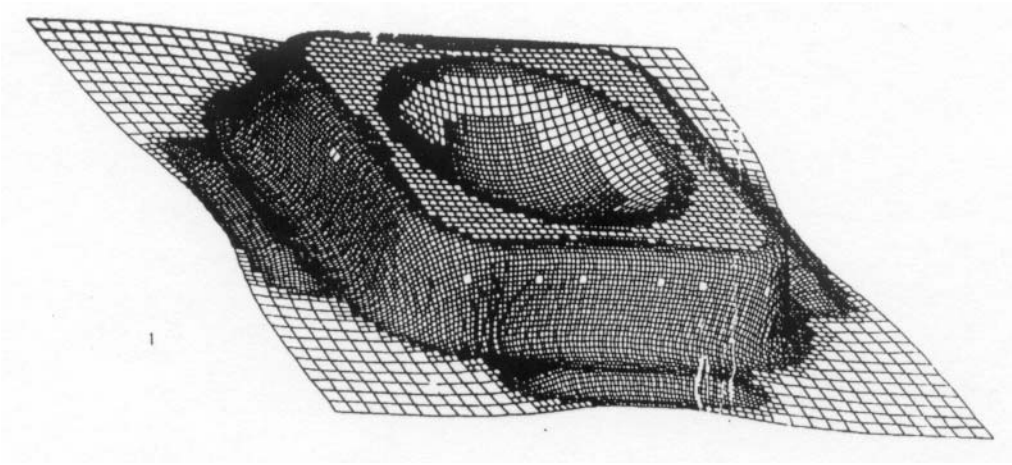


Figure 33.4. Final shape of formed part with 4315 shell elements per quarter.

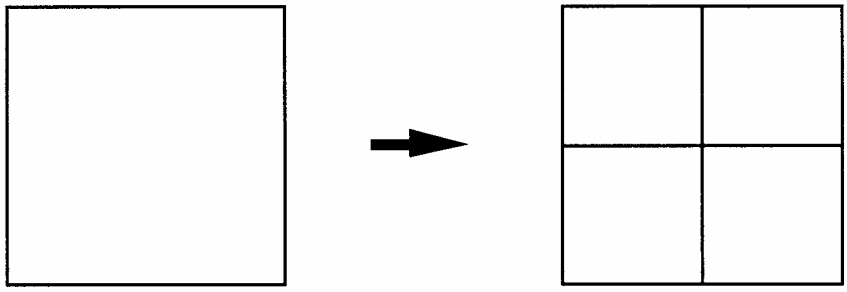


Figure 33.5. Fissioning of a Quadrilateral Element

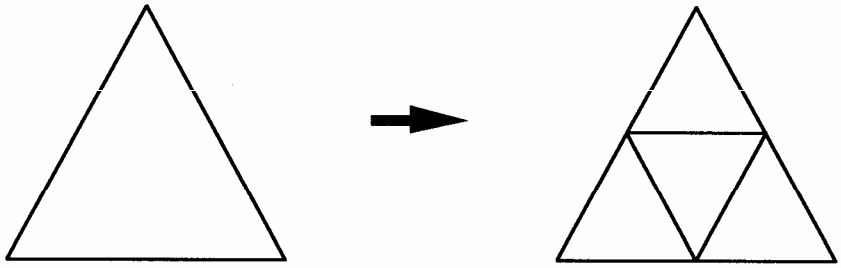


Figure 33.6. Fissioning of a Triangular Element

The fission process for a triangular element is shown in Figure 33.6 where the element is subdivided into four triangles by using the mid-points of the three sides. The adaptive process can consist of several levels of fission. Figure 33.5 shows one subdivision, which is called the second refinement level. In subsequent steps, the fissioned elements can again be fissioned in a third refinement level, and these elements can again, in turn, be fissioned in a fourth level, as shown in Figure 33.7. The levels of adaptivity that occur in a mesh are restricted by three rules:

1. The number of levels is restricted by the maximum level of adaptivity that is allowed in the mesh, which is generally set at 3 or 4. At the fourth level up to 64 elements will be generated for each element in the initial mesh.
2. The levels of adaptivity implemented in a mesh must be such that the levels of adaptivity implemented in adjacent elements differ by, at most, one level.
3. The total number of elements can be restricted by available memory. Once the specified memory usage is reached, adaptivity ceases.

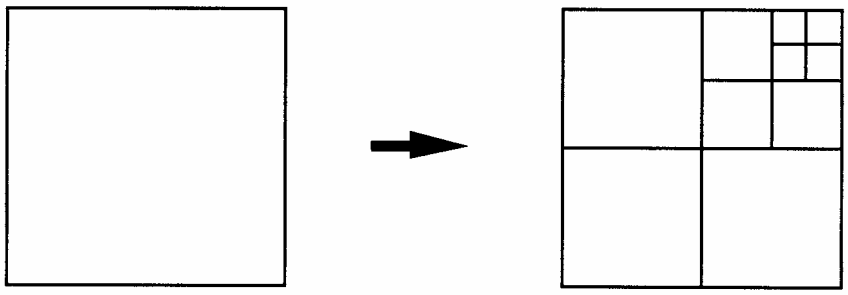


Figure 33.7. Quadrilateral Element Fissioned to the fourth level

The second rule is used to enforce a 2-to-1 rule given by Oden, Devloo and Strouboulis [1986], which restricts the number of elements along the side of any element in the mesh to two. The enforcement of this rule is necessary to accommodate limitations in the data structure.

The original mesh provided by the user is known as the parent mesh, the elements of this mesh are called the parent elements, and the nodes are called parent nodes. Any elements that are generated by the adaptive process are called descendant elements, and any nodes that are generated by the adaptive process are called descendant nodes. Elements generated by the second level of adaptivity are called first-generation elements, those generated by third level of adaptivity are called second-generation elements, etc.

The coordinates of the descendant nodes are generated by using linear interpolation. Thus, the coordinates of any node generated during fission of an element are given by

$$x_N = \frac{1}{2}\left(x_I + x_J\right) \tag{33.1}$$

where x_N is the position of the generated node and x_I and x_J are the nodes along the side on which x_N was generated for a typical element as shown in Fig. 33.1. The coordinate of the mid-point node, which is generated by fission of a quadrilateral element, is given by

$$x_M = \frac{1}{4}\left(x_I + x_J + x_K + x_L\right) \tag{33.2}$$

where x_M is the new midpoint node of the fissioned quadrilateral and x_I, x_J, x_K and x_L are the nodes of the original quadrilateral. The velocities of the nodes are also given by linear interpolation. The velocities of edge nodes are given by

$$v_N = \frac{1}{2}(v_I + v_J) \tag{33.3}$$

and the angular velocities are given by

$$\omega_N = \frac{1}{2}(\omega_I + \omega_J) \tag{33.4}$$

The velocities of a mid-point node of a fissioned quadrilateral element are given by

$$v_M = \frac{1}{4}(v_I + v_J + v_K + v_L) \tag{33.5}$$

$$\omega_M = \frac{1}{4}(\omega_I + \omega_J + \omega_K + \omega_L) \tag{33.6}$$

The stresses in the descendant element are obtained from the parent element by setting the stresses in the descendant elements equal to the stresses in the parent element at the corresponding through-the-thickness quadrature points.

In subsequent steps, nodes which are not corner nodes of an all attached elements are treated as slave nodes. They are handled by the simple constraint equation.

Refinement indicators are used to decide the locations of mesh refinement. One deformation based approach checks for a change in angles between contiguous elements as shown in Figure 33.8. If $\zeta > \zeta_{tol}$ then refinement is indicated, where ζ_{tol} is user defined.

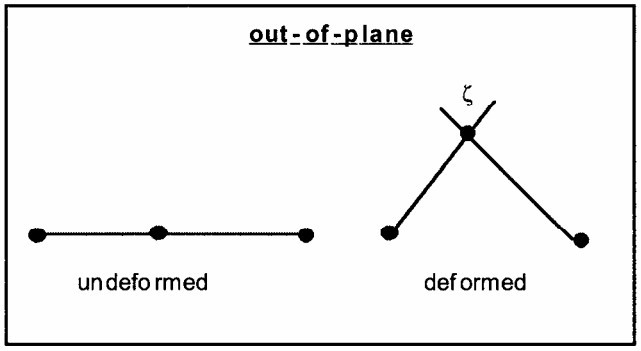


Figure 33.8. Refinement indicator based on angle change.

After the mesh refinement is determined, we can refine the mesh and continue the calculation or back up to an earlier time and repeat part of the calculation with the new mesh. For accuracy and stability reasons the latter method is generally preferred; however, the former method is preferred for speed. Whether LS-DYNA backs up and repeats the calculation or continues after remeshing is determined by an input parameter, ADPASS. This is illustrated in Figure 33.9.

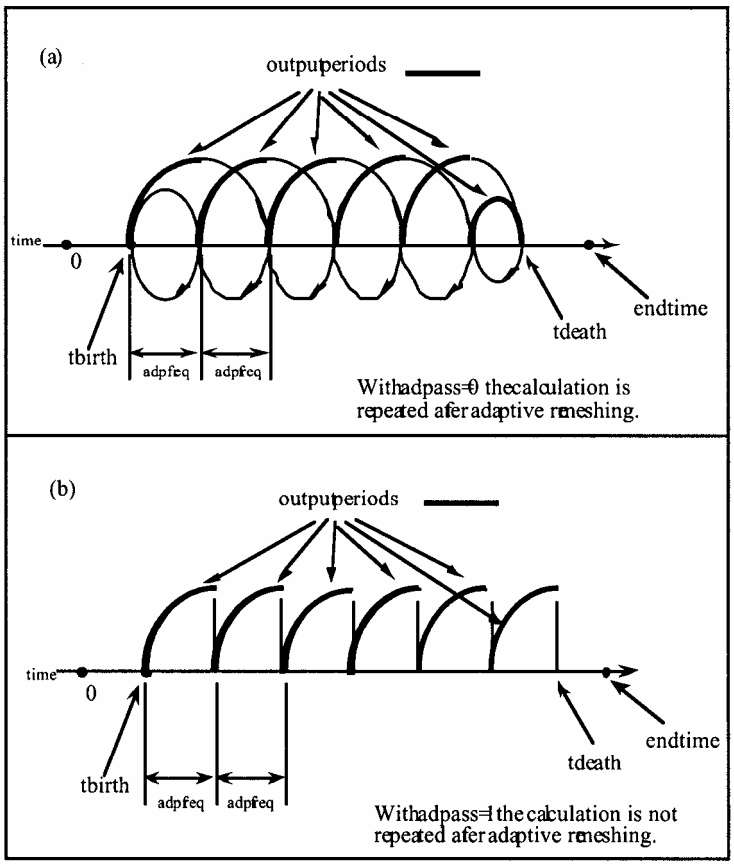


Figure 33.9. The input parameter, ADPASS, controls whether LS-DYNA backs up and repeats the calculation after adaptive refinement.

34. IMPLICIT

Implicit solvers are properly applied to static, quasi-static, and dynamic problems with a low frequency content. Such applications include but are not limited to

- Static and quasi-static structural design and analysis
- Metal forming, especially, the binderwrap and springback

An advantage of the implicit solver on explicit integration is that the number of load or time steps are typically 100 to 10000 times fewer. The major disadvantage is that the cost per step is unknown since the speed depends mostly on the convergence behavior of the equilibrium iterations which can vary widely from problem to problem.

An incremental-iterative numerical algorithm is implemented in LS-DYNA. The method is stable for wide range of nonlinear problems that involve finite strain and arbitrarily large rotations. Accuracy consideration usually limits the load increment or time step size. An inaccurate solution will often not converge. Nine iterative schemes are available including the full Newton method and eight quasi-Newton methods. These are:

- Full Newton
- BFGS (default),
- Broyden,
- Davidon-Fletcher-Powell (DFP) [Schweizerhof 1986],
- Davidon symmetric, [Schweizerhof 1986],
- modified constant arc length with BFGS,
- modified constant arc length with Broyden's,
- modified constant arc length with DFP,
- modified constant arc length with Davidon.

A line search is combined with each of these schemes along with automatic stiffness reformations, as needed, to avoid non-convergence. LS-DYNA defaults to the BFGS quasi - Newton method which is the most robust although the other methods are sometimes superior. Generally, the quasi-Newton methods require fewer iterations than the modified Newton method since they exhibit superlinear local convergence due to the rank one or rank two updates of the stiffness matrix as the iterations proceed. In this chapter, important aspects of the static algorithm are explained, hopefully, in a way that will be understandable to all users. The arc length methods are generally used in solving snap through buckling problems.

Discretization leads to the matrix equations

$$\underset{\sim}{M}\ddot{\underset{\sim}{x}}(t) + \underset{\sim}{F}\left(\underset{\sim}{x}, \dot{\underset{\sim}{x}}\right) - \underset{\sim}{P}\left(\underset{\sim}{x}, t\right) = \underset{\sim}{0} \tag{34.1}$$

where

$\ddot{\underset{\sim}{x}}, \dot{\underset{\sim}{x}}, \underset{\sim}{x}$ = acceleration, velocity, coordinate vectors
$\underset{\sim}{M}$ = mass matrix

$\underset{\sim}{P}$ = body force and external load vector

$\underset{\sim}{F} = \sum_e \underset{\sim}{L}^t \int_{V_e} \underset{\sim}{B}^t \underset{\sim}{\sigma} dV_e$

A diagonal lumped mass matrix is obtained by row summing according to Equation (34.2).

$$M_{ii} = \int \rho \phi_i \sum \phi_i \, dv = \int \rho \phi_i \, dv \tag{34.2}$$

The primary nonlinearities, which are due to geometric effects and inelastic material behavior, are accounted for in $\underset{\sim}{F}$.

$$\underset{\sim}{F}_e \left(\underset{\sim}{x}, \dot{\underset{\sim}{x}} \right) = \int_{V_e} \underset{\sim}{B}^t \underset{\sim}{\sigma} \left(\underset{\sim}{\varepsilon}, \dot{\underset{\sim}{\varepsilon}} \right) dV_e \tag{34.3}$$

where $\underset{\sim}{\varepsilon}$ and $\dot{\underset{\sim}{\varepsilon}}$ are the strains and strainrates. If linear

$$\underset{\sim}{F}_e \left(\underset{\sim}{x}, \dot{\underset{\sim}{x}} \right) = \underset{\sim}{C} \dot{\underset{\sim}{x}} + \underset{\sim}{K} \Delta \underset{\sim}{x} \tag{34.4}$$

where $\underset{\sim}{K}$, $\underset{\sim}{C}$, and $\Delta \underset{\sim}{x}$ are the stiffness matrix, damping matrix, and displacement vector. Additional nonlinearities arise in $\underset{\sim}{P}$ due to geometry dependent applied loads.

Regardless of whether an implicit or explicit integration scheme is used, we require that

$$\underset{\sim}{Q} = \underset{\sim}{M} \ddot{\underset{\sim}{x}} + \underset{\sim}{F} - \underset{\sim}{P} = \underset{\sim}{0} \tag{34.5}$$

If linear

$$\underset{\sim}{Q} = \underset{\sim}{M} \ddot{\underset{\sim}{x}} + \underset{\sim}{C} \dot{\underset{\sim}{x}} + \underset{\sim}{K} \Delta \underset{\sim}{x} - \underset{\sim}{P} = \underset{\sim}{0} \tag{34.6}$$

Explicit integration trivially satisfies these equations since the calculation of the acceleration guarantees equilibrium, i.e.,

$$\ddot{\underset{\sim}{x}} = \underset{\sim}{M}^{-1} \left[\underset{\sim}{P}^n - \underset{\sim}{F}^n \right] \tag{34.7}$$

The explicit update of the velocities and coordinates is given by:

$$\dot{\underset{\sim}{x}}^{n+1/2} = \dot{\underset{\sim}{x}}^{n-1/2} + \Delta t^n \ddot{\underset{\sim}{x}}^n \tag{34.8}$$

$$\underset{\sim}{x}^{n+1} = \underset{\sim}{x}^n + \Delta t^{n+1/2} \dot{\underset{\sim}{x}}^{n+1/2} \tag{34.9}$$

Stability places a limit on the time size. This step size may be very small and, consequently, a large number of steps may be required.

For the implicit solution the residual vector becomes an implicit function of $\underset{\sim}{x}^{n+1}$ only. We seek the vector $\underset{\sim}{x}^{n+1}$ such that

$$\underset{\sim}{Q}\left(\underset{\sim}{x}^{n+1}\right)=\underset{\sim}{0} \tag{34.10}$$

Assume an approximation $\underset{\sim}{x}_k^{n+1}$ to $\underset{\sim}{x}^{n+1}$ for k = 1, 2, 3...etc. In the neighborhood of $\underset{\sim}{x}_k^{n+1}$ we use the linear approximation to $\underset{\sim}{Q}\left(\underset{\sim}{x}^{n+1}\right)=\underset{\sim}{0}$ given by

$$\underset{\sim}{Q}\left(\underset{\sim}{x}^{n+1}\right)=\underset{\sim}{Q}\left(\underset{\sim}{x}_{k-1}^{n+1}\right)+\underset{\sim}{J}\left(\underset{\sim}{x}_{k-1}^{n+1}\right)\Delta\underset{\sim}{x}_k \tag{34.11}$$

and iterate for the solution:

$$\underset{\sim}{x}_k^{n+1}=\underset{\sim}{x}_{k-1}^{n+1}+\Delta\underset{\sim}{x}_k \tag{34.12}$$

$$\underset{\sim}{J}\left(\underset{\sim}{x}_{k-1}^{n+1}\right)=\frac{\partial\underset{\sim}{Q}}{\partial\underset{\sim}{x}}\Big|\underset{\sim}{x}_{k-1}^{n+1} \tag{34.13}$$

The Jacobian matrix is expressed as

$$\underset{\sim}{J}=\underset{\sim}{M}\frac{\partial\ddot{\underset{\sim}{x}}}{\partial\underset{\sim}{x}}+\frac{\partial\underset{\sim}{F}}{\partial\dot{\underset{\sim}{x}}}\frac{\partial\dot{\underset{\sim}{x}}}{\partial\underset{\sim}{x}}+\frac{\partial\underset{\sim}{F}}{\partial\underset{\sim}{x}}-\frac{\partial\underset{\sim}{P}}{\partial\underset{\sim}{x}}=\underset{\sim}{M}\frac{\partial\ddot{\underset{\sim}{x}}}{\partial\underset{\sim}{x}}+\underset{\sim}{C}\frac{\partial\dot{\underset{\sim}{x}}}{\partial\underset{\sim}{x}}+\underset{\sim}{K}_t-\frac{\partial\underset{\sim}{P}}{\partial\underset{\sim}{x}} \tag{34.14}$$

where $\underset{\sim}{C}=\dfrac{\partial\underset{\sim}{F}}{\partial\dot{\underset{\sim}{x}}}$ is the tangent damping matrix, and $\underset{\sim}{K}_t=\dfrac{\partial\underset{\sim}{F}}{\partial\underset{\sim}{x}}$ is the tangent stiffness matrix.

To obtain the solution at load increment n+1 given the solution at load increment n, linearized equations of the form

$$\underset{\sim}{K}_t\left(\underset{\sim}{x}^n\right)\Delta\underset{\sim}{u}_o=\underset{\sim}{P}\left(\underset{\sim}{x}^n\right)^{n+1}-\underset{\sim}{F}\left(\underset{\sim}{x}^n\right) \tag{34.15}$$

are assembled where

$\underset{\sim}{K}_t$ = Positive-definite tangent stiffness matrix.

$\Delta\underset{\sim}{u}_o$ = Desired increment in displacements.

$\underset{\sim}{P}\left(\underset{\sim}{x}^n\right)^{n+1}$ = External load vector at n+1 based on geometry at time n.

$\underset{\sim}{F}\left(\underset{\sim}{x}^n\right)$ = Stress divergence vector at time n.

The displacement vector is updated:

$$\underset{\sim}{x}_1^{n+1}=\underset{\sim}{x}^n+s_0\Delta\underset{\sim}{u}_0 \tag{34.16}$$

and equilibrium iterations begin:

$$\underset{\sim}{K}_{t_j} \Delta \underset{\sim}{u}_i = \underset{\sim}{P}\left(\underset{\sim}{x}_i^{n+1}\right)^{n+1} - \underset{\sim}{F}\left(\underset{\sim}{x}_i^{n+1}\right) = \underset{\sim}{Q}_i^{n+1} \tag{34.17}$$

where the subscripts i denotes the iterate and $j < i$ and s_0 is a parameter between 0 and 1 found from a line search. After each iteration, convergence is checked. Convergence is assumed if the conditions

$$\frac{\left\| \Delta \underset{\sim}{u}_i \right\|}{\underset{\sim}{u}_{\max}} < \varepsilon_d \tag{34.18}$$

and

$$\frac{\left| \Delta \underset{\sim}{u}_i^t \underset{\sim}{Q}_i \right|}{\left| \Delta \underset{\sim}{u}_0^t \underset{\sim}{Q}_0 \right|} < \varepsilon_e \tag{34.19}$$

are satisfied. If convergence is not attained, the displacement vector is updated:

$$\underset{\sim}{x}_{i+1}^{n+1} = \underset{\sim}{x}_i^{n+1} + s_i \Delta \underset{\sim}{u}_i \tag{34.20}$$

and another iteration is performed. A lack of convergence within the allowable number of iterations (default = 10) or divergence

$$\left\| \underset{\sim}{Q}_0^{n+1} \right\| < \left\| \underset{\sim}{Q}_{i+1}^{n+1} \right\| \tag{34.21}$$

causes $\underset{\sim}{K}_t$ to be reformed. Termination occurs if the allowable number of reformations is reached (default =15).

The foregoing integration method with $\underset{\sim}{K}_t$ held constant is called the modified-Newton method. When convergence problems arise, the stiffness matrix is reformed.

Four methods, called quasi-Newton methods, for updating the stiffness matrix are also available:

- Broyden's first method
- Davidon
- DFP
- BFGS

These methods involve

- Line searches
- Rank 1 or rank 2 stiffness updates

and are more expensive than modified Newton but result in a more robust program.

The secant matrices K_i are found via the quasi-Newton equations

$$K_i \Delta u_{i-1} = \Delta Q_i \tag{34.22}$$

where

$$\Delta Q_i = Q_{i-1} - Q_i = \left(P_{i-1} - F_{i-1}\right) - \left(P_i - F_i\right) \tag{34.23}$$

Figure 34.1 below illustrates the calculation of the secant stiffness.

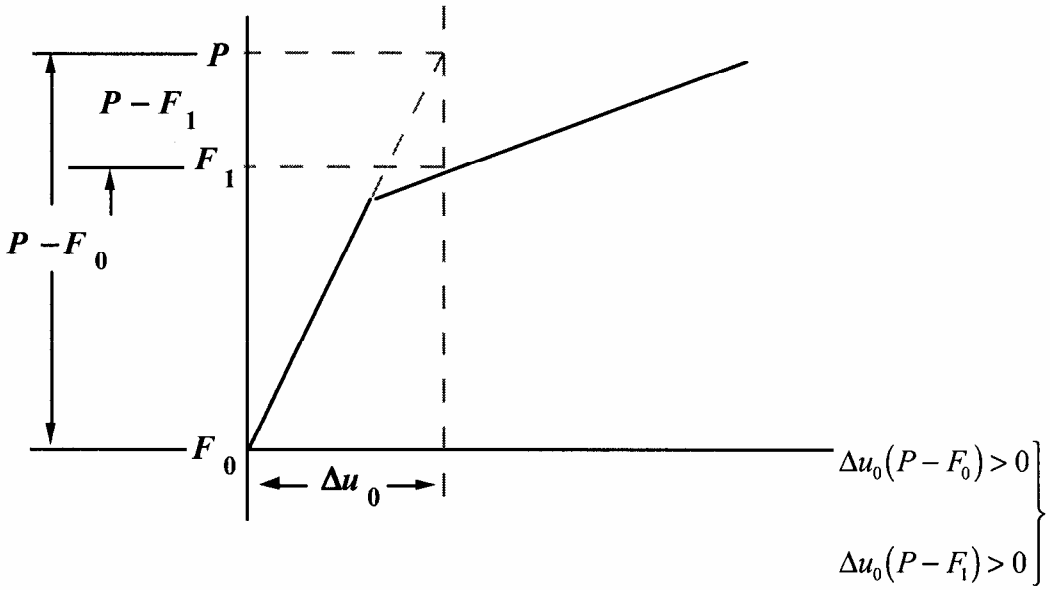


Figure 34.1. The same sign indicates a stable softening system and that s > 1. No line search is necessary.

Two classes of matrix updates that satisfy the quasi-Newton equations are of interest:

- Rank one update: $K_i = K_{i-1} + \alpha z x^t$ (34.24)

- Rank two update: $K_i = K_{i-1} + \alpha z x^t + \beta v y^t$ (34.25)

Substituting $K_i = K_{i-1} + \alpha z x^t$ into Equation (34.22) gives

$$K_{i-1}\Delta u_{i-1} + \alpha z x^t \Delta u_{i-1} = \Delta Q_i \tag{34.26}$$

By choosing

1. $\alpha = \dfrac{1}{x^t \Delta u_{i-1}}$ (34.27)

2. $z = \Delta Q_i - K_{i-1}\Delta u_{i-1}$ (34.28)

Equation (34.22) is satisfied. Note that x is an arbitrary vector but is restricted such that

$$\Delta u_{i-1}^t x \neq 0 \tag{34.29}$$

Broyden set $x = \Delta u_{i-1}$ and obtained the update

$$K_i = K_{i-1} + \frac{\Delta Q - K_{i-1}\Delta u_{i-1}}{\left(\Delta Q - K_{i-1}\Delta u_{i-1}\right)^t \Delta u_{i-1}}\left(\Delta Q - K_{i-1}\Delta u_{i-1}\right)^t$$

resulting in nonsysmmetric secant matrices. The inverse forms are found by the Sherman-Morrison formula:

$$\left(A + a\,b^t\right)^{-1} = A^{-1} - \frac{A^{-1}a\,b^t A^{-1}}{1 + b^t A^{-1} a} \tag{34.30}$$

where A is a nonsingular matrix. The inverse form for Broyden's update can be found by letting

$$a = \frac{\Delta Q_i - K_{i-1}\Delta u_{i-1}}{\Delta u_{i-1}{}^t\, \Delta u_{i-1}} \quad b = \Delta u_{i-1} \tag{34.31}$$

in the Sherman-Morrison formula. Therefore,

$$K_i^{-1} = K_{i-1}^{-1} + \frac{\left(\Delta u_{i-1} - K_{i-1}^{-1}\Delta Q_i\right)\left(\Delta u_{i-1} - K_{i-1}^{-1}\Delta Q_i\right)^t}{\Delta u_{i-1}^t\, K_{i-1}^{-1}\, \Delta Q_i} \tag{34.32}$$

Again, recalling the quasi-Newton equation, Equation (34.22), and substituting Equation (34.25) gives:

$$K_{i-1}\Delta i_{i-1} + \alpha\, z\, x^t \Delta i_{i-1} + \beta\, v\, y^t \Delta i_{i-1} = \Delta Q_i \tag{34.33}$$

Set

1. $$\alpha = \frac{1}{x^t \Delta u_{i-1}} \tag{34.34a}$$

2. $$z = -K_{i-1}\Delta u_{i-1} \tag{34.34b}$$

3. $$\beta = \frac{1}{y^t \Delta u_{i-1}} \tag{34.34c}$$

4. $$v = \Delta Q_i \tag{34.34d}$$

Here x and y are arbitrary vectors that are nonorthogonal to Δu_{i-1}, i.e.,

$$\Delta u_{i-1}^t\, x \neq 0 \tag{34.35}$$

and $\Delta u_{i-1}^{t}\ y \neq 0$ (34.36)

In the BFGS method:

$$x = K_{i-1}\Delta u_{i-1} \qquad y = \Delta Q_i \tag{34.37}$$

which leads to the following update formula

$$K_i = K_{i-1} + \frac{\Delta Q_i \Delta Q_i^{t}}{\Delta u_{i-1}^{t}\,\Delta Q_i} - \frac{K_{i-1}\Delta u_{i-1}\Delta u_{i-1}^{t} K_{i-1}}{\Delta u_{i-1}^{t}\,K_{i-1}\Delta u_{i-1}} \tag{34.38}$$

that preserves the symmetry of the secant matrix. A double application of the Sherman-Morrison formula then leads to the inverse form.

Special product forms have been derived for the DFP and BFGS updates and exploited by Matthies and Strang [1979].

$$K_i^{-1} = \left(I + w_i v_i^{t}\right) K_{i-1}^{-1} \left(I + v_i w_i^{t}\right) \tag{34.39}$$

The primary advantage of the product form is that the determinant of K_i and therefore, the change in condition number can be easily computed to control updates. The updates vectors are easely defined. Let,

$$\delta_i = \Delta u_{i-1} \tag{34.40}$$

$$\tau_i = Q_{i-1} - Q_i \tag{34.41}$$

$$v_i = Q_i - Q_{i-1}\left[1 + \left(\frac{\Delta u_i \tau_i}{\delta_i Q_{i-1}}\right)^{\frac{1}{2}}\right] \tag{34.42}$$

$$w_i = \frac{\delta_i}{\delta_i \tau_i} \tag{34.43}$$

Noting that the determinant of K_i is given by:

$$\det\left(K_i\right) = \det\left(K_{i-1}\right)\left(1 + w^{t} v\right)^{2} \tag{34.44}$$

it can be shown that the change in condition number, C, is

$$C = \frac{\left[\left(v^t v\right)^{\frac{1}{2}}\left(w^t w\right)^{\frac{1}{2}} + \left\{v^t v\, w^t + 4\left(1 + v^t w\right)\right\}^{\frac{1}{2}}\right]^4}{\left[4\left(1 + v^t w\right)\right]^2} \tag{34.45}$$

Following the approach of Matthies and Strang [1979] this condition number is used to decide whether or not to do a given update.

With quasi-Newton updates line searches are usually necessary to achieve convergence. The incremental displacement is uses the variable, s, which is determined during the line search, in the calculation of Δu_i:

$$\Delta u_i = s_i\, d_i \tag{34.46}$$

where

$$d_i = K_i^{-1}\, Q_i \tag{34.47}$$

Therefore, the update of the displacement, u_{i+1}, becomes:

$$u_{i+1} = u_i + s_i d_i \tag{34.48}$$

The line search computes s_i such that the component of $P - F$ in the search direction is zero:

$$d_i \cdot \left[P\left(x_{i+1}\right) - F_i\left(x_{i+1}\right)\right] = 0 \tag{34.49}$$

In LS-DYNA, s is not allowed to exceed unity. In a system which is softening such as that in Figure 34.2 s would be greater than 1 so the line search is skipped. A line search is performed if the stiffness of the structure is increasing, i.e., s is less than unity, also See Figure 34.3:

$$0 < s < 1 \tag{34.50}$$

Since the line search involves additional right hand side evaluations it is very expensive. Improvements in the convergence behavior, however, more than justifiy the expense. See Figure 34.4.

The implementation of Broyden's method illustrates how the quasi-Newton method works. The inverse of K_i^{-1} for the ith iterate is:

$$K_i^{-1} = \left(I + \alpha_{i-1} w_{i-1} \Delta u_{i-1}\right) \ldots \left(I + \alpha_1 w_1 \Delta u_1^t\right)\left(I + \alpha_0 w_0 \Delta u_0^t\right) K_0^{-1} \tag{34.51}$$

where

$$w_i = \Delta u_i - K_i^{-1} Q_i + K_i^{-1} Q_{i+1}$$

$$\alpha_i = \frac{1}{\Delta u_i^t \left(\Delta u_i - w_i \right)} \tag{34.52}$$

are the update factors. Broyden's algorithm can be outlined as follows:

1. Solve $d = K_0^{-1} Q_i$

2. Do $i = 1,\ k-1$

   $$\beta = \Delta u_{i-1}^t d$$

   Recall

   $$d = d + \beta \alpha_{i-1} w_{i-1}$$

3. $$\frac{w_{k-1} = \Delta u_{i-1} - \Delta u_{k-1}}{s + d}$$

4. $$\alpha_{k-1} = \frac{1}{\Delta u_{k-1}^t \left(\Delta u_{k-1} - w_{k-1} \right)}$$

   $$\beta = \Delta u_{k-1}^t d$$

5. $$d_{k-1} = d + \beta \alpha_{k-1} w_{k-1}$$

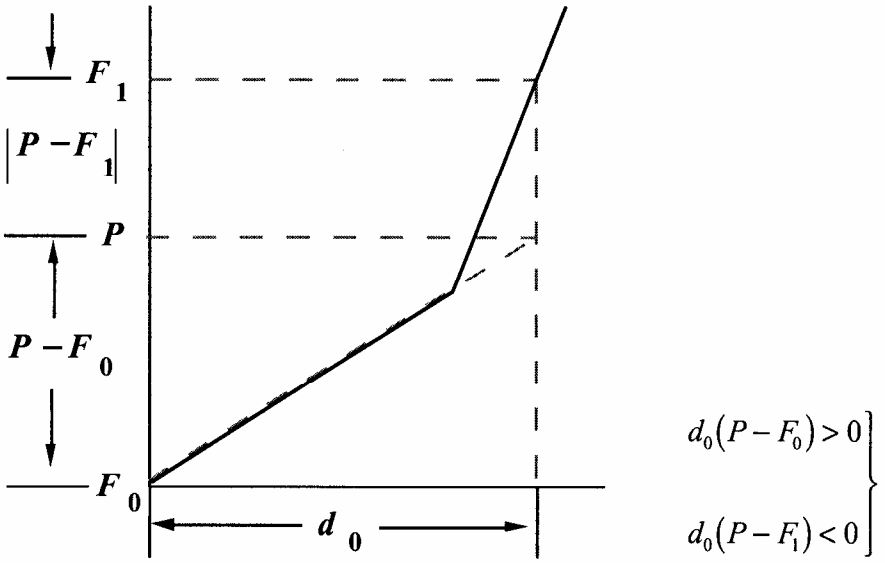


Figure 34.2. The sign change indicates a hardening system and the need for a line search ($0 < s < 1$).

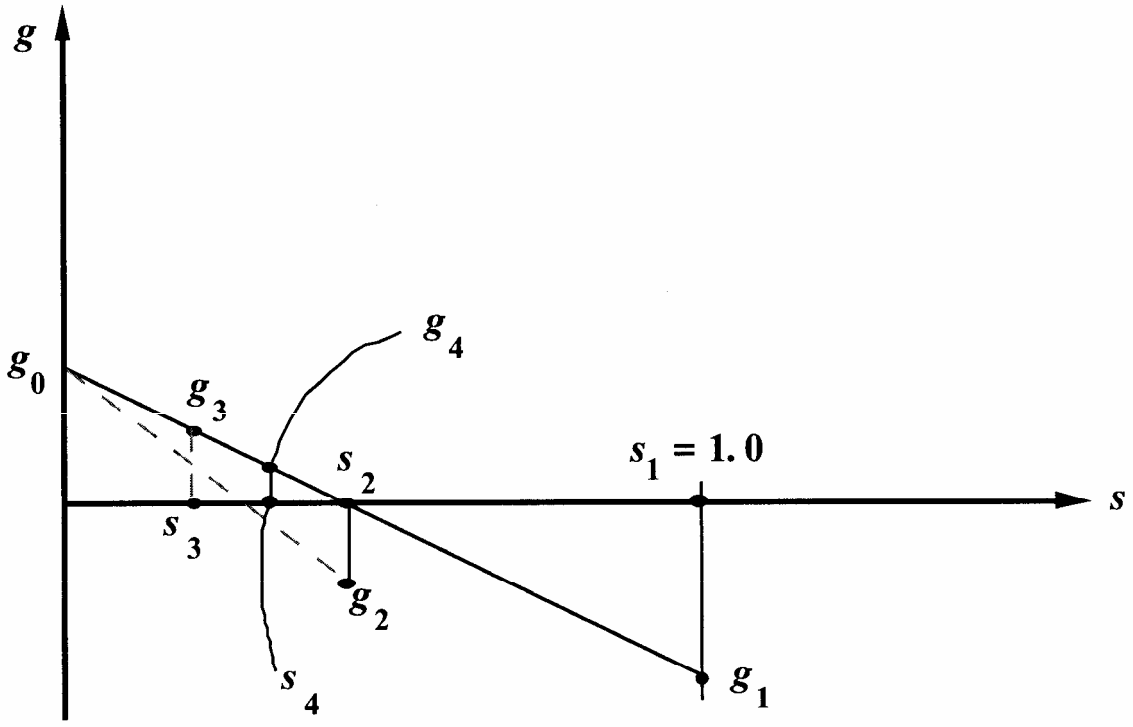


Figure 34.3. Line search with linear interpolation.

In the implementation note that:

- The inverse matrices are not computed in full form
- The solution is restricted to one backward substitution plus several vector operations for each update.
- The update parameters and update vectors, two per iteration, are stored in core.

When the automatic time step option is turned the following steps are taken if convergence fails:

- DT is set to DTMIN
- LS-DYNA backs up to beginning of step
- Solution is retried
- If convergence again fails, solution is retried with a larger step
- And so on.

An optimal number of iteration, IOPT, per step is chosen. ILS is the number of iterations during the last increment. If

ILS < IOPT

$$DT = DT + \left(DTMAX - DT\right) Min\left\{.50, \left[\left(IOPT / ILS\right)^{\frac{1}{2}} - 1.\right]\right\} \tag{34.53}$$

ILS > IOPT

$$DT = DT - \left(DT - DTMIN\right)\left[1. - \left(IOPT / ILS\right)^{\frac{1}{2}}\right] \tag{34.54}$$

where DT, DTMAX, and DTMIN are the current step size, the maximum step size permitted, and the minimum step size permitted, respectively.

34.1 Implicit Dynamics

For the implicit structural dynamics problem, the linearized equilibrium equations may be written in the form

$$\underset{\sim}{M}\ddot{\underset{\sim}{u}}^{n+1} + \underset{\sim}{D}\dot{\underset{\sim}{u}}^{n+1} + \underset{\sim}{K}_t\left(\underset{\sim}{x}^n\right)\Delta\underset{\sim}{u}_0 = \underset{\sim}{P}\left(\underset{\sim}{x}^n\right)^{n+1} - \underset{\sim}{F}\left(\underset{\sim}{x}^n\right), \tag{34.55}$$

where

$\underset{\sim}{M}$ = mass matrix

$\underset{\sim}{D}$ = damping matrix

$\underset{\sim}{u}^{n+1} = \underset{\sim}{x}^{n+1} - \underset{\sim}{X}$ = nodal displacement vector

$\dot{\underset{\sim}{u}}^{n+1}$ = nodal point velocities at time n+1

$\ddot{\underset{\sim}{u}}^{n+1}$ = nodal point accelerations at time n+1.

Equation (34.55) is solved by the unconditionally stable, one-step, Newmark-β time integration scheme

$$\ddot{\underset{\sim}{u}}^{n+1} = \frac{\Delta \underset{\sim}{u}}{\beta \Delta t^2} - \frac{\dot{\underset{\sim}{u}}^n}{\beta \Delta t} - \frac{1}{\beta}\left(\frac{1}{2} - \beta\right)\ddot{\underset{\sim}{u}}^n, \tag{34.56a}$$

$$\dot{\underset{\sim}{u}}^{n+1} = \dot{\underset{\sim}{u}}^n + \Delta t\left(1-\gamma\right)\ddot{\underset{\sim}{u}}^n + \gamma \Delta t \ddot{\underset{\sim}{u}}^{n+1} \tag{34.56b}$$

$$\underset{\sim}{x}^{n+1} = \underset{\sim}{x}^n + \Delta \underset{\sim}{u} \tag{34.56c}$$

Here, Δt is the time step size, and β and γ are the free parameters of integration. For γ = ½ and β =¼ the method becomes the trapezoidal rule and is energy conserving. If

$$\gamma > \frac{1}{2}, \qquad \beta > \frac{1}{4}\left(\frac{1}{2} + \gamma\right)^2 \tag{34.57}$$

numerical damping is induced into the solution leading to a loss of energy and momentum.

Substitution of Equations (34.56) into Equation (34.55) leads to an equation that is similar in form to Equation (34.1):

$$\underset{\sim}{K}^* \, \Delta \underset{\sim}{u}_0 = \underset{\sim}{P}\left(\underset{\sim}{x}^n\right)^{n+1} - \underset{\sim}{F}^*\left(\underset{\sim}{x}^n\right) \tag{34.58}$$

where

$$\underset{\sim}{K}^* = \underset{\sim}{K}_t \frac{1}{\beta \Delta t^2}\underset{\sim}{M} + \frac{\gamma}{\beta \Delta t}\underset{\sim}{D}, \tag{34.59a}$$

$$\underset{\sim}{F}^* = \underset{\sim}{F} - \underset{\sim}{M}\left[\frac{\dot{\underset{\sim}{u}}^n}{\beta \Delta t} + \frac{1}{\beta}\left(\frac{1}{2} - \beta\right)\ddot{\underset{\sim}{u}}^n\right] - \underset{\sim}{D}\left[\left(\frac{\gamma}{\beta} - 1\right)\dot{\underset{\sim}{u}}^n + \Delta t\left(\frac{\gamma}{2\beta} - 1\right)\ddot{\underset{\sim}{u}}^n\right] \tag{34.59b}$$

The solution of Equation (34.58) yields $\Delta \underset{\sim}{u}_0$, the displacement, velocity, and acceleration vectors are updated

$$\ddot{\underset{\sim}{u}}_1^{n+1} = \frac{\Delta \underset{\sim}{u}_0}{\beta \Delta t^2} - \frac{\dot{\underset{\sim}{u}}^n}{\Delta t} - \frac{1}{\beta}\left(\frac{1}{2} - \beta\right)\ddot{\underset{\sim}{u}}^n, \tag{34.60a}$$

$$\dot{\underset{\sim}{u}}_1^{n+1} = \dot{\underset{\sim}{u}}^n + \Delta t\left(1-\gamma\right)\ddot{\underset{\sim}{u}}^n + \gamma \Delta t \ddot{\underset{\sim}{u}}_1^{n+1}, \tag{34.60b}$$

$$\underset{\sim}{x}_1^{n+1} = \underset{\sim}{x}^n + \Delta\underset{\sim}{u}_0, \tag{34.60c}$$

and equilibrium iterations begin with

$$\underset{\sim}{K}^* \Delta\underset{\sim}{u}_i = P\left(\underset{\sim}{x}_i^{n+1}\right)^{n+1} - \underset{\sim}{F}^*\left(\underset{\sim}{x}_i^{n+1}\right) \tag{34.61}$$

where

$$\underset{\sim}{F}^* = \underset{\sim}{F}\left(\underset{\sim}{x}_i^{n+1}\right) + \underset{\sim}{M}\ddot{\underset{\sim}{u}}_i^{n+1} + \underset{\sim}{D}\dot{\underset{\sim}{u}}_i^{n+1} \tag{34.62}$$

At this point the method is essentially the same as the static algorithm.

35. SPARSE DIRECT LINEAR EQUATION SOLVERS

LS-DYNA has 5 options for direct solution of the sparse systems of linear equations that arise in LS-DYNA. All 5 options are based on the multifrontal algorithm [Duff and Reid, 1983]. Multifrontal is a member of the current generation of sparsity preserving factorization algorithms that also have very fast computational rates. That is multifrontal works with a sparsity preserving ordering to reduce the overall size of the direct factorization and the amount of work it takes to compute that factorization.

35.1 Sparsity Preserving Orderings

In LS-DYNA there are two ordering algorithms for preserving the sparsity of the direct factorization. The algorithms are Multiple Minimum Degree (MMD) and METIS [Karypis and Kumar, 1998]. MMD computes the ordering using locally based decisions and a bottom-up approach. It is inexpensive and very effective for small problems that are problems with fewer than 100,000 rows. METIS computes the ordering from a top down approach. While METIS usually takes more time than MMD to compute the ordering, the METIS ordering reduces the work for the factorization enough to recover the additional ordering cost. METIS is especially effective for large problems, especially those that are modeling three-dimensional solids.

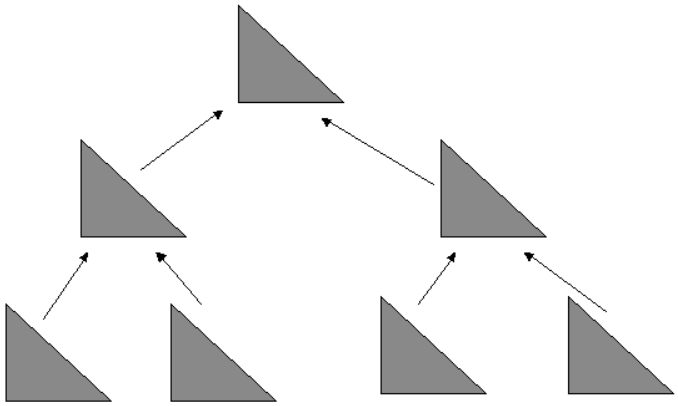
The user can specify either algorithm using keyword *CONTROL_IMPLICIT_LINEAR. The default is to use MMD for problems with fewer than 100,000 rows and METIS for problems with more than 100,000 rows. We recommend that the user try both orderings as sometimes MMD is better than METIS on large problems that are not three-dimensional solids.

35.2 Multifrontal Algorithm

The multifrontal algorithm factors a sparse matrix in a way that vastly reduces the amount of work required to compute the factorization compared to methods such as the frontal, profile, skyline, and variable band. These older methods counted on clustering the nonzero entries of the factorization close to the diagonal to keep the size of the factorization and the amount of work required to compute the factorization to a minimum. The factorization was then computed in a serial, left-to-right fashion, essentially following a chain of computations.

The multifrontal algorithm instead follows a tree of computations where the tree structure is established by the sparsity preserving orderings, See Figure 35.1. It is this tree structure that greatly reduces the work required to compute a factorization and the size of the resulting factorization. At the bottom of the tree, a frontal matrix is assembled with the original matrix data and those columns that are fully assembled are eliminated. The remainder of the frontal matrix is updated from the factored columns and passed up the tree to the parent front in what is called an update matrix. As the computation works its way up the tree, a frontal matrix is formed by assembling the original matrix data and the update matrices from its children in the tree. The fully assembled columns are factored and the remaining columns updated and passed up the tree. At the root (top) of the tree, the remaining columns are factored.

By organizing the factorization as a sequence of partial factorization of dense frontal matrices, the multifrontal algorithm can be very fast in performing the required computations. It can use all of the modern technology for dense linear algebra to get high performance computational kernels that should achieve near peak computational performance for a given processor. Only 1% to 5% of the work of the factorization is performed with slower operations such as scatter/gather.



Multifrontal Elimination Tree

Figure 35.1.

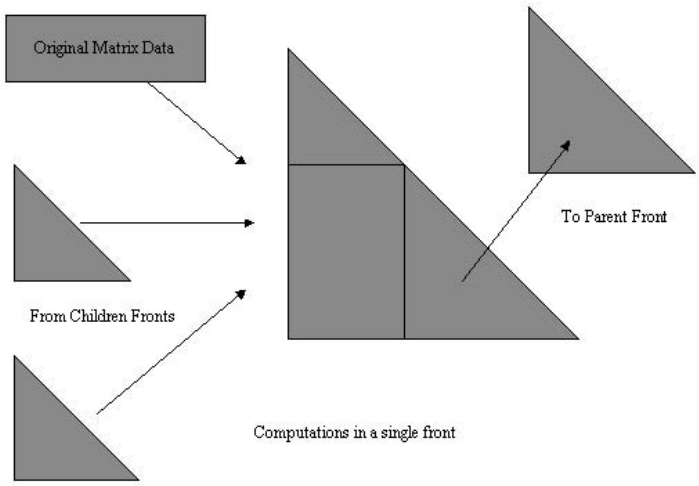


Figure 35.2.

35.3 The Five Solver Options

In LS-DYNA three new direct solution options were added. For backward compatibility, the two older options were kept. The five options are:

Solver No.	Method
1	Older implementaion of Solver No. 4. Uses Real*4 arithmetic, has out of memory capabilities as well as distributed memory parallelism. Only uses MMD ordering. Was former default method. Retained for backward compatibility. We recommend switching to Solver No. 4 for improved performance.
3	Same as 1 except uses Real*8 arithmetic. We recommend switching to Solver No. 5 or 6 for improved performance.
4	Real*4 implementation of multifrontal which includes automatic out-of-memory capabilities as well as distributed memory parallelism. Can use either MMD or METIS orderings. Default method.
5	Real*8 implementation of Solver No. 4.
6	Multifrontal solver from BCSLIB-EXT [Boeing Company, 1999]. Uses Real*8 arithmetic with extensive capabilities for large problems and some Shared Memory Parallelism. Can use either MMD or METIS orderings. If the other solvers cannot factor the problem in the allocated memory, try using this solver.

We strongly recommend using Solvers 4 through 6. Solvers 1 and 3 are included for backward compatibility with older versions of LS-DYNA but are slower the Solvers 4 through 6. Solvers 4 and 5 are 2 to 6 times faster than the older versions, respectively. Solver 6 on a single processor computer should be comparable to Solver 5 but has more extensive capabilities for solving very large problems with limited memory. Solvers 4 and 5 should be used for distributed memory parallel implementations of LS-DYNA. Solver 6 can be used in shared memory parallel.

In an installation of LS-DYNA where both integer and real numbers are stored in 8 byte quantities, then Solvers 1 and 3 are equivalent and Solvers 4 and 5 are equivalent.

35.4 Treating Matrix Singularities

LS-DYNA has two different techniques for preventing singularities in the stiffness matrix, K. The most common type of matrix singularity arises from the use of certain types of shell elements. These shell elements generate no matrix contribution in the normal direction for each node. Depending on the geometry around the node and what other types of elements are connected to the node, there may or may not be a matrix singularity associated with the rotation around the normal direction at one or more nodes. This is commonly called the drilling rotation singularity.

The first way LS-DYNA has for preventing such matrix singularities is to add a small amount of stiffness in the normal direction at each node of every shell element that has the drilling rotation problem. This “drilling” stiffness matrix is orthogonal to rigid body motions. The user can control whether this approach is used and how much stiffness is added via the *CONTROL_IMPLICIT_SOLUTION keyword card. DRLMTH and DRLPARM are set in

fields 5 and 6. If DRLMTH = 1 then this approach is used. The amount of stiffness added is controlled via DRLPARM. The default for DRLPARM is 1.0 for linear problems and 100.0 for nonlinear problems. DRLPARM*.0001 is added in the normal direction at each node to the diagonal terms associated with the rotational degrees of freedom for certain types of elemental matrices. For eigenvalue problems the amount of stiffness added is 1.E-12.

Adding stiffness to handle the drilling rotation problem has been used extensively. While a robust and reliable approach, its drawback is that the added stiffness may affect the quality of the computed results. The user can also select not to use this approach and depend solely on AUTOSPC, the other method for preventing matrix singularities.

AUTOSPC stands for AUTOmatic Single Point Constraints. AUTOSPC examines K after all of the elemental matrices have been assembled and all of the constraints have been applied for columns that are singular. The user controls AUTOSPC using ASPCMTH and ASPCTOL, fields 7 and 8 of the CONTROL_IMPLICIT_SOLUTION keyword card. If ASPCMTH = 1, AUTOSPC is used. For every set of columns of K that correspond to the translational or rotational degrees of freedom for a node or rigid body those columns are examined. The singular values of the diagonal block of the columns are computed. If the ratio of the smallest and largest singular values is less than ASPCTOL then the set of the columns is declared singular and a constraint is imposed to remove the singularity. The defaults for ASPCTOL is 1.E-6 when the matrix is assembled in REAL*4 precision and 1.E-8 when REAL*8 is used. The imposed constraint sets the degree of freedom to zero that is associated with the column that has the largest component in the null space of the columns. If all of the singular values are less than ASPCTOL all of the degrees of freedom in the block are constrained to zero.

36. SPARSE EIGENSOLVER

LS-DYNA now includes the Block Shift and Invert Lanczos eigensolver from BCSLIB-EXT. This eigensolver is used in LS-DYNA to compute the normal modes and mode shapes for the vibration analysis problem

$$K\Phi = M\Phi\Lambda$$

where K and M are the assembled stiffness and mass matrices, Φ are the eigenvectors (normal mode shapes) and Λ are the eigenvalues (normal modes).

The Lanczos algorithm iteratively computes a better and better approximation to the extreme eigenvalues and the corresponding eigenvectors of the ordinary eigenvalue problem $A\Phi = \Phi\Lambda$ where A is a real symmetric matrix using only matrix-vector multiplies. To use Lanczos on the vibration analysis problem it must be changed to

$$(K - \sigma M)^{-1} M\Phi = \Phi\Theta$$

where each shifted and inverted eigenvalue $\theta_i = 1/(\lambda_i - \sigma)$. This change to an ordinary eigenvalue problem makes the eigenvalues of the original problem near σ become the extreme eigenvalues of the ordinary eigenvalue problem. This helps the Lanczos algorithm compute those eigenvalues quickly.

BCSLIB-EXT uses a sophisticated logic to choose a sequence of shifts, σ_i, to enable the computation of a large number of eigenvalues and eigenvectors. At each shift the factorization of $K - \sigma M$ is computed. The factorization provides the matrix inertia that tells the algorithm how many eigenvalues are to the left of any given σ_i. Given the inertia information, BCSLIB-EXT can tell how many eigenvalues are in a given interval and determine if all of the eigenvalues in that interval have been computed. As a result, BCSLIB-EXT is a very robust eigensolver.

The implementation of BCSLIB-EXT in LS-DYNA includes a shared memory implementation. However only limited parallel speed-up is available for most problems. This is because the eigensolution requires a vast amount of data that for most problems this data has to be stored on I/O files. The wall clock time for the eigensolver is as much a function of the speed of the I/O subsystem on the computer as the CPU time. Parallelism can only speed up the CPU time and does nothing to speed-up the I/O time.

The user can request how many and which eigenvalues to compute using the keyword *CONTROL_IMPLICIT_EIGENVALUE. Via the parameters on this keyword, the user can request any of the following problems:

- Compute the lowest 50 modes (that is nearest to zero)
- Compute the 20 modes nearest to 30 Hz.
- Compute the lowest 20 modes between 10 Hz and 50 Hz.
- Compute all of the modes between 10 Hz and 50 Hz.
- Compute all of the modes below 50 Hz.
- Compute the 30 modes nearest to 30 Hz between 10 Hz and 50 Hz.

36.1 The Eigenvalue Problem for Rotating Systems

Rotating systems, such as the compressor and turbine assembly of a jet engine, have large inertial forces that are functions of the distance from the axis of rotation. These forces are naturally generated in LS-DYNA if the system is modeled as rotating at the proper angular velocity. However, this is often inconvenient for postprocessing because the solution has an oscillatory character imposed it due to the rotation. A commonly used approach to bypass this difficulty is to impose body forces that are equivalent to the inertial forces due to rotation. In LS-DYNA, these forces are imposed through *LOAD_BODY_GENERALIZED and related keywords.

For a system with a constant angular velocity $\vec{\omega} = \{\omega_x, \omega_y, \omega_z\}^T$, the body force added to the applied load is

$$\vec{F}_B = -M\{2\vec{\omega}\times\dot{\vec{u}} + \vec{\omega}\times(\vec{\omega}\times(\vec{r}+\vec{u}))\}.$$

In this equation, $\vec{r}$ is the initial coordinate at a point and $\vec{u}$ is the displacement. Because the body force is a function of both the velocity and displacement, it contributes both damping and stiffness matrices to the eigenvalue problem. Furthermore, since the term involving the initial coordinate creates an initial stress in the structure, the initial stress matrix K^σ (also called the nonlinear stiffness) is also added to the eigenvalue problem.

The damping and stiffness terms are easily derived in matrix form once the cross product is expressed in matrix form.

$$\vec{\omega}\times\vec{r} = \Omega\vec{r} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}$$

The linearized equation for vibration is

$$M\ddot{\vec{u}} + C\dot{\vec{u}} + \left[K + K^\sigma\right]u = -M\{\Omega\dot{\vec{u}} + \Omega^2\vec{u}\}.$$

Rewriting this equation into the traditional form for eigenvalue analysis produces:

$$M\ddot{\vec{u}} + C^R\dot{\vec{u}} + K^R\vec{u} = 0$$
$$C^R = C + M\Omega$$
$$K^R = K + K^\sigma + M\Omega^2$$

The inertial contribution to the damping matrix is not symmetric, nor does it fulfill the requirements for Rayleigh damping, and therefore the resulting eigenvectors and eigenvalues are complex. The inertial term to the stiffness matrix is, however, symmetric and it softens the structure, thereby reducing its natural frequencies.

If the damping term is omitted, the matrices are real and symmetric, and the resulting eigenvalue problem may be solved with the standard eigenvalue methods. The natural frequencies won't be correct, but they are typically close enough to the complex solution that they can be used for initial design calculations.

37. BOUNDARY ELEMENT METHOD

LS-DYNA can be used to solve for the steady state or transient fluid flow about a body using a boundary element method. The method is based on the work of Maskew [1987], with the extension to unsteady flow with arbitrary body motion following the work of Katz and Maskew [1988]. The theory which underlies the method is restricted to inviscid, incompressible, attached fluid flow. The method should not be used to analyze flows where shocks or cavitation are present.

In practice the method can be successfully applied to a wider class of fluid flow problems than the assumption of inviscid, incompressible, attached flow would imply. Many flows of practical engineering significance have large Reynolds numbers (above 1 million). For these flows the effects of fluid viscosity are small if the flow remains attached, and the assumption of zero viscosity may not be a significant limitation. Flow separation does not necessarily invalidate the analysis. If well-defined separation lines exist on the body, then wakes can be attached to these separation lines and reasonable results can be obtained. The Prandtl-Glauert rule can be used to correct for non-zero Mach numbers in air, so the effects of aerodynamic compressibility can be correctly modeled (as long as no shocks are present).

37.1 Governing Equations

The partial differential equation governing inviscid, incompressible fluid flow is given by LaPlace's equation

$$\nabla^2 \Phi = 0 \tag{37.1}$$

where Φ is the velocity potential (a scalar function). The fluid velocity anywhere in the flow field is equal to the gradient of Φ. The boundary condition on this partial differntial equation is provided by the condition that there must be no flow in the direction normal to the surface of the body. Note that time does not appear in Equation (37.1). This is because the assumption of incompressibility implies an infinite sound speed; any disturbance is felt everywhere in the fluid instantaneously. Although this is not true for real fluids, it is a valid approximation for a wide class of low-speed flow problems.

Equation (37.1) is solved by discretizing the surface of the body with a set of quadrilateral or triangular surface segments (boundary elements). Each segment has an associated source and doublet strength. The source strengths are computed from the free-stream velocity, and the doublet strengths are determined from the boundary condition. By requiring that the normal component of the fluid velocity be zero at the center of each surface segment, a linear system of equations is formed with the number of equations equal to the number of unknown doublet strengths. When this system is solved, the doublet strengths are known. The source and doublet distributions on the surface of the body then completely determine the flow everywhere in the fluid.

The linear system for the unknown doublet strengths is shown in Equation (37.2).

$$[mic]\{\mu\} = \{rhs\} \tag{37.2}$$

In this equation μ are the doublet strengths, [*mic* is the matrix of influence coefficients which relate the doublet strength of a given segment to the normal velocity at another segment's

mid-point, and *rhs* is a right-hand-side vector computed from the known source strengths. Note that *mic* is a fully-populated matrix. Thus, the cost to compute and store the matrix increases with the square of the number of segments used to discretize the surface of the body, while the cost to factor this matrix increases with the cube of the number of segments. Users should keep these relations in mind when defining the surface segments. A surface of 1000 segments can be easily handled on most any computer, but a 10,000 segment representation would not be feasible on any but the most powerful supercomputers.

37.2 Surface Representation

The surface of the body is discretized by a set of triagular or quadrilateral surface segments. The best fluid-structure interaction results will be obtained if the boundary element segments coincide with, and use identical nodes as, the structural segments used to define the body. An input format has been implemented to make this easy if thin shell elements are used to define the structure (see the User's Manual). Using the same nodes to define the boundary elements and the structure guarantees that the boundary elements follow the structure as it deforms, and provides a means for the fluid pressure to load the structure.

The nodes used to define the corners of the boundary element segments must be ordered to provide a normal vector which points into the fluid (Figure 37.1).

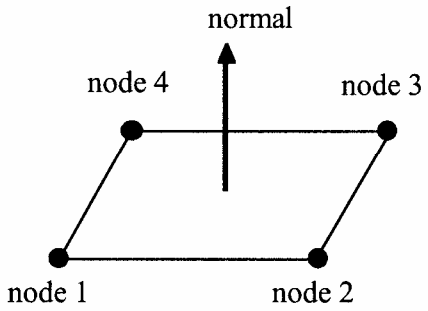


Figure 37.1. Counter-clockwise ordering of nodes when viewed from fluid looking towards solid provides unit normal vector pointing into the fluid.

Triangular segments are specified by using the same node for the 3rd and 4th corner of the segment (the same convention used for shell elements in LS-DYNA). Very large segments can be used with no loss of accuracy in regions of the flow where the velocity gradients are small. The size of the elements should be reduced in areas where large velocity gradients are present. Finite-precision arithmetic on the computer will cause problems if the segment aspect ratios are extremely large (greater than 1000). The most accurate results will be obtained if the segments are rectangular, and triangular segments should be avoided except for cases where they are absolutely required.

37.3 The Neighbor Array

The fluid velocities (and, therefore, the fluid pressures) are determined by the gradient of the velocity potential. On the surface of the body, this can be most easily computed by taking derivatives of the doublet distribution on the surface. These derivatives are computed using the doublet strengths on the boundary element segments. The "Neighbor Array" is used to specify

how the gradient is computed for each boundary element segment. Thus, accurate results will not be obtained unless the neighbor array is correctly specified by the user.

Each boundary element segment has 4 sides (see Figure 37.2). Side 1 connects the 1st and 2nd nodes, side 2 connects the 2nd and 3rd nodes, etc. The 4th side is null for triangular segments.

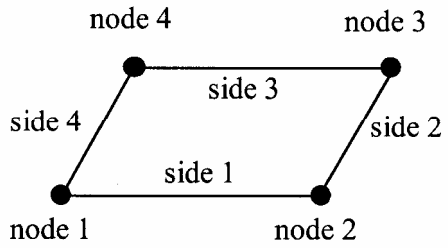


Figure 37.2. Each segment has 4 sides.

For most segments the specification of neighbors is straightforward. For the typical case a rectangular segment is surrounded by 4 other segments, and the neighbor array is as shown in Figure 37.3. A biquadratic curve fit is computed, and the gradient is computed as the analytical derivative of this biquadratic curve fit evaluated at the center of segment j.

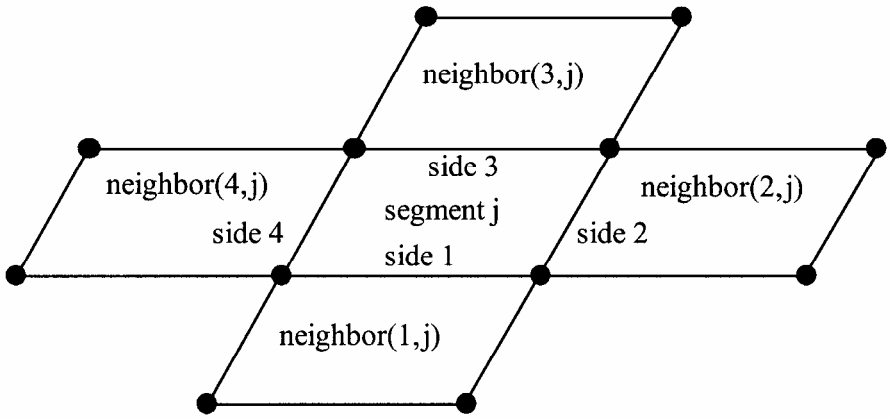


Figure 37.3. Typical neighbor specification.

There are several situations which call for a different specification of the neighbor array. For example, boundary element wakes result in discontinuous doublet distributions, and the biquadratic curve fit should not be computed across a wake. Figure 37.4 illustrates a situation where a wake is attached to side 2 of segment j. For this situation two options exist. If neighbor(2,j) is set to zero, then a linear computation of the gradient in the side 2 to side 4 direction will be made using the difference between the doublet strengths on segment j and segment neighbor(4,j). By specifying neighbor(2,j) as a negative number the biquadratic curve fit will be retained. The curve fit will use segment j, segment neighbor(4,j), and segment -neighbor(2,j); which is located on the opposite side of segment neighbor(4,j) as segment j. The

derivative in the side 2 to side 4 direction is then analytically evaluated at the center of segement j using the quadratic curve fit of the doublet strengths on the three segments shown.

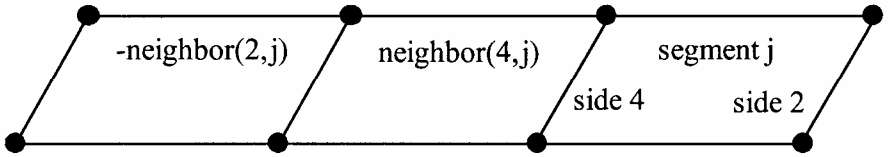


Figure 37.4. If neighbor(2,j) is a negative number it is assumed to lie on the opposite side of neighbor(4,j) as segment j.

A final possibility is that no neighbors at all are available in the side 2 to side 4 direction. In this case both neighbor(2,j) and neighbor(4,j) can be set to zero, and the gradient in that direction will be assumed to be zero. This option should be used with caution, as the resulting fluid pressures will not be accurate for three-dimensional flows. However, this option is occaisionally useful where quasi-two dimensional results are desired. All of the above options apply to the side 1 to side 3 direction in the obvious ways.

For triangular boundary element segments side 4 is null. Gradients in the side 2 to side 4 direction can be computed as described above by setting neighbor(4,j) to zero (for a linear derivative computation) or to a negative number (to use the segment on the other side of neighbor(2,j) and a quadratic curve fit). There may also be another triangular segment which can be used as neighbor(4,j) (see Figure 37.5).

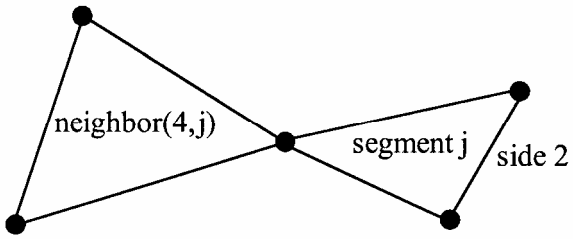


Figure 37.5. Sometimes another triangular boundary element segment can be used as neighbor(4,j).

37.4 Wakes

Wakes should be attached to the boundary element segments at the trailing edge of a lifting surface (such as a wing, propeller blade, rudder, or diving plane). Wakes should also be attached to known separation lines (such as the sharp leading edge of a delata wing at high angles of attack). Wakes are required for the correct computation of surface pressures for these situations. As described above, two segments on opposite sides of a wake should never be used as neighbors. Correct specification of the wakes is required for accurate results.

Wakes convect with the free-stream velocity. The number of segments in the wake is controlled by the user, and should be set to provide a total wake length equal to 5-10 times the characteristic streamwise dimension of the surface to which the wake is attached. For example, if the wake is attached to the trailing edge of a wing whose chord is 1, then the total length of the

wake should at least 5, and there is little point in making it longer than 10. Note that each wake segment has a streamwise length equal to the magnitude of the free stream velocity times the time increment between calls to the Boundary Element Method routine. This time increment is the maximum of the LS-DYNA time step and DTBEM specified on Card 1 of the BEM input. The influence coefficients for the wake segments must be recomputed for each call to the Boundary Element Method, but these influence coefficients do not enter into the matrix of influence coefficients which must be factored.

37.5 Execution Time Control

The Boundary Element Method will dominate the total execution time of a LS-DYNA calculation unless the parameters provided on Card 1 of the BEM input are used to reduce the number of calls to the BEM. This can usually be done with no loss in accuracy since the characteristic time of the structural dynamics and the fluid flow are so different. For example, the characteristic time in LS-DYNA is given by the characteristic length of the smallest structural element divided by the speed of sound of the material. For typical problems this characteristic time might be on the order of microseconds. Since the fluid is assumed to be incompressible (infinite speed of sound), the characteristic time of the fluid flow is given by the streamwise length of the smallest surface (e.g. a rudder) divided by the fluid velocity. For typical problems this characteristic time might be on the order of milliseconds. Thus, for this example, the boundary element method could be called only once for every 1000 LS-DYNA iterations, saving an enormous amount of computer time.

The parameter DTBEM on Card 1 of the BEM input is used to control the time increment between calls to the boundary element method. The fluid pressures computed during the last call to the BEM will continue to be used for subsequent LS-DYNA iterations until DTBEM has elapsed.

A further reduction in execution time may be obtained for some applications using the input parameter IUPBEM. This parameter controls the number of calls to the BEM routine between computation (and factorization) of the matrix of influence coefficients (these are time-consuming procedures). If the motion of the body is entirely rigid body motion there is no need to recompute and factor the matrix of influence coefficients, and the execution time of the BEM can be significantly reduced by setting IUPBEM to a very large number. For situations where the motion of the body is largely rigid body motion with some structural deformation an intermediate value (e.g. 10) for IUPBEM can be used. It is the user's responsibility to verify the accuracy of calculations obtained with IUPBEM greater than 1.

The final parameter for controlling the execution time of the boundary element method is FARBEM. The routine which calculates the influence coefficients switches between an expensive near-field and an inexpensive far-field calculation depending on the distance from the boundary element segment to the point of interest. FARBEM is a nondimensional parameter which determines where the far-field boundary lies. Values of FARBEM of 5 and greater will provide the most accurate results, while values as low as 2 will provide slightly reduced accuracy with a 50% reduction in the time required to compute the matrix of influence coefficients.

37.6 Free-Stream Flow

The free-stream flow is specified in the second card of input. The free-stream velocity is assumed to be uniform. The free-stream static pressure is assumed to be uniform, and can be used to load the structure for hydrostatic pressure. If the structure has an internal pressure, the free-stream static pressure should be set to the difference between the external and internal static

pressures. The Mach number can be used to correct for the effect of compressibility in air (as long as no shocks are present). Following the Prandtl-Glauert correction, the pressures due to fluid flow are increased as follows

$$dp_{corrected} = \frac{dp_{uncorrected}}{\sqrt{1-M^2}} \tag{37.3}$$

where M is the free-stream Mach number. Note that this correction is only valid for flows in a gas (it is not valid for flows in water).

38. SPH IN LS-DYNA

38.1 Introduction

Smoothed Particle Hydrodynamics (SPH) is an N-body integration scheme developed by Lucy, Gingold and Monaghan [1977]. The method was developed to avoid the limitations of mesh tangling encountered in extreme deformation problems with the finite element method. The main difference between classical methods and SPH is the absence of a grid. Therefore, the particles are the computational framework on which the governing equations are resolved. This new model requires a new calculation method, which is briefly explained in the following.

38.1 Basic Sph Formalism

38.1.1 Definitions

The particle approximation of a function is:

$$\Pi^h f(x) = \int f(y) W(x-y,h) dy \tag{38.1}$$

Where W is the kernel function.

The Kernel function W is defined using the function θ by the relation:

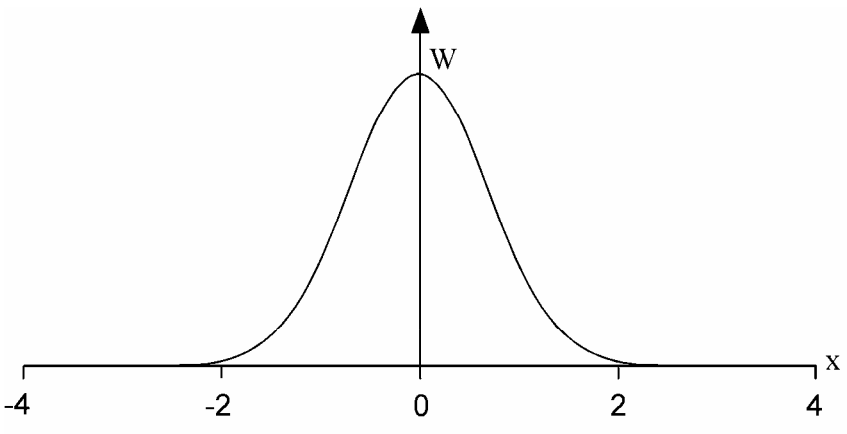
$$W(\mathbf{x},h) = \frac{1}{h(\mathbf{x})^d}\theta(\mathbf{x}) \tag{38.2}$$

Where d is the number of space dimensions and h is the so-called smoothing length which varies in time and in space.

$W(\mathbf{x},h)$ should be a centrally peaked function. The most common smoothing kernel used by the SPH community is the cubic B-spline which is defined by choosing θ as:

$$\theta(u) = C \times \begin{cases} 1 - \frac{3}{2}u^2 + \frac{3}{4}u^3 & \text{for } /u/ \le 1 \\ \frac{1}{4}(2-u)^3 & \text{for } 1 \le /u/ \le 2 \\ 0 & \text{for } 2 < /u/ \end{cases} \tag{38.3}$$

Where C is a constant of normalization that depends on the number of space dimensions.



The SPH method is based on a quadrature formula for moving particles $((\mathbf{x}_i(t))$ $i \in \{1..N\}$, where $\mathbf{x}_i(t)$ is the location of particle i, which moves along the velocity field ***v***.

The particle approximation of a function can now be defined by:

$$\Pi^h f(\mathbf{x}_i) = \sum_{j=1}^{N} w_j f(\mathbf{x}_i) W(\mathbf{x}_i - \mathbf{x}_j, h) \tag{38.4}$$

Where $w_j = \frac{m_j}{\rho_j}$ is the "weight" of the particle. The weight of a particle varies proportionally to the divergence of the flow.

The SPH formalism implies a derivative operator. A particle approximation for the derivative operator must be defined. Before giving the definition of this approximation, we define the gradient of a function as:

$$\nabla f(x) = \nabla f(x) - f(x)\nabla 1(x) \tag{38.5}$$

Where 1 is the unit function.

Starting from this relation, we can define the particle approximation to the gradient of a function:

$$\Pi^h \nabla f(\mathbf{x}_i) = \sum_{j=1}^{N} \frac{m_j}{\rho_j} \left[f(\mathbf{x}_j) A_{ij} - f(\mathbf{x}_i) A_{ij} \right] \tag{38.6}$$

where $A_{ij} = \frac{1}{h^{d+1}} \theta'(\frac{// x_i - x_j //}{h})$

We can also define the particle approximation of the partial derivative $\frac{\partial}{\partial x^\alpha}$:

$$\Pi^h(\frac{\partial f}{\partial x^\alpha})(\mathbf{x}_i) = \sum_{j=1}^{N} w_j f(\mathbf{x}_j A^\alpha(\mathbf{x}_i, \mathbf{x}_j) \tag{38.7}$$

where A is the operator defined by: $A(x_i, x_j) = \frac{1}{h^{d+1}(x_i, x_j)} \frac{(x_i - x_j)}{// x_i - x_j //} \theta' \left(\frac{// x_i - x_j //}{h(x_i, x_j)} \right)$

A^α is the component α of the A vector.

38.1.2 Discrete Form of Conservative Equations

We are looking for the solution of the equation:

$$L_v(\phi) + divF(\mathbf{x}, t, \phi) = S \tag{38.8}$$

where $\phi \in R^d$ is the unknown, F^β with $\beta \in \{1..d\}$ represents the conservation law and L_v is the transport operator defined by:

$$L_v : \phi \to L_v(\phi) = \frac{\partial \phi}{\partial t} + \sum_{l=1}^{d} \frac{\partial(\mathbf{v}^l \phi)}{\partial x^l} \tag{38.9}$$

The strong formulation approximation:

In the search of the strong solution, the equation is kept at its initial formulation. The discrete form of this problem implies the definition of the operator of derivation D defined by:

$$D : \phi \to D\phi(x) = \nabla\phi(x) - \phi(x)\nabla 1(x) \tag{38.10}$$

The particle approximation of this operator is:

$$D_h\phi(\mathbf{x}_i) = \sum_{j=1}^{N} w_j(\phi(\mathbf{x}_j) - \phi(\mathbf{x}_i))A_{ij} \tag{38.11}$$

where A_{ij} is defined previously.

Finally, the discrete form of the strong formulation is written:

$$\frac{d}{dt}(w_i\phi(\mathbf{x}_i)) + w_i D_h F(\mathbf{x}_i) = w_i S(\mathbf{x}_i) \tag{38.12}$$

But this form is not conservative; therefore the strong formulation is not numerically acceptable. Thus, we are compelled to use the weak form.

The weak formulation approximation:

In the weak formulation, the adjoint of the L_v operator is used:

$$L_v^* : \phi \to L_v^*(\phi) = \frac{\partial \phi}{\partial t} + \sum_{l=1}^{d} v^l \frac{\partial \phi}{\partial x^l} \tag{38.13}$$

The discrete form of this operator corresponds to the discrete formulation of the adjoint of $D_{h,s}$:

$$D_{h,s}^* \phi(\mathbf{x}_i) = \sum_{j=1}^{N} w_j (\phi(\mathbf{x}_i) A_{ij} - \phi(\mathbf{x}_j) A_{ji}) \tag{38.14}$$

A discrete adjoint operator for the partial derivative is also necessary, and is taken to be the $\alpha - th$ component of the operator:

$$D_\alpha^* \phi(\mathbf{x}_i) = \sum_{j=1}^{N} w_j \phi(\mathbf{x}_j) A^\alpha(\mathbf{x}_i, \mathbf{x}_j) - w_j \phi(\mathbf{x}_i) A^\alpha(\mathbf{x}_j, \mathbf{x}_i) \tag{38.15}$$

These definitions are leading to a conservative method. Hence, all the conservative equations encountered in the SPH method will be solved using the weak form.

38.1.3 Applications to Conservative Equations

With the definitions explained above, the conservative equations can now be written in their discrete form.

Momentum conservation equation:

The momentum equation is:

$$\frac{d\mathbf{v}^\alpha}{dt}(\mathbf{x}_i(t)) = \frac{1}{\rho_i} \frac{\partial(\sigma^{\alpha\beta})}{\partial x_i}(\mathbf{x}_i(t)) \tag{38.16}$$

where α, β are the space indices.

The particle approximation of the weak form of this equation is:

$$\frac{d\mathbf{v}^\alpha}{dt}(\mathbf{x}_i) = \sum_{j=1}^{N} m_j \left(\frac{\sigma^{\alpha,\beta}(\mathbf{x}_i)}{\rho_i^2} A_{ij} - \frac{\sigma^{\alpha,\beta}(\mathbf{x}_j)}{\rho_j^2} A_{ji} \right) \tag{38.17}$$

Energy conservation equation:

The energy equation is:

$$\frac{dE}{dt} = -\frac{P}{\rho} \nabla \mathbf{v} \tag{38.18}$$

The particle approximation of the weak form of this equation is:

$$\frac{dE}{dt}(\mathbf{x}_i) = -\frac{P_i}{\rho_i^2}\sum_{j=1}^{N} m_j (v(\mathbf{x}_j) - v(\mathbf{x}_i))A_{ij} \tag{38.19}$$

38.1.4 Formulation available in LS-DYNA

It is easy from the general formulation displayed in Equation (38.14) to extend the SPH formalism to a set of equations of discretization for the momentum equation.

For example, if we choose the smoothing function to be symmetric, this can lead to the following equation:

$$\frac{d\mathbf{v}^{\alpha}}{dt}(\mathbf{x}_i) = \sum_{j=1}^{N} m_j \left(\frac{\sigma^{\alpha,\beta}(\mathbf{x}_i)}{\rho_i^2} + \frac{\sigma^{\alpha,\beta}(\mathbf{x}_j)}{\rho_j^2}\right)A_{ij} \tag{38.20}$$

This is what we call the “symmetric formulation”, which is chosen in the *CONTROL_SPH card (IFORM=2).

Another possible choice is to define the momentum equation by:

$$\frac{d\mathbf{v}^{\alpha}}{dt}(\mathbf{x}_i) = \sum_{j=1}^{N} m_j \left(\frac{\sigma^{\alpha,\beta}(\mathbf{x}_i)}{\rho_i\,\rho_j} A_{ij} - \frac{\sigma^{\alpha,\beta}(\mathbf{x}_j)}{\rho_i\,\rho_j} A_{ji}\right) \tag{38.21}$$

This is the “fluid formulation” invoked with IFORM=5 which gives better results than other SPH formulations when fluid material are present, or when material with very different stiffness are used.

38.2.1 Sorting

In the SPH method, the location of neighboring particles is important. The sorting consists of finding which particles interact with which others at a given time. A bucket sort is used that consists of partitioning the domain into boxes where the sort is performed. With this partitioning the closest neighbors will reside in the same box or in the closest boxes. This method reduces the number of distance calculations and therefore the CPU time.

38.2.2 Artificial Viscosity

The artificial viscosity is introduced when a shock is present. Shocks introduce discontinuities in functions. The role of the artificial viscosity is to smooth the shock over several particles. To take into account the artificial viscosity, an artificial viscous pressure term Π_{ij} [Monaghan & Gingold 1983] is added such that:

$$p_i \rightarrow p_i + \Pi_{ij} \tag{38.22}$$

where $\Pi_{ij} = \frac{1}{\bar{\rho}_{ij}}(-\alpha\mu_{ij}\bar{c}_{ij} + \beta\mu_{ij}^2)$

The notation $\bar{X}_{ij} = \frac{1}{2}(X_i + X_j)$ has been used for median between X_i and X_j, c is the adiabatic sound speed, and

$$\mu_{ij} = \begin{cases} \bar{h}_{ij} \dfrac{v_{ij} r_{ij}}{r_{ij}^2 + \eta^2} & if\ v_{ij} r_{ij} < 0 \\ 0 \qquad otherwise \end{cases} \tag{38.23}$$

Here, $v_{ij} = (v_i - v_j)$, and $\eta^2 = 0.01\,\bar{h}_{ij}^2$ which prevents the denominator from vanishing.

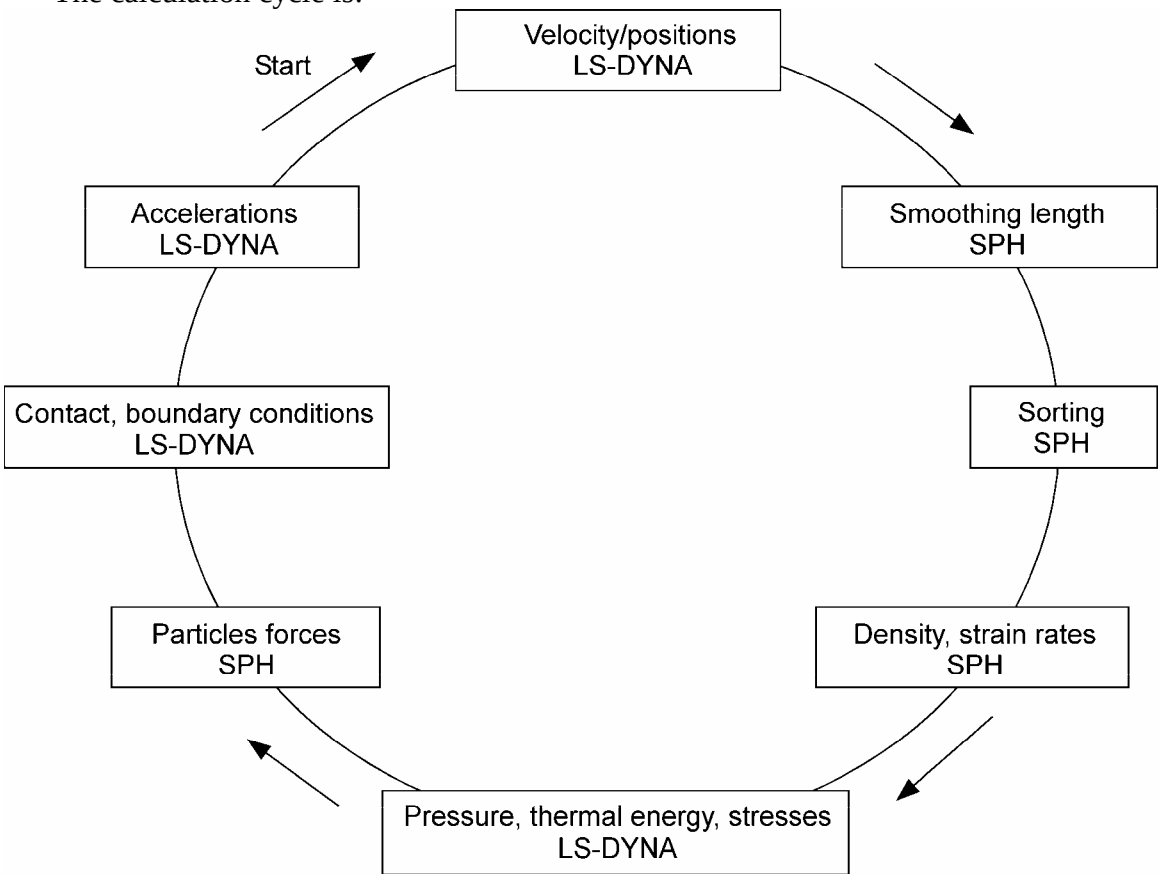
38.2.3 Time Integration

We use a simple and classical first-order scheme for integration. The time step is determined by the expression:

$$\delta t = C_{CFL} Min_i \left(\frac{h_i}{c_i + v_i} \right) \tag{38.24}$$

where the factor C_{CFL} is a numerical constant.

The calculation cycle is:



38.2.4 Initial Setups

Initially, we have a set of particles with two kinds of properties: physical properties and geometrical properties.

Physical Properties:

The mass, density, constitutive laws are defined in the ELEMENT_SPH and the PART cards.

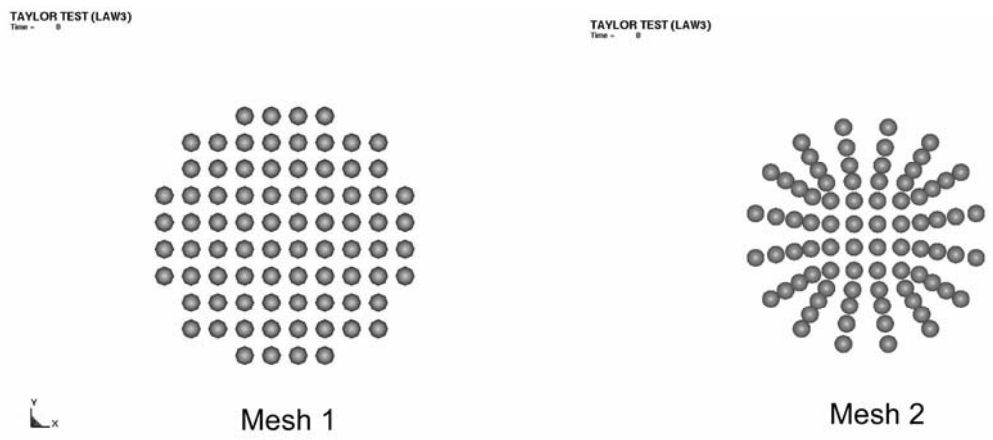
Geometrical Properties:

The geometrical properties of the model concern the way particles are initially placed. Two different parameters are to be fixed: Δx_i lengths and the CSLH coefficient.

These parameters are defined in the SECTION_SPH card.

A proper SPH mesh must satisfy the following conditions: it must be as regular as possible and must not contain too large variations.

For instance, if we consider a cylinder SPH mesh, we have at least two possibilities:



The mesh number 2 includes too many inter-particle distance discrepancies. Therefore, the first mesh, more uniform, is better.

Finite element coupling

Coupling finite elements and SPH elements is realized by using contact algorithms. Users can choose any “nodes_to_surface” contact type where the slave part is defined with SPH elements and the master part is defined with finite elements.

39. ELEMENT-FREE GALERKIN

Mesh-free methods, which construct the approximation entirely in terms of nodes, permit reduced restriction in the discretization of the problem domain and are less susceptible to distortion difficulties than finite elements. For a variety of engineering problems with extremely large deformation, moving boundaries or discontinuities, mesh-free methods are very attractive.

The two most commonly used approximation theories in mesh-free methods are the moving least-squares (MLS) approximation in the Element-free Galerkin (EFG) method [Belytschko et al. 1994], and the reproducing kernel (RK) approximation in the reproducing kernel particle method (RKPM) [Liu et al. 1995]. Since these two methods lead to an identical approximation when monomial basis functions are used, the MLS approximation is used as a basis to formulate the mesh-free discrete equations in this section.

39.1 Moving least-squares

The Element-free Galerkin method uses the moving least-squares approximation to construct the numerical discretization. The discrete MLS approximation of a function $u(\boldsymbol{x})$, denoted by $u^h(\boldsymbol{x})$, is constructed by a combination of the monomials as

$$u^h(\boldsymbol{x}) = \sum_{i=1}^{n} H_i(\boldsymbol{x}) b_i(\boldsymbol{x}) \equiv \boldsymbol{H}^T(\boldsymbol{x})\boldsymbol{b}(\boldsymbol{x}) \tag{39.1}$$

where n is the order of completeness in this approximation, the monomial $H_i(\boldsymbol{x})$ are basis functions, and $b_i(\boldsymbol{x})$ are the coefficients of the approximation.

The coefficients $b_i(\boldsymbol{x})$ at any point $\boldsymbol{x}$ are depending on the sampling points $\boldsymbol{x}_I$ that are collected by a weighting function $w_a(\boldsymbol{x} - \boldsymbol{x}_I)$. This weighting function is defined to have a compact support measured by 'a', i.e., the sub-domain over which it is nonzero is small relative to the rest of the domain. Each sub-domain $\Delta\Omega_I$ is associated with a node I. The most commonly used sub-domains are disks or balls. A typical numerical model is shown in Figure 1.

In this development, we employ the cubic B-spline kernel function as the weighting function:

$$w_a(\boldsymbol{x} - \boldsymbol{x}_I) = \begin{Bmatrix} \dfrac{2}{3} - 4\left(\dfrac{\|\boldsymbol{x}-\boldsymbol{x}_I\|}{a}\right)^2 + 4\left(\dfrac{\|\boldsymbol{x}-\boldsymbol{x}_I\|}{a}\right)^3 & \text{for } 0 \le \dfrac{\|\boldsymbol{x}-\boldsymbol{x}_I\|}{a} \le \dfrac{1}{2} \\ \dfrac{4}{3} - 4\left(\dfrac{\|\boldsymbol{x}-\boldsymbol{x}_I\|}{a}\right) + 4\left(\dfrac{\|\boldsymbol{x}-\boldsymbol{x}_I\|}{a}\right)^2 - \dfrac{4}{3}\left(\dfrac{\|\boldsymbol{x}-\boldsymbol{x}_I\|}{a}\right)^2 & \text{for } \dfrac{1}{2} < \dfrac{\|\boldsymbol{x}-\boldsymbol{x}_I\|}{a} \le 1 \\ 0 & \text{otherwise} \end{Bmatrix} \tag{39.2}$$

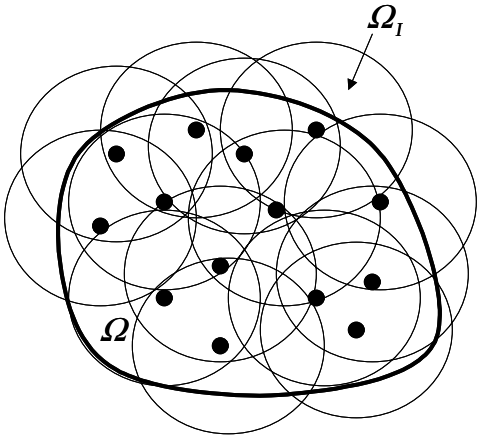


Figure 39.1. Graphical representation of mesh-free discretization

The moving least-squares technique consists in minimizing the weighted L_2-Norm

$$J = \sum_{I=1}^{NP} w_a(\boldsymbol{x} - \boldsymbol{x}_I)\left[\sum_{i=1}^{n} H_i(\boldsymbol{x}) b_i(\boldsymbol{x}) - u(\boldsymbol{x}_I)\right]^2 \tag{39.3}$$

where *NP* is the number of nodes within the support of $\boldsymbol{x}$ for which the weighting function $w_a(\boldsymbol{x} - \boldsymbol{x}_I) \neq 0$.

Equation (39.3) can be written in the form

$$J = (\boldsymbol{Hb} - \boldsymbol{u})^T \boldsymbol{W}_a(\boldsymbol{x})(\boldsymbol{Hb} - \boldsymbol{u}) \tag{39.4}$$

where

$$\boldsymbol{u}^T = (u_1, u_2, \cdots u_{NP}) \tag{39.5}$$

$$\boldsymbol{H} = \begin{bmatrix} \{\boldsymbol{H}(\boldsymbol{x}_1)\}^T \\ \cdots \\ \{\boldsymbol{H}(\boldsymbol{x}_{NP})\}^T \end{bmatrix} \tag{39.6}$$

$$\{\boldsymbol{H}(\boldsymbol{x}_i)\}^T = \{H_1(\boldsymbol{x}_i), \ldots H_n(\boldsymbol{x}_i)\} \tag{39.7}$$

$$\boldsymbol{W}_a = diag[w_a(\boldsymbol{x} - \boldsymbol{x}_1), \cdots, w_a(\boldsymbol{x} - \boldsymbol{x}_{NP})] \tag{39.8}$$

To find the coefficients $\boldsymbol{b}$ we obtain the extremum of *J* by

$$\partial J / \partial \boldsymbol{b} = \boldsymbol{M}^{[n]}(\boldsymbol{x})\boldsymbol{b}(\boldsymbol{x}) - \boldsymbol{B}(\boldsymbol{x})\boldsymbol{u} = 0 \tag{39.9}$$

where $\boldsymbol{M}^{[n]}(\boldsymbol{x})$ is called the moment matrix of $w_a(\boldsymbol{x} - \boldsymbol{x}_I)$ and is given by

$$\boldsymbol{M}^{[n]}(\boldsymbol{x}) = \boldsymbol{H}^T \boldsymbol{W}_a(\boldsymbol{x})\boldsymbol{H} \tag{39.10}$$

$$\boldsymbol{B}(\boldsymbol{x}) = \boldsymbol{H}^T \boldsymbol{W}_a(\boldsymbol{x}) \tag{39.11}$$

So we have

$$\boldsymbol{b}(\boldsymbol{x}) = \boldsymbol{M}^{[n]-1}(\boldsymbol{x})\boldsymbol{B}(\boldsymbol{x})\boldsymbol{u} \tag{39.12}$$

For $\boldsymbol{M}^{[n]}(\boldsymbol{x})$ to be invertible, the support of $w_a(\boldsymbol{x} - \boldsymbol{x})$ needs to be greater than a minimum size that is related to the order of basis functions.

Using the solution of Equations (39.1), (39.10), (39.11) and (39.12), the EFG approximation is obtained by

$$u^h(\boldsymbol{x}) = \sum_{I=1}^{NP} \Psi_I(\boldsymbol{x}) u_I \tag{39.13}$$

where the EFG shape functions $\Psi_I(\boldsymbol{x})$ are given by

$$\Psi_I(\boldsymbol{x}) = \boldsymbol{H}^T(\boldsymbol{x})\boldsymbol{M}^{[n]-1}(\boldsymbol{x})\boldsymbol{B}(\boldsymbol{x}) \tag{39.14}$$

and $\Psi_I(\boldsymbol{x})$ are *n*th-order complete, i.e.

$$\sum_{I=1}^{NP} \Psi_I(\boldsymbol{x}) x_{1I}^p x_{2I}^q = x_1^p x_2^q \quad for \quad p+q = 0, \cdots n \tag{39.15}$$

39.2 Integration constraint and strain smoothing

The convergence of the Galerkin method for a partial differential equation is determined by approximation for the unknowns and the numerical integration of the weak form. EFG shape functions with linear consistency can be obtained from MLS approximation with linear basis functions. The employment of linearly consistent mesh-free shape functions in the Galerkin approximation, however, does not guarantee a linear exactness in the solution of the Galerkin method. It has been shown by Chen et al. [2001] that two integration constraints are required for the linear exactness solution in the Galerkin approximation.

$$\sum_{L=1}^{NIT} \nabla \Psi_I(\hat{\boldsymbol{x}}_L) A_L = \boldsymbol{0} \quad for \ \{I : supp(\Psi_I) \cap \Gamma = 0\} \tag{39.16}$$

$$\sum_{L=1}^{NIT} \nabla \Psi_I(\hat{\boldsymbol{x}}_L) A_L = \sum_{L=1}^{NITh} \boldsymbol{n} \Psi_I(\tilde{\boldsymbol{x}}_L) s_L \quad for \ \{I : supp(\Psi_I) \cap \Gamma^h \neq 0\} \tag{39.17}$$

where Γ^h is the natural boundary, Γ is the total boundary, $\boldsymbol{n}$ is the surface normal on Γ^h, $\hat{\boldsymbol{x}}_L$ and A_L are the spatial co-ordinate and weight of the domain integration point, respectively, $\tilde{\boldsymbol{x}}_L$ and s_L are the spatial co-ordinate and weight of the domain of natural boundary integration

point, respectively, *NIT* is the number of integration points for domain integration and *NITh* is the number of integration points for natural boundary integration.

A strain smoothing method proposed by Chen and Wu [1998] as a regularization for material instabilities in strain localization was extended in their nodal integration method [Chen et al, 2001] to meet the integration constraints. Here, we adopt the similar concept for the domain integration. If starts with a strain smoothing at the representative domain of a Gauss point by

$$\tilde{\nabla} u_i^h(\boldsymbol{x}_L) = \frac{1}{A_L}\int_{\Omega_L} \nabla u_i^h(\boldsymbol{x}_L)\, d\Omega,\ A_L = \int_{\Omega_L} d\Omega \tag{39.18}$$

where Ω_L is a representative domain at each Guass point and $\tilde{\nabla}$ is the smoothed gradient operator. By applying divergence theorem to Equation (39.18) to yield

$$\tilde{\nabla} u_i^h(\boldsymbol{x}_L) = \frac{1}{A_L}\int_{\Gamma_L} \boldsymbol{n} u_i^h(\boldsymbol{x}_L)\, d\Gamma \tag{39.19}$$

where Γ_L is the boundary of the representative domain of Guass point *L*. Introducing EFG shape functions into Equation (39.20) yields

$$\tilde{\nabla} u_i^h(\boldsymbol{x}_L) = \sum_I \frac{1}{A_L}\int_{\Gamma_L} \Psi_I(\boldsymbol{x})\boldsymbol{n} d\Gamma \cdot d_{iI} \equiv \sum_I \tilde{\nabla}\Psi_I(\boldsymbol{x}_L)\cdot d_{iI} \tag{39.20}$$

It can be shown that the smoothed EFG shape function gradient $\tilde{\nabla}\Psi_I(\boldsymbol{x}_L)$ meets the integration constraints in Equations (39.16) and (39.17) regardless of the numerical integration employed.

39.3 Lagrangian strain smoothing for path-dependent problems

To avoid the tensile instability caused by the Eulerian kernel functions, the Lagrangian kernel functions are implemented in the current LS-DYNA.

To introduce the Lagrangian EFG shape function into the approximation of a path-dependent problem, the strain increment $\Delta u_{i,j}$ is computed by

$$\Delta u_{i,j} = \frac{\partial \Delta u_i}{\partial x_j} = \frac{\partial \Delta u_i}{\partial X_k} F_{kj}^{-1} = \Delta F_{ik} F_{kj}^{-1} \tag{39.21}$$

The strain smoothing of $\Delta u_{i,j}$ at a material point $\boldsymbol{x}_L$ is computed by

$$\Delta\tilde{u}_{i,j}(\boldsymbol{x}_L) = \Delta\tilde{F}_{ik}(\boldsymbol{x}_L)\tilde{F}_{kj}^{-1}(\boldsymbol{x}_L) \tag{39.22}$$

where $\tilde{F}_{ij}(\boldsymbol{x}_L)$ is the Langrangian strain smoothing of deformation gradient and is given by

$$\tilde{F}_{ij}(\boldsymbol{x}_L) = \frac{1}{A_L}\int_{\Gamma_L} u_i^h N_j d\Gamma + \delta_{ij} \tag{39.23}$$

39.4 Galerkin approximation for explicit dynamic computation

The strong form of the initial/boundary value problem for elasto-dynamics is as follows:

$$\rho\ddot{\boldsymbol{u}} = \nabla\cdot\boldsymbol{\sigma} + \boldsymbol{f}_b \text{ in } \Omega \tag{39.24}$$

with the divergence operator ∇, the body force $\boldsymbol{f}_b$, mass density ρ ,and with the boundary conditions:

$$\begin{aligned} \boldsymbol{u} &= \boldsymbol{u}_0 \text{ on } \Gamma_u \\ \boldsymbol{\sigma}\cdot\boldsymbol{n} &= \boldsymbol{h} \text{ on } \Gamma_h \end{aligned} \tag{39.25}$$

and initial conditions

$$\begin{aligned} \boldsymbol{u}(\boldsymbol{X},0) &= \boldsymbol{u}^0(\boldsymbol{X}) \\ \dot{\boldsymbol{u}}(\boldsymbol{X},0) &= \dot{\boldsymbol{u}}^0(\boldsymbol{X}) \end{aligned} \tag{39.26}$$

To introduce the Lagrangian strain smoothing formulation into the Galerkin approximation, an assumed strain method is employed. The corresponding weak form becomes:

$$\int_{\Omega_x}\rho\delta\boldsymbol{u}\cdot\ddot{\boldsymbol{u}}d\Omega + \int_{\Omega_x}\delta\tilde{\boldsymbol{\varepsilon}} : \boldsymbol{\sigma}d\Omega = \int_{\Omega_x}\delta\boldsymbol{u}\cdot\boldsymbol{f}_b\, d\Omega + \int_{\Gamma_h}\delta\boldsymbol{u}\cdot\boldsymbol{h}\, d\Gamma \tag{39.27}$$

Following the derivation for explicit time integration, the equations to be solved have the form

$$\delta\boldsymbol{u}^T\boldsymbol{M}\ddot{\boldsymbol{u}} = \delta\boldsymbol{u}^T\boldsymbol{R} \tag{39.28}$$

where

$$\begin{aligned} \ddot{\boldsymbol{u}}_I &= [\ddot{d}_{1I}, \ddot{d}_{2I}, \ddot{d}_{3I}]^T \\ M_{IJ} &= \int_{\Omega_x}\rho\Psi_I(\boldsymbol{x})\Psi_J(\boldsymbol{x})d\Omega = \int_{\Omega_X}\rho^0\Psi_I(\boldsymbol{X})\Psi_J(\boldsymbol{X})d\Omega \\ \boldsymbol{R}_I &= \int_{\Omega_x}\tilde{\boldsymbol{B}}_I^T(\boldsymbol{x})\cdot\boldsymbol{\sigma}(\tilde{\boldsymbol{F}})d\Omega - [\Psi_I(\boldsymbol{x})\boldsymbol{h}]\big|_{\Gamma_h} - \int_{\Omega_x}\Psi_I(\boldsymbol{x})\boldsymbol{f}_b d\Omega \end{aligned} \tag{39.29}$$

where $\tilde{\boldsymbol{B}}_I^T(\boldsymbol{x})$ is the smoothed gradient matrix obtained from Equation (39.22), d_{iI} is the coefficient of the approximation or the "generalized" displacement.

39.5 Imposition of essential boundary condition

In general, mesh-free shape functions Ψ_I do not possess Kronecker delta properties of the standard FEM shape functions, i.e.

$$\Psi_I(\boldsymbol{x}_J) \neq \delta_{IJ} \tag{39.30}$$

This is because, in general, the mesh-free shape functions are not interpolation functions. As a result, a special treatment is required to enforce essential boundary conditions. There are many techniques for mesh-free methods to impose the essential boundary condition. Here, we adopt the transformation method as originally proposed for the RKPM method by Chen et al. [1996].

Therefore, to impose the essential boundary conditions using kinematically admissible mesh-free shape functions by the transformation method, Equation (39.28) can be written as

$$\delta\hat{\boldsymbol{u}}^T \hat{\boldsymbol{M}}\ddot{\hat{\boldsymbol{u}}} = \delta\hat{\boldsymbol{u}}^T \hat{\boldsymbol{F}}^{int} \tag{39.31}$$

where

$$\hat{\boldsymbol{u}} = \boldsymbol{A}\boldsymbol{u}; A_{IJ} = \Psi_J\left(X_I\right) \tag{39.32}$$

or

$$\boldsymbol{u} = \boldsymbol{A}^{-1}\hat{\boldsymbol{u}} \tag{39.33}$$

and

$$\hat{\boldsymbol{M}} = \boldsymbol{A}^{-T}\boldsymbol{M}\boldsymbol{A}^{-1}; \hat{\boldsymbol{F}}^{int} = \boldsymbol{A}^{-T}\boldsymbol{F}^{int} \tag{39.34}$$

A mixed transformation method [Chen et al. 2000] is also considered as an alternative to impose the essential boundary conditions. The mixed transformation method is an improved transformation method that the coordinate transformation is only applied for the degrees of freedom associated with the essential and contact boundaries.

The nodes are partitioned into three groups: a boundary group G^{B_1} which contains all the nodes subjected to kinematic constraints; group G^{B_2} which contains all the nodes whose kernel supports cover nodes in group G^{B_1}; and internal group G^I which contains the rest of nodes. Nodes numbers are re-arranged in the following order in the generalized displacement vector:

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}^{B_1} \\ \boldsymbol{u}^{B_2} \\ \boldsymbol{u}^{I} \end{bmatrix} \tag{39.35}$$

where $\boldsymbol{u}^{B_1}, \boldsymbol{u}^{B_2}$ and $\boldsymbol{u}^I$ are the generalized displacement vectors associated with groups G^{B_1}, G^{B_2} and G^I respectively. The transformation in Equation (39.32) is also re-arranged as

$$\hat{\boldsymbol{u}} = \begin{bmatrix} \hat{\boldsymbol{u}}^B \\ \hat{\boldsymbol{u}}^I \end{bmatrix} \begin{bmatrix} \boldsymbol{\Lambda}^{BB} & \boldsymbol{\Lambda}^{BI} \\ \boldsymbol{\Lambda}^{IB} & \boldsymbol{\Lambda}^{II} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^B \\ \boldsymbol{u}^I \end{bmatrix} \equiv \hat{\boldsymbol{\Lambda}}\boldsymbol{u} \tag{39.36}$$

where

$$\hat{\boldsymbol{u}}^B = \begin{bmatrix} \hat{\boldsymbol{u}}^{B_1} \\ \hat{\boldsymbol{u}}^{B_2} \end{bmatrix}; \boldsymbol{u}^B = \begin{bmatrix} \boldsymbol{u}^{B_1} \\ \boldsymbol{u}^{B_2} \end{bmatrix}; \boldsymbol{\Lambda}^{BB} = \begin{bmatrix} \boldsymbol{\Lambda}^{B_1B_1} & \boldsymbol{\Lambda}^{B_1B_2} \\ \boldsymbol{\Lambda}^{B_2B_1} & \boldsymbol{\Lambda}^{B_2B_2} \end{bmatrix}; \boldsymbol{\Lambda}^{BI} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{\Lambda}^{B_2I} \end{bmatrix}; \boldsymbol{\Lambda}^{IB} = \begin{bmatrix} \boldsymbol{\Lambda}^{IB_1} & \boldsymbol{\Lambda}^{IB_2} \end{bmatrix} \tag{39.37}$$

Here, we introduce a mixed displacement vector $\boldsymbol{u}^*$,

$$\boldsymbol{u}^* = \begin{bmatrix} \hat{\boldsymbol{u}}^B \\ \boldsymbol{u}^I \end{bmatrix} \begin{bmatrix} \boldsymbol{\Lambda}^{BB} & \boldsymbol{\Lambda}^{BI} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^B \\ \boldsymbol{u}^I \end{bmatrix} \equiv \boldsymbol{\Lambda}^* \boldsymbol{u} \tag{39.38}$$

and $\boldsymbol{\Lambda}^*$ and its inverse are:

$$\boldsymbol{\Lambda}^* = \begin{bmatrix} \boldsymbol{\Lambda}^{BB} & \boldsymbol{\Lambda}^{BI} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}; \boldsymbol{\Lambda}^{*^{-1}} = \begin{bmatrix} \boldsymbol{\Lambda}^{BB^{-1}} & -\boldsymbol{\Lambda}^{BB^{-1}} \boldsymbol{\Lambda}^{BI} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \tag{39.39}$$

Only the inversion of $\boldsymbol{\Lambda}^{BB}$ is required in Equation (39.39).

Using the mixed coordinates in Equation (39.38), the transformed discrete Equation (39.31) becomes

$$\delta \boldsymbol{u}^{*^T} \boldsymbol{M}^* \ddot{\boldsymbol{u}}^* = \delta \boldsymbol{u}^{*^T} \boldsymbol{R}^* \tag{39.40}$$

where

$$\boldsymbol{M}^* = \boldsymbol{A}^{*^{-T}} \boldsymbol{M} \boldsymbol{A}^{*^{-1}}; \boldsymbol{R}^* = \boldsymbol{A}^{*^{-T}} \boldsymbol{R} \tag{39.41}$$

The computation in Equations (39.41) is much less intensive than that in Equation (39.31), especially when the number of boundary and contact nodes is much smaller than the number of interior nodes.

39.6 Mesh-free Shell

The extension of explicit mesh-free solid analysis to shell analysis is described in this section. Two projection methods are developed to generate the shell mid-surface using the moving-least-squares approximations. A co-rotational, updated Lagrangian procedure is adopted to handle arbitrarily large rotations with moderate strain responses of the shell structures. A local boundary integration method in conjunction with the selective reduced integration method is introduced to enforce the linear exactness and relieve shear locking.

39.6.1 Mesh-free Shell Surface Representation

Surface reconstruction from disorganized nodes is very challenging in three dimensions. The problem is ill posed, i.e., there is no unique solution. Lancaster et al. [1981] first proposed a fast surface reconstruction using moving least squares method. Their approach was then applied to the computational mechanics under the name ‘mesh-free method’. Implicitly, the mesh-free method uses a combination of smooth basis functions (primitives) to find a scalar function such that all data nodes are close to an iso-contour of that scalar function in a global sense. In reality, the shell surface construction using the 3D mesh-free method is inadequate. This is because the topology of the real surface can be very complicated in three dimensions. Without the information on the ordering or connectivity of nodes, the reconstructed surface will not be able to represent shell intersections, exterior boundaries and shape corners.

In our development of mesh-free shells, we assume that a shell surface is described by a finite element mesh. This can be easily accomplished by converting a part of shell finite elements into mesh-free zone. With the connectivity of nodes provided by the finite element mesh, a shell surface can be reconstructed with mesh-free interpolation from the nodal positions

$$\bar{\mathbf{x}} = \tilde{\Psi}_I(\mathbf{X})\mathbf{x}_I \tag{39.42}$$

where $\mathbf{x}_I$ is the position vector of the finite element node on the shell surface and $\tilde{\Psi}_I(\mathbf{X})$ is the mesh-free shape function. In the above surface representation, a 3D arbitrary shell surface needs to be projected to a 2D plane. Two approaches for the projection of mesh-free shell surface are used:

(1) Global parametric representation: The whole shell surface is projected to a parametric plane and the global parametric coordinates are obtained with a parameterization algorithm from the patch of finite elements.

(2) Local projection representation: A local area of the shell is projected to a plane based on the existing element where the evaluated point is located.

Global parametric approach

In the global approach, a mesh-free zone with a patch of finite elements is mapped onto a parametric plane with an angle-based triangular flattening algorithm [Sheffer and de Sturler 2001], (see Figure 39.2). The idea of this algorithm is to compute a projection that minimizes the distortion of the FE mesh angles. The mesh-free shape functions are defined in this parametric domain and given by

$$\tilde{\Psi}_I(\mathbf{X}) = \tilde{\Psi}_I(\xi,\eta) \tag{39.43}$$

where (ξ,η) is the parametric coordinates corresponding to a point **X**.

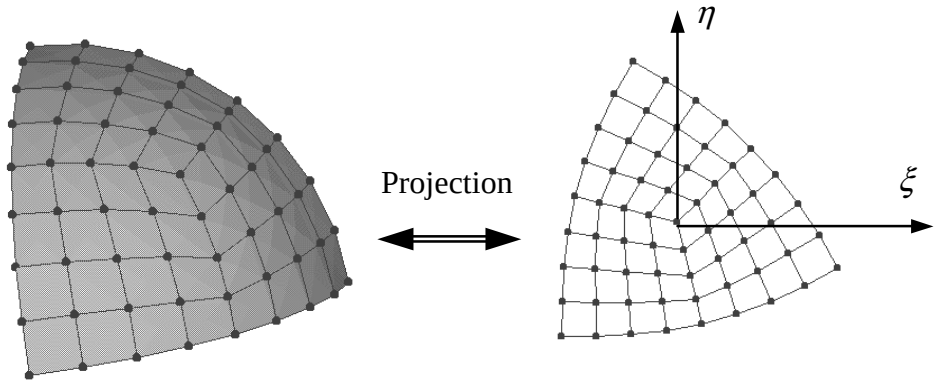


Figure 39.2. Mesh-free shell global approach

Local projection approach

Different from the parameterization algorithm that constructs the surface globally, we reconstruct the surface locally by projecting the surrounding nodes onto one element. In the local projection method, nodes in elements neighboring the element where the evaluated point is located (for example, the element i in Figure 39.3) are projected onto the plane which the element defines (the "M-plane" in Figure 39.3). In this figure, $(\hat{x}, \hat{y}, \hat{z})_i$ is a local system defined for each projected plane and $(\bar{x}, \bar{y}, \bar{z})_I$ is a nodal coordinate system defined for each node where $\bar{z}$ is the initial averaged normal direction.

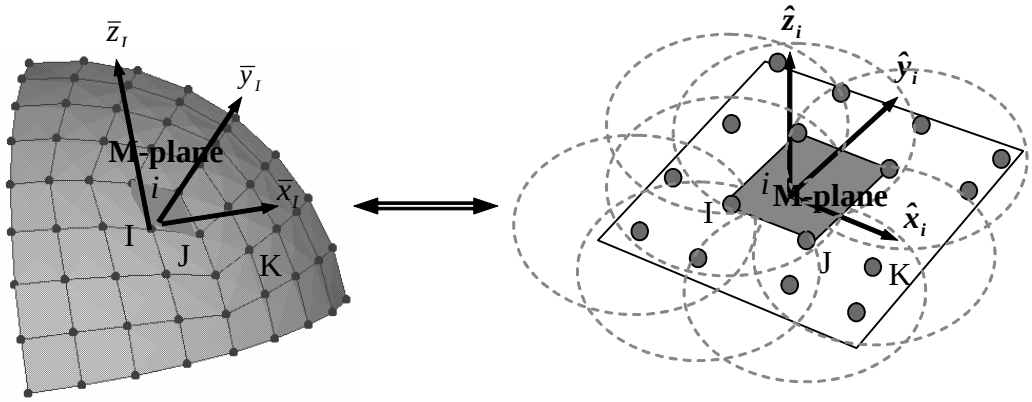


Figure 39.3. Mesh-free shell local projection

The mesh-free shape functions are then defined with those locally projected coordinates of the nodes

$$\Psi_I(\mathbf{X}) = \Psi_I(\hat{x}, \hat{y}) \tag{39.44}$$

However, the shape functions obtained directly above are non-conforming, i.e.

$$\Psi_I(\mathbf{X}_J)\big|_{\text{M-plane}} \neq \Psi_I(\mathbf{X}_J)\big|_{\text{N-plane}} \tag{39.45}$$

When the shell structure degenerates to a plate, the constant stress condition cannot be recovered. To remedy this problem, an area-weighed smoothing across different projected planes is used to obtain the conforming shape functions that are given by

$$\tilde{\Psi}_I(\mathbf{X}) = \tilde{\Psi}_I(\hat{x}, \hat{y}) = \frac{\sum_{i=1}^{NIE} \Psi_I(\hat{x}_i, \hat{y}_i) A_i}{\sum_{i=1}^{NIE} A_i} \tag{39.46}$$

where *NIE* is the number of surrounding projected planes that can be evaluated at point **X**, A_i is the area of the element *i*, and $(\hat{x}_i, \hat{y}_i)$ is the local coordinates of point **X** in the projected plane *i*.

With this smoothing technique, we can prove that the modified shape functions satisfy at least the *partition of unity* property in the general shell problems. This property is important for the shell formulation to preserve the rigid-body translation.

When the shell degenerates to a plate, we can also prove that the shape functions obtained from this smoothing technique will meet the *n*-th order completeness condition as

$$\sum_{I=1}^{NP} \tilde{\Psi}_I(\mathbf{X}) X_{1I}^i X_{2I}^j X_{3I}^k = X_1^i X_2^j X_3^k, \qquad i+j+k=n \tag{39.47}$$

This is a necessary condition for the plate to pass the constant bending patch test.

39.6.2 Updated Lagrangian Formulation and Co-rotational Procedure

The mesh-free shell formulation is based on the Mindlin-Reissner plate theory, thus the geometry and kinematical fields of the shell can be described with the reference surface and fiber direction. The modified Mindlin-Reissner assumption requires that the motion and displacement of the shell are linear in the fiber direction. Assume that the reference surface is the mid-surface of the shell, the global coordinates and displacements at an arbitrary point within the shell body are given by

$$\mathbf{x} = \bar{\mathbf{x}} + \zeta \frac{h}{2} \mathbf{V}_3 \tag{39.48}$$

$$\mathbf{u} = \bar{\mathbf{u}} + \zeta \frac{h}{2} \mathbf{U} \tag{39.49}$$

where $\bar{\mathbf{x}}$ and $\bar{\mathbf{u}}$ are the position vector and displacement of the reference surface, respectively. $\mathbf{V}_3$ is the fiber director and $\mathbf{U}$ is the displacement resulting from the fiber rotation (see Figures 39.4 and 39.5). *h* is the length of the fiber.

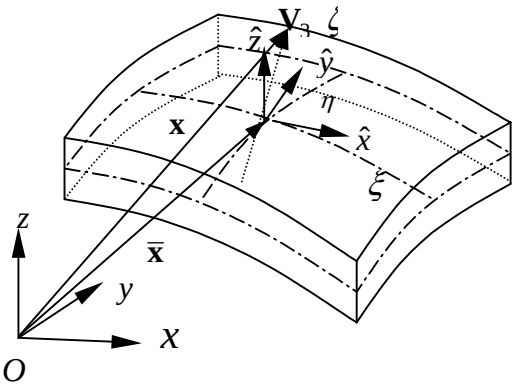


Figure 39.4. Geometry of a shell.

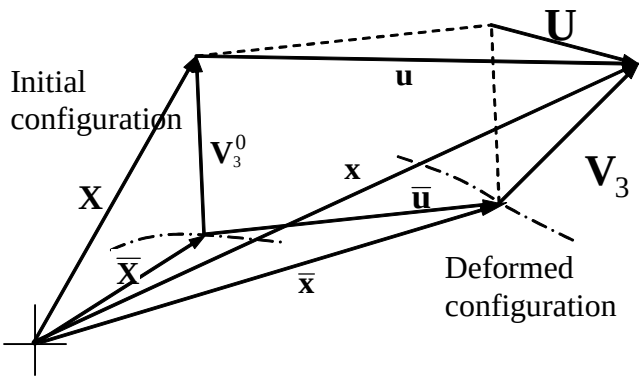


Figure 39.5. Deformation of a shell.

With the mesh-free approximation, the motion and displacements are given by

$$\mathbf{x}(\xi,\eta,\zeta) = \bar{\mathbf{x}}(\xi,\eta) + \mathbf{V}(\xi,\eta,\zeta) \approx \sum_{I=1}^{NP} \tilde{\Psi}_I(\xi,\eta)\mathbf{x}_I + \sum_{I=1}^{NP} \tilde{\Psi}_I(\xi,\eta)\frac{\zeta h_I}{2}\mathbf{V}_{3I} \tag{39.50}$$

$$\mathbf{u}(\xi,\eta,\zeta) = \bar{\mathbf{u}}(\xi,\eta) + \mathbf{U}(\xi,\eta,\zeta) \approx \sum_{I=1}^{NP} \tilde{\Psi}_I(\xi,\eta)\mathbf{u}_I + \sum_{I=1}^{NP} \tilde{\Psi}_I(\xi,\eta)\frac{\zeta h_I}{2}\begin{bmatrix}-\mathbf{V}_{2I} & \mathbf{V}_{1I}\end{bmatrix}\begin{Bmatrix}\alpha_I \\ \beta_I\end{Bmatrix} \tag{39.51}$$

where $\mathbf{x}_I$ and $\mathbf{u}_I$ are the global coordinates and displacements at mesh-free node I, respectively. $\mathbf{V}_{3I}$ is the unit vector of the fiber director and $\mathbf{V}_{1I}$, $\mathbf{V}_{2I}$ are the base vectors of the nodal coordinate system at node I. α_I and β_I are the rotations of the director vector $\mathbf{V}_{3I}$ about the $\mathbf{V}_{1I}$ and $\mathbf{V}_{2I}$ axes. h_I is the thickness. The variables with a superscripted bar refer to the shell mid-surface. $\tilde{\Psi}_I$ is the 2D mesh-free shape functions constructed based on one of the two mesh-free surface representations described in the previous section, with (ξ,η) either the parametric coordinates or local coordinates of the evaluated point.

The local co-rotational coordinate system ($\hat{x}$, $\hat{y}$, $\hat{z}$) is defined at each integration point on the shell reference surface, with $\hat{x}$ and $\hat{y}$ tangent to the reference surface and $\hat{z}$ in the thickness direction (see Figure 39.6). The base vectors are given as

$$\hat{\mathbf{e}}_1 = \frac{\mathbf{x}_{,\xi}}{\|\mathbf{x}_{,\xi}\|}, \qquad \hat{\mathbf{e}}_3 = \frac{\mathbf{x}_{,\xi}\times\mathbf{x}_{,\eta}}{\|\mathbf{x}_{,\xi}\times\mathbf{x}_{,\eta}\|}, \qquad \hat{\mathbf{e}}_2 = \hat{\mathbf{e}}_3\times\hat{\mathbf{e}}_1 \tag{39.52}$$

In order to describe the fiber rotations of a mesh-free node in a shell, we introduce a nodal coordinate system whose three base vectors are $\mathbf{V}_1$, $\mathbf{V}_2$ and $\mathbf{V}_3$, see Figure 39.6, where $\mathbf{V}_3$ is the fiber director at the node and $\mathbf{V}_1$, $\mathbf{V}_2$ are defined as follows

$$\mathbf{V}_1 = \frac{\hat{\mathbf{x}}\times\mathbf{V}_3}{|\hat{\mathbf{x}}\times\mathbf{V}_3|}, \qquad \mathbf{V}_2 = \mathbf{V}_3\times\mathbf{V}_1 \tag{39.53}$$

The rotation of the fiber director is then obtained from the global rotations:

$$\begin{Bmatrix}\alpha \\ \beta\end{Bmatrix} = \begin{bmatrix}\mathbf{V}_1^T \\ \mathbf{V}_2^T\end{bmatrix}\Delta\boldsymbol{\theta}, \qquad \Delta\boldsymbol{\theta} = \begin{bmatrix}\Delta\theta_1 & \Delta\theta_2 & \Delta\theta_3\end{bmatrix}^T \tag{39.54}$$

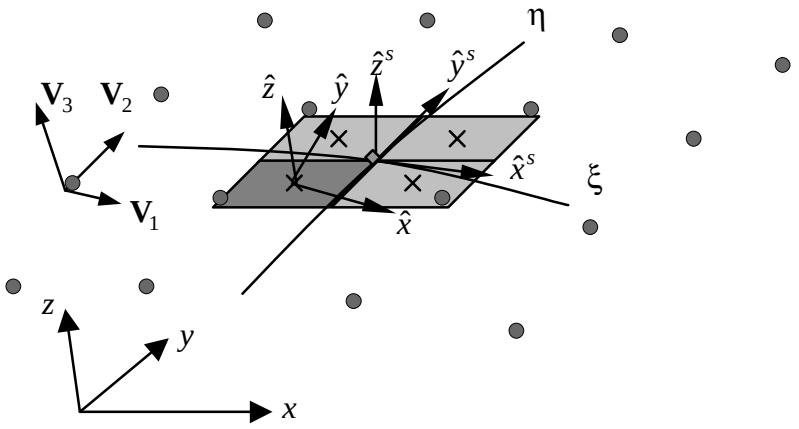


Figure 39.6. Local co-rotational and nodal coordinate systems.

In the local co-rotational coordinate system, the motion and displacements are approximated by the mesh-free shape functions

$$\hat{x}_i = \sum_{I=1}^{NP} \tilde{\Psi}_I \hat{x}_{iI} + \zeta \sum_{I=1}^{NP} \tilde{\Psi}_I \frac{h_I}{2} \hat{V}_{3iI} \tag{39.55}$$

$$\hat{u}_i = \sum_{I=1}^{NP} \tilde{\Psi}_I \hat{u}_{iI} + \zeta \sum_{I=1}^{NP} \tilde{\Psi}_I \frac{h_I}{2} \begin{bmatrix} -\hat{V}_{2iI} & \hat{V}_{1iI} \end{bmatrix} \begin{Bmatrix} \alpha_I \\ \beta_I \end{Bmatrix} \tag{39.56}$$

The Lagrangian smoothed strains [Chen et al. 2001b] are given by

$$\tilde{\boldsymbol{\varepsilon}}^m = \sum_I \tilde{\mathbf{B}}_I^m \hat{\mathbf{d}}_I, \quad \tilde{\boldsymbol{\varepsilon}}^b = \zeta \sum_I \tilde{\mathbf{B}}_I^b \hat{\mathbf{d}}_I, \quad \tilde{\boldsymbol{\varepsilon}}^s = \sum_I \tilde{\mathbf{B}}_I^s \hat{\mathbf{d}}_I \tag{39.57}$$

where the smoothed strain operators are calculated by averaging the consistent strain operators over an area around the evaluated point

$$\tilde{\mathbf{B}}_I^{\mathrm{m}}(\mathbf{x}_l) = \frac{1}{A_l} \int_{\Omega_l} \hat{\mathbf{B}}_I^{\mathrm{m}} dA, \qquad \tilde{\mathbf{B}}_I^{\mathrm{b}}(\mathbf{x}_l) = \frac{1}{A_l} \int_{\Omega_l} \hat{\mathbf{B}}_I^{\mathrm{b}} dA, \qquad \tilde{\mathbf{B}}_I^{\mathrm{s}}(\mathbf{x}_L) = \frac{1}{A_L} \int_{\Omega_L} \hat{\mathbf{B}}_I^{\mathrm{s}} dA \tag{39.58}$$

with

$$\hat{\mathbf{B}}_I^{\mathrm{m}} = \begin{bmatrix} \tilde{\Psi}_{I,x} & 0 & 0 & -J_{13}^{-1}\tilde{\Psi}_I \frac{h_I}{2}\hat{V}_{2xI} & J_{13}^{-1}\tilde{\Psi}_I \frac{h_I}{2}\hat{V}_{1xI} \\ 0 & \tilde{\Psi}_{I,y} & 0 & -J_{23}^{-1}\tilde{\Psi}_I \frac{h_I}{2}\hat{V}_{2yI} & J_{23}^{-1}\tilde{\Psi}_I \frac{h_I}{2}\hat{V}_{1yI} \\ \tilde{\Psi}_{I,y} & \tilde{\Psi}_{I,x} & 0 & -J_{23}^{-1}\tilde{\Psi}_I \frac{h_I}{2}\hat{V}_{2xI} - J_{13}^{-1}\tilde{\Psi}_I \frac{h_I}{2}\hat{V}_{2yI} & J_{23}^{-1}\tilde{\Psi}_I \frac{h_I}{2}\hat{V}_{1xI} + J_{13}^{-1}\tilde{\Psi}_I \frac{h_I}{2}\hat{V}_{1yI} \end{bmatrix} \tag{39.59a}$$

$$\hat{\mathbf{B}}_I^{\mathrm{b}} = \begin{bmatrix} 0 & 0 & 0 & -\tilde{\Psi}_{I,x}\frac{h_I}{2}\hat{V}_{2xI} & \tilde{\Psi}_{I,x}\frac{h_I}{2}\hat{V}_{1xI} \\ 0 & 0 & 0 & -\tilde{\Psi}_{I,y}\frac{h_I}{2}\hat{V}_{2yI} & \tilde{\Psi}_{I,y}\frac{h_I}{2}\hat{V}_{1yI} \\ 0 & 0 & 0 & -\tilde{\Psi}_{I,y}\frac{h_I}{2}\hat{V}_{2xI} - \tilde{\Psi}_{I,x}\frac{h_I}{2}\hat{V}_{2yI} & \tilde{\Psi}_{I,y}\frac{h_I}{2}\hat{V}_{1xI} + \tilde{\Psi}_{I,x}\frac{h_I}{2}\hat{V}_{1yI} \end{bmatrix} \tag{39.59b}$$

$$\hat{\mathbf{B}}_I^{\mathrm{s}} = \begin{bmatrix} 0 & 0 & \tilde{\Psi}_{I,y} & -J_{33}^{-1}\tilde{\Psi}_I\frac{h_I}{2}\hat{V}_{2yI} - J_{23}^{-1}\tilde{\Psi}_I\frac{h_I}{2}\hat{V}_{2zI} & J_{33}^{-1}\tilde{\Psi}_I\frac{h_I}{2}\hat{V}_{1yI} + J_{23}^{-1}\tilde{\Psi}_I\frac{h_I}{2}\hat{V}_{1zI} \\ 0 & 0 & \tilde{\Psi}_{I,x} & -J_{33}^{-1}\tilde{\Psi}_I\frac{h_I}{2}\hat{V}_{2xI} - J_{13}^{-1}\tilde{\Psi}_I\frac{h_I}{2}\hat{V}_{2zI} & J_{33}^{-1}\tilde{\Psi}_I\frac{h_I}{2}\hat{V}_{1xI} + J_{13}^{-1}\tilde{\Psi}_I\frac{h_I}{2}\hat{V}_{1zI} \end{bmatrix} \tag{39.59c}$$

and $\mathbf{J}^{-1}$ is the inverse of the Jacobian matrix at the integration point. The local degrees-of-freedom are

$$\hat{\mathbf{d}}_I = \begin{bmatrix} \hat{u}_{xI} & \hat{u}_{yI} & \hat{u}_{zI} & \alpha_I & \beta_I \end{bmatrix}^T \tag{39.60}$$

The internal nodal force vector is

$$\hat{\mathbf{F}}_I^{\mathrm{int}} = \int_\Omega \tilde{\mathbf{B}}_I^{\mathrm{m}T}\hat{\boldsymbol{\sigma}}d\Omega + \int_\Omega \zeta\tilde{\mathbf{B}}_I^{\mathrm{b}T}\hat{\boldsymbol{\sigma}}d\Omega + \int_\Omega \tilde{\mathbf{B}}_I^{\mathrm{s}T}\hat{\boldsymbol{\sigma}}d\Omega \tag{39.61}$$

The above integrals are calculated with the local boundary integration method. Each background finite element is divided into four integration zones, shown as Ω_l in Figure 39.7. In order to avoid shear locking in the analysis of thin shells, the shear term (third term in Eq. (39.61)), should be under-integrated by using one integration zone in each background element (Ω_L in Figure 39.7). Accordingly, the co-rotational coordinate systems are defined separately at the center of each integration zone, as shown in Figure 39.6.

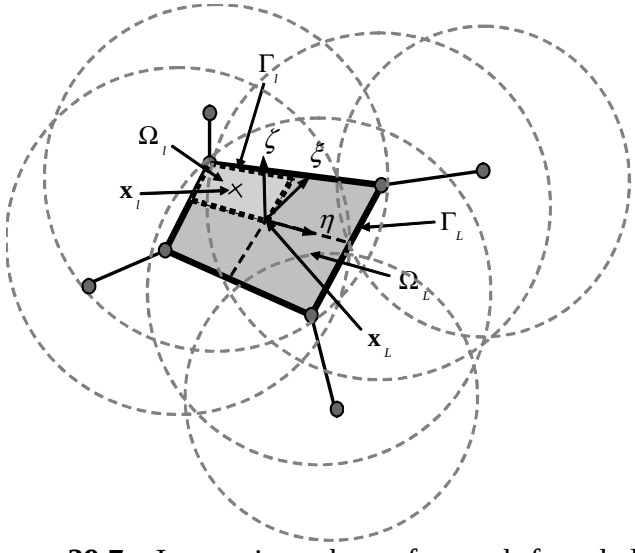


Figure 39.7. Integration scheme for mesh-free shells.

The use of the updated Lagrangian formulation implies that the reference coordinate system is defined by the co-rotational system in the configuration at time *t*. Therefore, the local nodal force and displacement vectors referred to this coordinate system must be transformed to the global coordinate system prior to assemblage.

40. LINEAR SHELLS

40.1 Shells for Linear Analysis

It is common to construct elements for linear analysis by the superimposition of a plate and a membrane element. If the base plates and membrane elements involve only three translational degrees-of-freedom and two in-plane rotational degrees-of-freedom, the resulting element then contains 5 degrees-of-freedom per node since there is an unconstrained rotational degree-of-freedom normal to the mid surface of the shell. This unconstrained mode can cause problems when linking the shell to other elements such as beam elements in three-dimensional space. For this reason, the linear elements in LS-DYNA are based on published formulations that include a drilling degree-of-freedom, which is added to the membrane part of the element to form a 24 degree-of-freedom shell element. These elements pass all patch tests, have 6 rigid body modes, and have no spurious mechanisms.

40.2 Wilson's Shell (element #20)

This quadrilateral element is constructed as described above and is discussed in more detail by Wilson [2000]. The triangular element, which is an 18 degree-of-freedom complement to the quadrilateral elements, follows the same procedure. In a linear analysis in LS-DYNA, automatic sorting is invoked if a mesh has both quadrilateral and triangular elements within a single part ID. This sorting ensures the proper treatment of triangles.

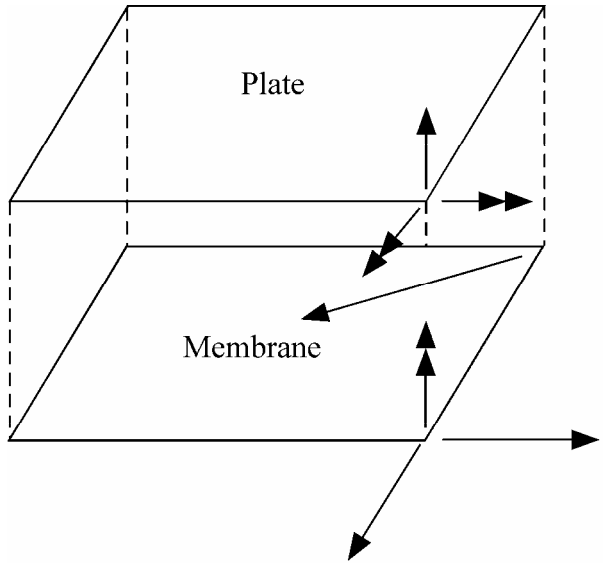


Figure 40.1. Shell as assembly of plate and membrane elements

40.2.1 Plate Element

The 4 node quadrilateral plate element is based on the 8 node, quadratic quadrilateral plate element, which has 16 rotational degrees-of-freedom, i.e., two per nodal point. The implementation in LS-DYNA directly follows the textbook by Wilson [2000] where the complete details of the element are provided. A condensed overview is given here. The shell theory makes the following assumptions:

- The fiber remains straight and inextensible

- The normal stress in the thickness direction is zero

The local x and y rotations of the shell are interpolated from the equations:

$$\begin{aligned}\theta_x(r,s) &= \sum_{i=1}^{4} N_i(r,s)\theta_{xi} + \sum_{i=5}^{8} N_i(r,s)\Delta\theta_{xi} \\ \theta_y(r,s) &= \sum_{i=1}^{4} N_i(r,s)\theta_{yi} + \sum_{i=5}^{8} N_i(r,s)\Delta\theta_{yi}\end{aligned} \tag{40.1}$$

where nodes 5-8 are at the mid side of the element. The interpolation functions are given by

$$\begin{aligned} N_1 &= \frac{1}{4}(1-r)(1-s) & N_5 &= \frac{1}{2}(1-r^2)(1-s) \\ N_2 &= \frac{1}{4}(1+r)(1-s) & N_6 &= \frac{1}{2}(1+r)(1-s^2) \\ N_3 &= \frac{1}{4}(1+r)(1+s) & N_7 &= \frac{1}{2}(1-r^2)(1+s) \\ N_4 &= \frac{1}{4}(1-r)(1+s) & N_8 &= \frac{1}{2}(1-r)(1-s^2) \end{aligned} \tag{40.2}$$

In his formulation, Wilson resolves the rotation of the mid side node into tangential and normal components relative to the shell edges. The tangential component is set to zero leaving the normal component as the unknown, which reduces the rotational degrees-of-freedom from 16 to 12, see Figure 40.2.

$$\begin{aligned}\Delta\theta_x &= \sin\alpha_{ij}\Delta\theta_{ij} \\ \Delta\theta_y &= -\cos\alpha_{ij}\Delta\theta_{ij}\end{aligned} \tag{40.3}$$

$$\begin{aligned}\theta_x(r,s) &= \sum_{i=1}^{4} N_i(r,s)\theta_{xi} + \sum_{i=5}^{8} M_{xi}(r,s)\Delta\theta_i \\ \theta_y(r,s) &= \sum_{i=1}^{4} N_i(r,s)\theta_{yi} + \sum_{i=5}^{8} M_{yi}(r,s)\Delta\theta_i\end{aligned} \tag{40.4}$$

Ultimately, the 4 mid side rotations are eliminated by using static condensation, a procedure that makes this shell very costly if used in explicit calculations.

The local x and y displacements relative to the mid surface are functions of the z-coordinate and rotations:

$$\begin{aligned}u_x(r,s) &= z\theta_y(r,s) \\ u_y(r,s) &= -z\theta_x(r,s)\end{aligned} \tag{40.5}$$

Wilson shows, where it is assumed that the normal displacement along each side is cubic, that the transverse shear strain along each side is given by,

$$\gamma_{ij} = \frac{1}{L}\left(u_{zj} - u_{zi}\right) - \frac{1}{2}\left(\theta_i + \theta_j\right) - \frac{2}{3}\Delta\theta_{ij} \tag{40.6}$$

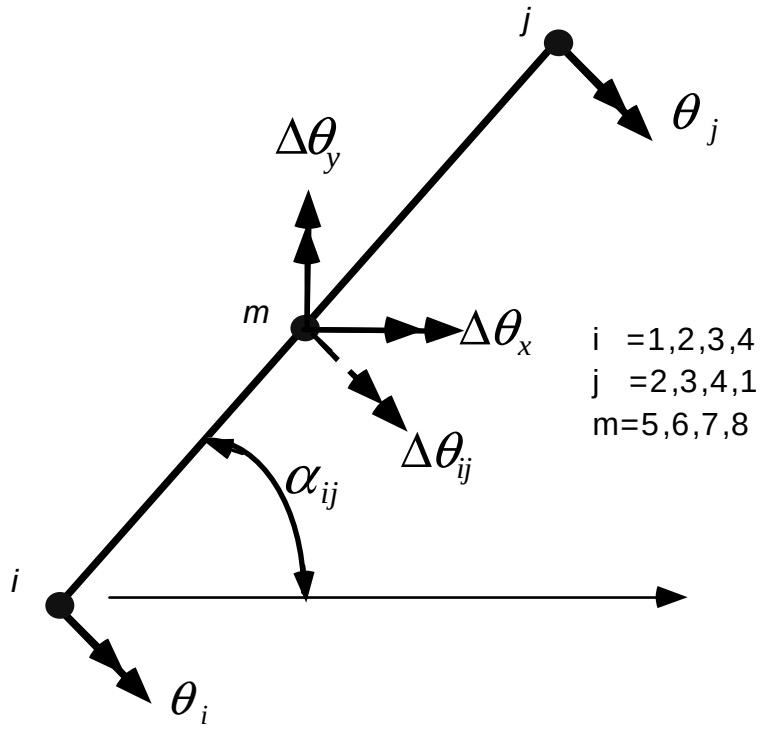


Figure 40.2. Element edge [Wilson, 2000].

which can be rewritten, referring to Figure 40.3 as:

$$\gamma_{ij} = \frac{1}{L}\left(u_{zj} - u_{zi}\right) - \frac{\sin\alpha_{ij}}{2}\left(\theta_{xi} + \theta_{xj}\right) + \frac{\cos\alpha_{ij}}{2}\left(\theta_{yi} + \theta_{yj}\right) - \frac{2}{3}\Delta\theta_{ij}$$

The nodal shears are then written in terms of the side shears as

$$\begin{bmatrix}\gamma_{ij}\\ \gamma_{ki}\end{bmatrix} = \begin{bmatrix}\cos\alpha_{ij} & \sin\alpha_{ij}\\ \cos\alpha_{ki} & \sin\alpha_{ki}\end{bmatrix}\begin{bmatrix}\gamma_{xz}\\ \gamma_{yz}\end{bmatrix}$$

which can be inverted to obtain the nodal shears:

$$\begin{bmatrix}\gamma_{xz}\\ \gamma_{yz}\end{bmatrix} = \frac{1}{\cos\alpha_{ij}\sin\alpha_{ki} - \cos\alpha_{ki}\sin\alpha_{ij}}\begin{bmatrix}\sin\alpha_{ki} & -\cos\alpha_{ki}\\ -\sin\alpha_{ij} & \cos\alpha_{ij}\end{bmatrix}\begin{bmatrix}\gamma_{ij}\\ \gamma_{ki}\end{bmatrix}$$

The standard bilinear basis functions are used to interpolate the nodal shears to the integration points.

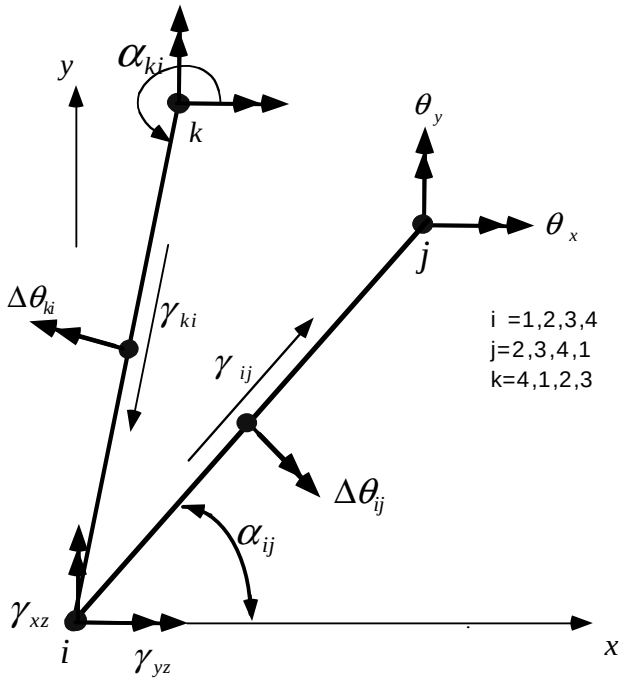


Figure 40.3. Nodal and edge shear strains [Wilson 2000].

40.2.2 Membrane Element

The membrane element, which is also coded from Wilson's textbook [2000], is based on the eight node isoparametric element, see Figure 40.4.

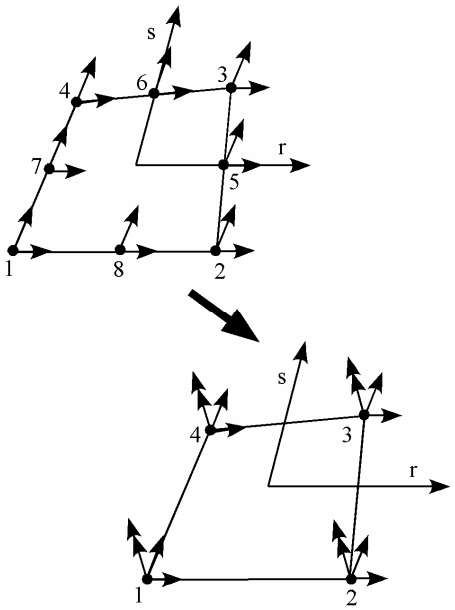


Figure 40.4. Eight node membrane element.

The inplane displacement field for the 8 node membrane is interpolated, using the serendipity shape functions with the mid-side relative displacements, from:

$$u_x(r,s) = \sum_{i=1}^{4} N_i(r,s)u_{xi} + \sum_{i=5}^{8} N_i(r,s)\Delta u_{xi}$$

$$u_y(r,s) = \sum_{i=1}^{4} N_i(r,s)u_{yi} + \sum_{i=5}^{8} N_i(r,s)\Delta u_{yi}$$

It is desired to replace the mid side relative displacment by drilling rotations at the corner nodes. Consider Figure 40.5:

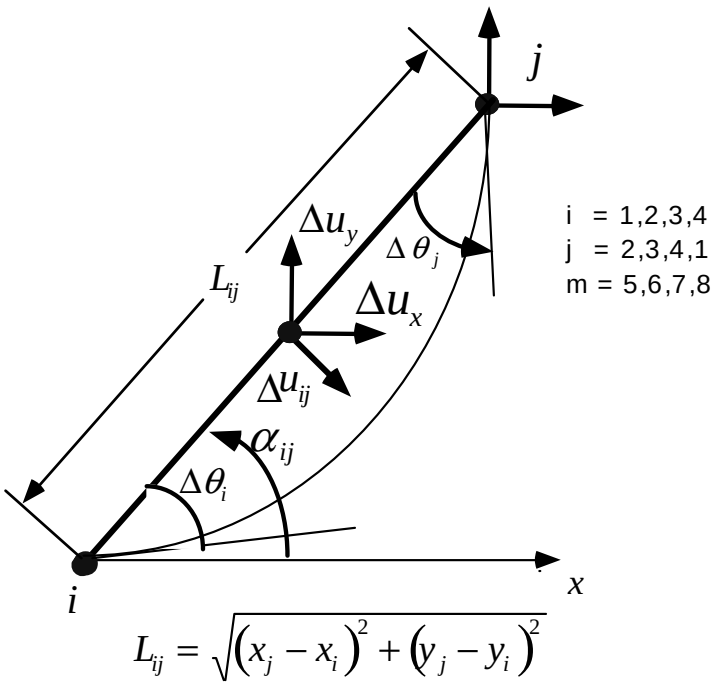


Figure 40.5. Corner node drilling rotations and mid side edge normal displacement [Wilson, 2000].

the mid-side normal displacements along the edge are parabolic, i.e.,

$$\Delta u_{ij} = \frac{L_{ij}}{8}\left(\Delta\theta_j - \Delta\theta_i\right)$$

while the mid-side tangential displacements are interpolated linearly from the end node displacements, thus,

$$\Delta u_x(r,s) = \cos\alpha_{ij}\Delta u_{ij} = \cos\alpha_{ij}\frac{L_{ij}}{8}\left(\Delta\theta_j - \Delta\theta_i\right)$$

$$\Delta u_y(r,s) = -\sin\alpha_{ij}\Delta u_{ij} = -\sin\alpha_{ij}\frac{L_{ij}}{8}\left(\Delta\theta_j - \Delta\theta_i\right)$$

$$u_x(r,s) = \sum_{i=1}^{4} N_i(r,s)u_{xi} + \sum_{i=5}^{8} M_{xi}(r,s)\Delta\theta_i$$

$$u_y(r,s) = \sum_{i=1}^{4} N_i(r,s)u_{yi} + \sum_{i=5}^{8} M_{yi}(r,s)\Delta\theta_i$$

This element has one singularity in the *drilling mode* of equal corner rotations, see Figure 40.6.

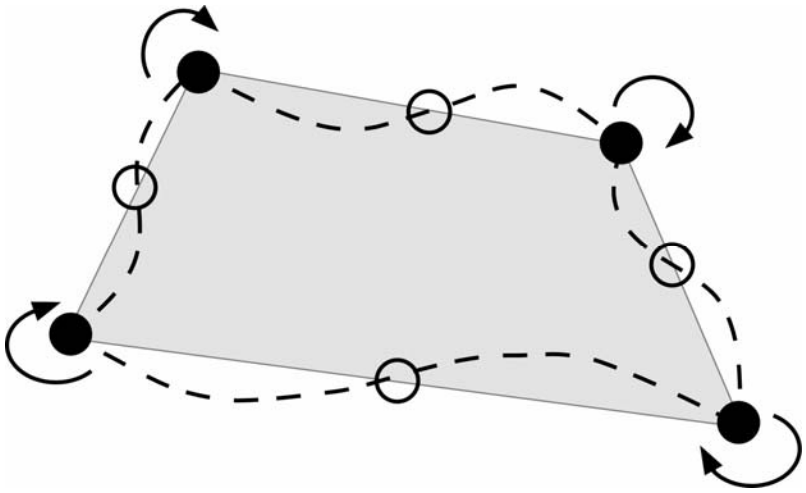


Figure 40.6. Zero energy mode

Ibrahimbegovic and Wilson [1991] added a penalty formulation to the potential energy of the element to eliminate the singularity. The following penalty term connects the averaged nodal rotation to the continuum mechanics rotation

$$\frac{1}{2}\left(\frac{\partial u_x}{\partial y} - \frac{\partial u_y}{\partial x}\right) - \omega$$

at the center of the element. The element performance is highly insensitive to the chosen value of the penalty factor and some fraction of the elastic modulii, *G* or *E*, is frequently used.

- 5, 8 or 9 point quadrature can be applied. The 5 and 8 point schemes induce a 'soft' first deformational mode, whereas the 9 point Gaussian quadrature results in a stiffer mode.
- A *membrane locking correction* (Taylor) is applied to (i) alleviate a membrane-bending interaction associated with the drilling degrees of freedom and (ii) allow the standard application of the consistent nodal load at the edge. The correction has a slight stiffening effect (see e.g. Cook Cantilever).
- A *warping correction* is applied using the *rigid link correction* (see Figure 40.7).

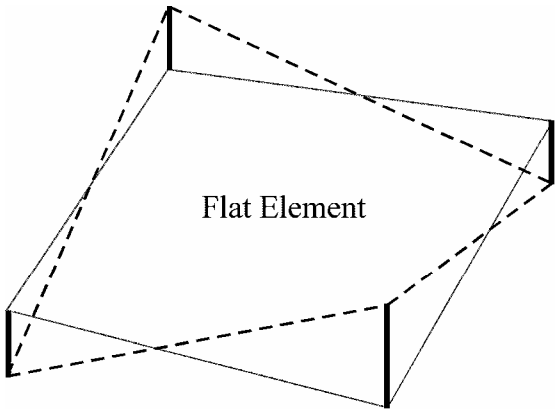


Figure 40.7

40.3 Assumed Strain/Membrane with Drilling Degree-of-freedom (element #18)

40.3.1 Membrane Element

Formulation is the same as above for element type 20.

40.3.2 Plate Element

The Discrete Kirchhoff Quadrilateral element is an excellent thin shell element based on

- Rotational field is interpolated using the 8-node isoparametric parent element.
- Transverse displacement *w* assumed as cubic along the sides and collocated along the sides and at the nodes using the Kirchhoff condition that equates the fiber rotation to the slope. The Kirchhoff assumptions are satisfied along the entire boundary of the element.
- The rotational field about an axis parallel to the side is constrained linearly along the sides.

The warping correction is applied as above.

40.4 Differences between Element Types 18 and 20.

- The DKQ does not account for transverse shear because it locally enforces the Kirchhoff condition. Hence, element type 20 is better for layered composites and thick plates.

REFERENCES

Abbo A.J., S.W. Sloan, "A Smooth Hyperbolic Approximation to the Mohr-Coulomb Yield Criterion," Computers and Structures, Vol 54, No 1, 1995.

Addessio, F.L., D.E. Carroll, J.K. Dukowicz, F.H. Harlow, J.N. Johnson, B.A. Kashiwa, M.E. Maltrud, H.M. Ruppel, "CAVEAT: A Computer Code for Fluid Dynamics Problems with Large Distortion and Internal Slip," Report LA-10613-MS, UC-32, Los Alamos National Laboratory (1986).

Ahmad, S., Irons, B.M. and Zienkiewicz, O.C., "Analysis of Thick and Thin Shell Structures by Curved Finite Elements," Int. J. Numer. Meths. Eng., **2** (1970).

Amsden, A. A., and Hirt, C. W., "YAQUI: An Arbitrary Lagrangian-Eulerian Computer Program for Fluid Flow at All Speeds," Los Alamos Scientific Laboratory, LA-5100 (1973).

Amdsden, A., A., Ruppel, H. M., and Hirt, C. W., "SALE: A Simplified ALE Computer Program for Fluid Flow at All Speeds," Los Alamos Scientific Laboratory (1980).

Argyris, J.H., Kelsey, S., and Kamel, H., "Matrix Methods of Structural Analysis: A Precis of recent Developments," Matrix Methods of Structural Analysis, Pergamon Press (1964).

Arruda, E., and M. Boyce, "A Three-Dimensional Constitutive Model for the Large Stretch Behavior of Rubber Elastic Materials," published in the Journal of the Mechanics and Physics of Solids, Vol. **41**, No. 2, 389-412 (1993).

Auricchio, F., Taylor, R.L. and Lubliner J., "Shape-memory alloys: macromodelling and numerical simulations of the superelastic behavior", Computer Methods in Applied Mechanics and Engineering **146**, 281-312 (1997).

Auricchio, F. and Taylor, R.L., "Shape-memory alloys: modeling and numerical simulations of the finite-strain superelastic behavior", Computer Methods in Applie Mechanics and Engineering **143**, 175-194 (1997).

Back, S.Y., and Will, K.M., "A Shear-flexible Element with Warping for Thin-Walled Open Sections," *International Journal for Numerical Methods in Engineering*, 43, 1173-1191 (1998).

Bahler AS: The series elastic element of mammalian skeletal muscle. Am J Physiol 213:1560-1564, 1967.

Bammann, D. J., and E.C. Aifantis, "A Model for Finite-Deformation Plasticity," Acta Mechanica, **70**, 1-13 (1987).

Bammann, D. J., "Modeling the Temperature and Strain Rate Dependent Large Deformation of Metals," Proceedings of the 11th US National Congress of Applied Mechanics, Tucson, AZ (1989).

Bammann, D. J., and Johnson, G., "On the Kinematics of Finite-Deformation Plasticity," Acta Mechanica **69**, 97-117 (1987).

Bammann, D.J., Chiesa, M.L., McDonald, A., Kawahara, W.A., Dike, J.J. and Revelli, V.D., "Predictions of Ductile Failure in Metal Structures," in AMD-Vol. 107, Failure Criteria and Analysis in Dynamic Response, edited by. H.E. Lindberg, 7-12 (1990).

Bandak, F.A., private communications, U.S. Dept. of Trans., Division of Biomechanics Research, 400 7th St., S.W. Washington, DC 20590 (1991).

Barlat and Lian, J., "Plastic Behavior and Stretchability of Sheet Metals, Part I: A Yield Function for Orthotropic Sheets Under Plane Stress Conditions," *International Journal of Plasticity*, **5**, 51-66 (1989).

Barlat, F., Lege, D.J., and Brem, J.C., "A Six-Component Yield Function for Anisotropic Materials," *International Journal of Plasticity*, **7**, 693-712 (1991).

Bathe, K. J., Finite Element Procedures in Engineering Analysis, Prentice-Hall (1982).

Bathe, K. J., and Wilson, E.L., Numerical Methods in Finite Element Analysis, Prentice-Hall (1976).

Bathe, K.J., and Dvorkin, E.N., "A Continuum Mechanics Based Four Node Shell Element for General Nonlinear Analysis," Int. J. Computer-Aided Eng. and Software, Vol. **1**, 77-88 (1984).

Bathe, K.-J. and Dvorkin, E.N. A four node plate bending element based on Mindlin-Reissner plate theory and a mixed interpolation, *Int. J. Num. Meth. Eng.*, **21**, 367-383, 1985.

Batoz, J.L. and Ben Tahar, M. Evaluation of a new quadrilateral thin plate bending element, *Int. J. Num. Meth. Eng*., **18**, 1644-1677, 1982**.**

Battini, J., and Pacoste C., "Co-rotational Beam Elements with Warping Effects in Instability Problems," *Computational Methods in Applied Mechanical Engineering*, 191, 1755-1789, (2002).

Bazeley, G.P., Cheung, W.K., Irons, B.M. and Zienkiewicz, O.C., "Triangular Elements in Plate Bending—Conforming and Nonconforming Solutions in Matrix Methods and Structural Mechanics," Proc. Conf. on Matrix Methods in Structural Analysis, Rept. AFFDL-R-66-80, Wright Patterson AFB, 547-576 (1965).

Belytschko, T., "Transient Analysis," Structural Mechanics Computer Programs, edited by W. Pilkey, et. al., University Press of Virginia, 255-276 (1974).

Belytschko, T., and Bindeman, L. P., "Assumed Strain Stabilization of the Eight Node Hexahedral Element," Comp. Meth. Appl. Mech. Eng. **105**, 225-260 (1993).

Belytschko, T., and Hsieh, B.J., "Nonlinear Transient Finite Element Analysis with Convected Coordinates," Int. J. Num. Meths. Engrg., **7**, 255–271 (1973).

Belytschko, T., and Lin, J., "A New Interaction Algorithm with Erosion for EPIC-3", Contract Report BRL-CR-540, U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, Maryland (1985).

Belytschko, T., Lin, J., and Tsay, C.S., "Explicit Algorithms for Nonlinear Dynamics of Shells," Comp. Meth. Appl. Mech. Eng. **42**, 225-251 (1984) [**a**].

Belytschko, T., Ong, J. S.-J., Liu, W.K. and Kennedy, J.M., "Hourglass Control in Linear and Nonlinear Problems", Comput. Meths. Appl. Mech. Engrg., **43**, 251–276 (1984b).

Belytschko, T., Stolarski, H., and Carpenter, N., "A C_0 Triangular Plate Element with One-Point Quadrature," International Journal for Numerical Methods in Engineering, **20**, 787-802 (1984) [**b**].

Belytschko, T., Schwer, L., and Klein, M. J., "Large Displacement Transient Analysis of Space Frames," International Journal for Numerical and Analytical Methods in Engineering, **11**, 65-84 (1977).

Belytschko, T., and Tsay, C.S., "Explicit Algorithms for Nonlinear Dynamics of Shells," AMD, **48**, ASME, 209-231 (1981).

Belytschko, T., and Tsay, C.S., "WHAMSE: A Program for Three-Dimensional Nonlinear Structural Dynamics," Report EPRI NP-2250, Project 1065-3, Electric Power Research Institute, Palo Alto, CA (1982).

Belytschko, T., and Tsay, C. S., "A Stabilization Procedure for the Quadrilateral Plate Element with One-Point Quadrature," Int. J. Num. Method. Eng. **19**, 405-419 (1983).

Belytschko, T., Yen, H. R., and Mullen R., "Mixed Methods for Time Integration," Computer Methods in Applied Mechanics and Engineering, **17**, 259-175 (1979).

Belytschko, T., "Partitioned and Adaptive Algorithms for Explicit Time Integration," in Nonlinear Finite Element Analysis in Structural Mechanics, ed. by Wunderlich, W. Stein, E, and Bathe, J. J., 572-584 (1980).

Belytschko, T., Wong, B.L., and Chiang, H.Y., "Improvements in Low-Order Shell Elements for Explicit Transient Analysis," Analytical and Computational Models of Shells, A.K. Noor, T. Belytschko, and J. Simo, editors, ASME, CED, **3**, 383-398 (1989).

Belytschko, T., Wong, B.L., and Chiang, H.Y., "Advances in One-Point Quadrature Shell Elements," Comp. Meths. Appl. Mech. Eng., **96**, 93-107 (1992).

Belytschko, T., Wong, B. L., Plaskacz, E. J., "Fission - Fusion Adaptivity in Finite Elements for Nonlinear Dynamics of Shells," Computers and Structures, Vol. 33, 1307-1323 (1989).

Belytschko, T., Lu, Y.Y. and Gu, L., "Element-free Galerkin Methods," Int. J. Numer. Methods Engrg. **37**, 229-256 (1994).

Benson, D.J., "Vectorizing the Right-Hand Side Assembly in an Explicit Finite Element Program," Comp. Meths. Appl. Mech. Eng., **73**, 147-152 (1989).

Benson, D. J., and Hallquist, J.O., "A Simple Rigid Body Algorithm for Structural Dynamics Program," Int. J. Numer. Meth. Eng., **22** (1986).

Benson, D.J., and Hallquist J.O., "A Single Surface Contact Algorithm for the Postbuckling Analysis of Shell Structures," Comp. Meths. Appl. Mech. Eng., **78**, 141-163 (1990).

Benson, D. J., "Momentum Advection on a Staggered Mesh," Journal of Computational Physics, **100**, No. 1, (1992).

Benson, D. J., "Vectorization Techniques for Explicit Arbitrary Lagrangian Eulerian Calculations," Computer Methods in Applied Mechanics and Engineering (1992).

Berstad, T., "Material Modelling of Aluminium for Crashworthiness Analysis", Dr.Ing. Dissertation, Department of Structural Engineering, Norwegian University of Science and Technology, Trondheim, Norway (1996).

Berstad T., Hopperstad, O.S., and Langseth, M., "Elasto-Viscoplastic Consitiutive Models in the Explicit Finite Element Code LS-DYNA," Proceedings of the Second International LS-DYNA Conference, San Francisco, CA (1994).

Blatz, P.J., and Ko, W.L., "Application of Finite Element Theory to the Deformation of Rubbery Materials," Trans. Soc. of Rheology, **6**, 223-251 (1962).

Bodig, Jozsef and Benjamin A. Jayne, Mechanics of Wood and Wood Composites, Krieger Publishing Company, Malabar, FL (1993).

Boeing, Boeing Extreme Mathematical Library BCSLIB-EXT User's Guide, The Boeing Company, Document Number 20462-0520-R4 (2000).

Borrvall, T., Development and implementation of material tangent stiffnesses for material model 76 in LS-DYNA, ERAB-02:46, Engineering Research Nordic AB, Linköping (2002).

Borrvall, T., Revision of the implementation of material 36 for shell elements in LS-DYNA, ERAB Report E0307, Engineering Research Nordic AB, Linköping (2003).

Brekelmans, W.A.M., Scheurs, P.J.G., and de Vree, J.H.P., "Continuum damage mechanics for softening of brittle materials", *Acta Mechanica*, 93, 133-143, (1991).

Brooks, A. N., and Hughes, T. J. R., "Streamline Upwind/Petrov-Galerkin Formulations for Convection Dominated Flows with Particular Emphasis on the Incompressible Navier-

Stokes Equations," Computer Methods in Applied Mechanics and Engineering, **32**, 199-259, (1982).

Burton, D.E., et. al., "Physics and Numerics of the TENSOR Code," Lawrence Livermore National Laboratory, Internal Document UCID-19428 (July 1982).

Chang, F.K., and Chang, K.Y., "Post-Failure Analysis of Bolted Composite Joints in Tension or Shear-Out Mode Failure," J. of Composite Materials, **21**, 809-833 (1987).[**a**]

Chang, F.K., and Chang, K.Y., "A Progressive Damage Model for Laminated Composites Containing Stress Concentration," J. of Composite Materials, **21**, 834-855 (1987).[**b**]

Chang, F.S., "Constitutive Equation Development of Foam Materials," Ph.D. Dissertation, submitted to the Graduate School, Wayne State University, Detroit, Michigan (1995).

Chang, F.S., J.O. Hallquist, D. X. Lu, B. K. Shahidi, C. M. Kudelko, and J.P. Tekelly, "Finite Element Analysis of Low Density High-Hystersis Foam Materials and the Application in the Automotive Industry," SAE Technical Paper 940908, in Saftey Technology (SP-1041), International Congress and Exposition, Detroit, Michigan (1994).

Chen, J.S., Pan, C., Wu, C.T. and Liu, W.K., "Reproducing Kernel Particle Methods for Large Deformation Analysis of Nonlinear Structures," Comput. Methods Appl. Mech. Engrg. **139**, 195-227 (1996).

Chen, J.S. and Wu, C.T., "Generalized Nonlocal Meshfree Method in Strain Localization," *Proceeding of International Conference on Computational Engineering Science*, Atlanta, Georiga, 6-9 October (1998).

Chen, J.S. and Wang, H.P. "New Boundary Condition Treatmens in Meshfree Computation of Contact Problems," Computer Methods in Applied Mechanics and Engineering, **187**, 441-468, (2001a).

Chen, J.S., Wu, C.T., Yoon, S. and You, Y. "A Stabilized Conforming Nodal Integration for Galerkin Meshfree Methods," Int. J. Numer. Methods Engrg. **50**, 435-466 (2001b).

Chen, W.F., and Baladi, G.Y., *Soil Plasticity: Theory and Implementation*, Elesvier, New York, (1985).

Christensen, R.M. "A Nonlinear Theory of Viscoelasticity for Application to Elastomers," *Journal of Applied Mechanics,* **47**, American Society of Mechanical Engineers, 762-768 (December 1980).

Chung, K., and K. Shah, "Finite Element Simulation of Sheet Metal Forming for Planar Anisotropic Metals," *Int. J. of Plasticity*, **8**, 453-476 (1992).

Cochran, S.G., and Chan, J., "Shock Initiation and Detonation Models in One and Two Dimensions," University of California, Lawrence Livermore National Laboratory, Rept. UCID-18024 (1979).

Cohen, M., and Jennings, P.C., "Silent Boundary Methods for Transient Analysis", in Computational Methods for Transient Analysis, T. Belytschko and T.J.R. Hughes, editors, North-Holland, New York, 301-360 (1983).

Cook, R. D., Concepts and Applications of Finite Element Analysis, John Wiley and Sons, Inc. (1974).

Couch, R., Albright, E. and Alexander, "The JOY Computer Code," University of California, Lawrence Livermore National Laboratory, Rept. UCID-19688 (1983).

Cray Research Inc., "CF77 Compiling System, Volume 4: Parallel Processing Guide," SG-3074 4.0, Mendota Heights, MN (1990).

Crisfield M.A., Non-linear Finite Element Analysis of Solids and Structures, Volume 2, Advanced Topics, John Wiley, New York (1997).

DeBar, R. B., "Fundamentals of the KRAKEN Code," Lawrence Livermore Laboratory, UCIR-760 (1974).

Deshpande, V.S. and N.A. Fleck, "Isotropic Models for Metallic Foams," Journal of the Mechanics and Physics of Solids, Vol. 48, 1253-1283, (2000).

Dobratz, B.M., "LLNL Explosives Handbook, Properties of Chemical Explosives and Explosive Simulants," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-52997 (1981).

Duff, I. S., and Reid, J. K., "The Multifrontal Solution of Indefinite Sparse Symmetric Linear Equations," ACM Transactions of Mathematical Software, **9**, 302-325 (1983).

Englemann, B.E. and Whirley, R.G., "A New Explicit Shell Element Formulation for Impact Analysis," In Computational Aspects of Contact Impact and Penetration, Kulak, R.F., and Schwer, L.E., Editors, Elmepress International, Lausanne, Switzerland, 51-90 (1991).

Englemann, B.E., Whirley, R.G., and Goudreau, G.L., "A Simple Shell Element Formulation for Large-Scale Elastoplastic Analysis," In Analytical and Computational Models of Shells, Noor, A.K., Belytschko, T., and Simo, J.C., Eds., CED-Vol. 3, ASME, New York, New York (1989).

Farhoomand, I., and Wilson, E.L., "A Nonlinear Finite Element Code for Analyzing the Blast Response of Underground Structures," U.S. Army Waterways Experiment Station, Contract Rept. N-70-1 (1970).

Feng, W.W., Private communication, Livermore, CA (1993).

Flanagan, D.P. and Belytschko, T., "A Uniform Strain Hexahedron and Quadrilateral and Orthogonal Hourglass Control," Int. J. Numer. Meths. Eng. **17**, 679-706 (1981)

Fleck, J.T., "Validation of the Crash Victim Simulator," **I - IV**, Report No. DOT-HS-806 279 (1981).

"Fundamental Study of Crack Initiation and Propagation," Author unknown, LLNL report, document received from LSTC (2003).

Galbraith, P.C., M.J. Finn, S.R. MacEwen, et.al., "Evaluation of an LS-DYNA3D Model for Deep-Drawing of Aluminum Sheet", FE-Simulation of 3-D Sheet Metal Forming Processes in Automotive Industry, VDI Berichte, **894**, 441-466 (1991).

Ginsberg, M. and Johnson, J.P., "Benchmarking the Performance of Physical Impact Simulation Software on Vector and Parallel Computers," Proc. of the Supercomputing 88: Vol. II, Science and Applications, Computer Society Press (1988).

Gingold, R.A., and Monaghan, J.J., "Smoothed Particle Hydrodynamics: Theory and Application to Non-Spherical Stars," Mon. Not. R. Astron. Soc. **181**, 375-389 (1977).

Ginsberg, M., and Katnik, R.B., "Improving Vectorization of a Crashworthiness Code," SAE Technical Paper No. 891985, Passenger Car Meeting and Exposition, Dearborn, MI (1989).

Giroux, E.D., "HEMP User's Manual," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-51079 (1973).

Govindjee, S., Kay G.J., and Simo, J.C., "Anisotropic Modeling and Numerical Simulation of Brittle Damage in Concrete," Report Number UCB/SEM M-94/18, University of California at Berkeley, Department of Civil Engineering (1994).

Govindjee, S., Kay G.J., and Simo, J.C., "Anisotropic Modeling and Numerical Simulation of Brittle Damage in Concrete," *International Journal for Numerical Methods in Engineering*, Volume 38, pages 3611-3633 (1995).

Green, A.E. and Naghdi, P.M., "A General Theory of Elastic-Plastic Continuum," *Archive for Rational Mechanics and Analysis*, **18**, 251 (1965).

Groenwold, A.A. and Stander, N. An efficient 4-node 24 d.o.f. thick shell finite element with 5-point quadrature. *Eng. Comput.* **12**(8), 723-747, 1995.

Grimes, Roger, Lewis, John G., and Simon, Horst D., "A Shifted Block Lanczos Algorithm for Solving Sparse Symmetric Generalized Eigenproblems," SIAM Journal of Matrix Analyis and Applications, **15**, 228-272 (1994).

Guccione, J., A. McCulloch, and L. Waldman, "Passive Material Properties of Intact Ventricular Myocardium Determined from a Cylindrical Model," published in the ASME Journal of Biomechanical Engineering, **113**, 42-55 (1991).

Hallquist, J.O., "Preliminary User's Manuals for DYNA3D and DYNAP (Nonlinear Dynamic Analysis of Solids in Three Dimension)," University of California, Lawrence Livermore National Laboratory, Rept. UCID-17268 (1976) and Rev. 1 (1979).[**a**]

Hallquist, J.O., "A Procedure for the Solution of Finite Deformation Contact-Impact Problems by the Finite Element Method," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-52066 (1976).

Hallquist, J.O., "A Numerical Procedure for Three-Dimensional Impact Problems," American Society of Civil Engineering, Preprint 2956 (1977).

Hallquist, J.O., "A Numerical Treatment of Sliding Interfaces and Impact," in: K.C. Park and D.K. Gartling (eds.) Computational Techniques for Interface Problems, AMD, **30**, ASME, New York (1978).

Hallquist, J.O., "NIKE2D: An Implicit, Finite-Element Code for Analyzing the Static and Dynamic Response of Two-Dimensional Solids," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-52678 (1979).[**b**]

Hallquist, J.O. (1998), LS-DYNA Theoretical Manual, May (1998).

Hallquist, J.O., "User's Manual for DYNA2D— An Explicit Two-Dimensional Hydrodynamic Finite Element Code with Interactive Rezoning," University of California, Lawrence Livermore National Laboratory, Rept. UCID-18756 (1980).

Hallquist, J.O., "User's Manual for DYNA3D and DYNAP (Nonlinear Dynamic Analysis of Solids in Three Dimensions)," University of California, Lawrence Livermore National Laboratory, Rept. UCID-19156 (1981).[**a**]

Hallquist, J. O., "NIKE3D: An Implicit, Finite-Deformation, Finite-Element Code for Analyzing the Static and Dynamic Response of Three-Dimensional Solids," University of California, Lawrence Livermore National Laboratory, Rept. UCID-18822 (1981).[**b**]

Hallquist, J.O., "DYNA3D User's Manual (Nonlinear Dynamic Analysis of Solids in Three Dimensions)," University of California, Lawrence Livermore National Laboratory, Rept. UCID-19156 (1982; Rev. 1: 1984; Rev. 2: 1986).

Hallquist, J.O., "Theoretical Manual for DYNA3D," University of California, Lawrence Livermore National Laboratory, Rept. UCID-19401 (March, 1983).

Hallquist, J.O., "DYNA3D User's Manual (Nonlinear Dynamic Analysis of Solids in Three Dimensions)," University of California, Lawrence Livermore National Laboratory, Rept. UCID-19156 (1988, Rev. 4).

Hallquist, J.O., "DYNA3D User's Manual (Nonlinear Dynamic Analysis of Solids in Three Dimensions)," Livermore Software Technology Corporation, Rept. 1007 (1990).

Hallquist, J.O., LS-DYNA Keyword User's Manual, version 970, Livermore Software Technology Corporation (April, 2003).

Hallquist, J.O., Benson, D.J., and Goudreau, G.L., "Implementation of a Modified Hughes-Liu Shell into a Fully Vectorized Explicit Finite Element Code," Proceedings of the

International Symposium on Finite Element Methods for Nonlinear Problems, University of Trondheim, Trondheim, Norway (1985).

Hallquist, J.O., and Benson, D.J., "A Comparison of an Implicit and Explicit Implementation of the Hughes-Liu Shell," Finite Element Methods for Plate and Shell Structures, T.J.R. Hughes and E. Hinton, Editors, 394-431, Pineridge Press Int., Swanea, U.K. (1986).

Hallquist, J.O., and Benson, D.J., "DYNA3D User's Manual (Nonlinear Dynamic Analysis of Solids in Three Dimensions)," University of California, Lawrence Livermore National Laboratory, Rept. UCID-19156 (1986, Rev. 2).

Hallquist, J.O. and Benson, D.J., "DYNA3D User's Manual (Nonlinear Dynamic Analysis of Solids in Three Dimensions)," University of California, Lawrence Livermore National Laboratory, Rept. UCID-19156 (1987, Rev. 3).

Hallquist, J.O., Stillman, D.W., Hughes, T.J.R., and Tarver, C.,"Modeling of Airbags Using MVMA/DYNA3D," LSTC Report (1990).

Harten, A., "ENO Schemes with Subcell Resolution," J. of Computational Physics, **83**, 148-184, (1989).

Hashin, Z, "Failure Criteria for Unidirectional Fiber Composites," Journal of Applied Mechanics, **47**, 329 (1980).

Herrmann, L.R. and Peterson, F.E., "A Numerical Procedure for Viscoelastic Stress Analysis," Seventh Meeting of ICRPG Mechanical Behavior Working Group, Orlando, FL, CPIA Publication No. 177 (1968).

Hill A.V., "The heat of shortening and the dynamic constants of muscle," Proc Roy Soc B126:136-195, (1938).

Hill, R., "A Theory of the Yielding and Plastic Flow of Anisotropic Metals," Proceedings of the Royal Society of London, Series A., **193,** 281 (1948).

Hill, R. "Consitiutive Modeling of Orthotropic Plasticity in Sheet Metals," *Journal of Mechanics and Physics of Solids*, **38**, Number 3, 405-417 (1999).

Holmquist, T.J., G.R. Johnson, and W.H. Cook, "A Computational Constitutive Model for Concrete Subjected to Large Strains, High Strain Rates, and High Pressures, pp. 591-600, (1993).

Holtz R.D., W. D. Kovacs, " An Introduction to Geotechnical Engineering", Prentice Hall, Inc. New Jersey, 1981

Hopperstad, O.S. and Remseth, S., "A Return Mapping Algorithm for a Class of Cyclic Plasticity Models," *International Journal for Numerical Methods in Engineering*, **38**, 549-564 (1995).

Hughes, T.J.R., The Finite Element Method, Linear Static and Dynamic Finite Element Analysis, Prentice-Hall Inc., Englewood Cliffs, NJ (1987).

Hughes, T.J.R., "Generalization of Selective Integration Procedures to Anisotropic and Nonlinear Media," Int. J. Numer. Meth. Eng. **15**, 9 (1980).

Hughes, T.J.R. and Carnoy, E., "Nonlinear Finite Element Shell Formulation Accounting for Large Membrane Strains," AMD-Vol.48, ASME, 193-208 (1981).

Hughes, T.J.R. and Liu, W.K., "Nonlinear Finite Element Analysis of Shells: Part I. Two-Dimensional Shells." Comp. Meths. Appl. Mechs. **27**, 167-181 (1981).

Hughes, T.J.R. and Liu, W.K., "Nonlinear Finite Element Analysis of Shells: Part II. Three-Dimensional Shells." Comp. Meths. Appl. Mechs. **27**, 331-362 (1981).

Hughes, T.J.R., Liu,W.K., and Levit, I., "Nonlinear Dynamics Finite Element Analysis of Shells." Nonlinear Finite Element Analysis in Struct. Mech., Eds. W. Wunderlich, E. Stein, and K.J. Bathe, Springer-Verlag, Berlin, 151–168 (1981).

Hughes, T.J.R., Taylor, R. L., Sackman, J. L., Curnier, A.C., and Kanoknukulchai, W., "A Finite Element Method for a Class of Contact-Impact Problems," J. Comp. Meths. Appl. Mechs. Eng. **8**, 249-276 (1976).

Hughes, T.J.R., and Winget, J., "Finite Rotation Effects in Numerical Integration of Rate Constitutive Equations Arising in Large-Deformation Analysis," Int. J. Numer. Meths. Eng., **15**, 1862-1867, (1980).

Hulbert, G.M. and Hughes, T.J.R., "Numerical Evaluation and Comparison of Subcycling Algorithms for Structural Dynamics," Technical Report, DNA-TR-88-8, Defense Nuclear Agency, Washington, DC (1988).

Ibrahimbegovic, A. and Wilson, E.L. A unified formulation for triangular and quadrilateral flat shell finite elements with six nodal degrees of freedom, *Comm. Applied Num. Meth*, **7**, 1-9, 1991.

Isenberg, J., D.K. Vaughn, and I. Sandler, "Nonlinear Soil-Structure Interaction," EPRI Report MP-945, Weidlinger Associates, December (1978).

Jaumann, G. "Geschlossenes System Physikalischer und Chemischer Differentialdesefz, Sitz Zer. Akad. Wiss. Wein, (IIa), **120**, 385 (1911).

Johnson, G.C. and Bammann D.J., "A Discussion of Stress Rates in Finite Deformation Problems" *International Journal of Solids and Structures*, **20**, 725-737 (1984).

Johnson, G.R. and Cook,W. H., "A Constitutive Model and Data for Metals Subjected to Large Strains, High Strain Rates and High Temperatures," presented at the Seventh International Symposium on Ballistics, The Hague, The Netherlands, April (1983).

Johnson, G.R. and T.J. Holmquist, "An Improved Computational Model for Brittle Materials" in High-Pressure Science and Technology - 1993 American Institute of Physics Conference Proceedings 309 (c 1994) pp.981-984 ISBN 1-56396-219-5.

Johnson, C., Navert, U., and Pitkaranta, J., "Finite Element Methods for Linear Hyperbolic Problems," Computer Methods in Applied Mechanics and Engineering, **45**, 285-312, (1984).

Jones, N., "Structural Aspects of Ship Collisions," Chapter 11, in Structural Crashworthiness, Eds. N. Jones and T Wierzbicki, Butterworths, London, 308-337 (1983).

Jones, R.M., Mechanics of Composite Materials, Hemisphere Publishing Co., New York (1975).

Ju J.W., "Energy-Based Coupled Elastoplastic Damage Models at Finite Strains", J. of Eng Mech., Vol 115, No 11, Nov 1989.

Karypis, G., and Kumar V., "METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices," Department of Computer Science, University of Minnesota (1998).

Katz, J., and Maskew, B., "Unsteady Low-Speed Aerodynamic Model for Complete Aircraft Configurations," Journal of Aircraft **25**, 4 (1988).

Kenchington, G. J., "A Non-Linear Elastic Material Model for DYNA3D," Proceedings of the DYNA3D Users Group Conference, September 1988, published by Boeing Computer Services (Europe) Limited.

Kennedy, J. M., Belytschko,T., and Lin, J. I., "Recent Developments in Explicit Finite Element Techniques and their Applications to Reactor Structures," Nuclear Engineering and Design **97**, 1-24 (1986).

Kim, N., Seo, K., and Kim, M., "Free Vibration and Spatial Stability of Non-symmetric Thin-Walled Curved Beams with Variable Curvatures," *International Journal of Solids and Structures*, 40, 3107-3128, (2003).

Klisinski M., "Degradation and Plastic Deformation of Concrete", Ph.D. thesis, Polish Academy of Sciences, 1985, IFTR report 38.

Kosloff, D. and Frazier, G.A.,"Treatment of Hourglass Patterns in Low Order Finite Element Codes," Int. J. Numer. Anal. Meth. Geomech. **2**, 57-72 (1978)

Kretschmann, D. E., and David W. Green, "Modeling Moisture Content-Mechanical Property Relationships for Clear Southern Pine," Wood and Fiber Science, 28(3), pp. 320-337 (1996).

Kreyszig, E., Advanced Engineering Mathematics, John Wiley and Sons, New York, New York (1972).

Krieg, R.D.,"A Simple Constitutive Description for Cellular Concrete," Sandia National Laboratories, Albuquerque, NM, Rept. SC-DR-72-0883 (1972).

Krieg, R.D. and Key, S.W., Implementation of a Time Dependent Plasticity Theory into Structural Computer Programs, Vol. 20 of Constitutive Equations in Viscoplasticity: Computational and Engineering Aspects (American Society of Mechanical Engineers, New York, N.Y., 1976), 125-137.

Kumar, P., Nukala, V. V., and White, D. W., "A Mixed Finite Element for Three-dimensional Nonlinear Analysis of Steel Frames," *Computational Methods in Applied Mechanical Engineering*, 193, 2507-2545, (2004).

Lancaster, P. and Salkauskas, K., "Surfaces generated by moving least squares methods", Math. Comput. **37**, 141-158, (1981).

Landshoff, R., "A Numerical Method for Treating Fluid Flow in the Presence of Shocks," Los Alamos Scientific Laboratory, Rept. LA-1930 (1955).

Lee, E.L. and Tarver, C.M., "Phenomenological Model of Shock Initiation in Heterogeneous Explosives," Phys. of Fluids, **23**, 2362 (1980).

Lemaitre, J., A Course on Damage Mechanics. Springer-Verlag, (1992).

Lemmen, P.P.M and Meijer, G.J., "Failure Prediction Tool Theory and User Manual," TNO Building and Construction Research, The Netherlands, Rept. 2000-CMC-R0018, (2001).

Lewis B.A., "Developing and Implementing a Road Side Safety Soil Model into LS-DYNA", FHWA Research and Development Turner-Fairbank Highway Research Center. Oct, 1999.

Librescu, L. Qin, Z., and Ambur D.R., "Implications of Warping Restraint on Statics and Dynamics of Elastically Tailored Thin-Walled Composite Beams," *International Journal of Mechanical Sciences*, 45, 1247-1267 (2003).

Liu, Joseph W. H., "Modification of the Minimum-Degree Algorithm by Multiple Elimination," ACM Transactions on Mathematical Software, **11**, 141-153 (1985).

Liu, W.K. and Belytschko, T., "Efficient Linear and Nonlinear Heat Conduction with a Quadrilateral Element," Int. J. Num. Meths. Engrg., **20**, 931–948 (1984).

Liu, W.K., Belytschko, T., Ong, J.S.J. and Law, E., "Use of Stabilization Matrices in Nonlinear Finite Element Analysis," Engineering Computations, **2**, 47–55 (1985).

Liu, W.K., Lecture Notes on Nonlinear Finite Element Methods, Northwestern University, (1983, revised in 1992).

Liu, W.K., Guo, Y., Tang, S. and Belytschko T., "A Multiple-Quadrature Eight-node Hexahedral Finite Element for Large Deformation Elastoplastic Analysis," Comput. Meths. Appl. Mech. Engrg., **154**, 69–132 (1998).

Liu, W.K., Hu, Y.K., and Belytschko, T., "Multiple Quadrature Underintegrated Finite Elements," Int. J. Num. Meths. Engrg., **37**, 3263–3289 (1994).

Liu, W. K., Chang, H., and Belytschko, T., "Arbitrary Lagrangian-Eulerian Petrov-Galerkin Finite Elements for Nonlinear Continua," Computer Methods in Applied Mechanics and Engineering, to be published.

Liu, W.K., Jun, S. and Zhang, Y.F., "Reproducing Kernel Particle Method," Int. J. Numer. Methods Fluids **20**, 1081-1106 (1995).

Lucy, L.B., "Numerical Approach to Testing the Fission Hyphothesis," Astron. J. **82**, 1013-1024 (1977).

Lysmer, J. and Kuhlemeyer, R.L., "Finite Dynamic Model for Infinite Media", J. Eng. Mech. Div. ASCE, 859-877 (1969).

MacNeal R.H. and Harder R.L., "A Proposed Standard Set of Problems to Test Finite Element Accuracy," Finite Elements Anal. Des., (1985, 3–20 (1985).

Maenchen, G. and Sack, S., "The Tensor Code," Meth. Comp. Phys. **3**, (Academic Press), 181-263 (1964).

Maffeo, R., "TRANCITS Program User's Manual," General Electric Company, Cincinnati, OH, NASA Report CR-174891, (1985).

Maffeo, R., "Burner Liner Thermal/Structural Load Modeling," General Electric Company, Cincinnati, OH, NASA Report CR-174892, (1984).

Maker, B. N., Private communication, Lawrence Livermore National Laboratory. Dr. Maker programmed and implemented the compressible Mooney-Rivlin rubber model.

Marchertas, A. H., and Belytschko,T. B., "Nonlinear Finite Element Formulation for Transient Analysis of Three Dimensional Thin Structures," Report ANL-8104, LMFBR Safety, Argonne National Laboratory, Argonne, IL (1974).

Margolin, L. G., personal communication to D. Benson (1989).

Maskew, B., "Program VSAERO Theory Document," NASA CR-4023 (1987).

Matthies, H., and G. Strang, The Solution of Nonlinear Finite Element Equations, Int., Journal for Numerical Methods in Engineering, **14**, No. 11, 1613-1626.

Matzenmiller, A., and Schweizerhof, K., "Crashworthiness Simulatios of Composite Structures - A First Step with Explicit Time Integration," Nonlinear Computational mechanics - A State of the Art, edited by P.W. Wriggers, et. al., Springer-Verlag, (1991).

Mauldin, P.J., R.F. Davidson, and R.J. Henninger, "Implementation and Assessment of the Mechanical=Threshold=Stress Model Using the EPIC2 and PINON Computer Codes," Report LA-11895-MS, Los Alamos National Laboratory (1990).

McGlaun, personal communication to D. Benson, Sandia National Laboratories, (1990).

McGlaun, J. M., Thompson, S. L., and Elrick, M. G., "CTH: A Three-Dimensional Shock Wave Physics Code," Proc. of the 1989 Hypervelocity Impact Symposium (1989).

Mindlin, R.D., "Influence of Rotary Inertia and Shear on Flexural Motions of Isotropic, Elastic Plates," J. Appl. Mech. **18**, 31-38 (1951).

Mizukami, A., and Hughes, T. J. R., "A Petrov-Galerkin Finite Element Method for Convection Dominated Flows: An Accurate Upwinding Technique for Satisfying the Maximum Principle," Computer Methods in Applied Mechanics and Engineering, Vol. 50, pp. 181-193 (1985).

Monaghan, J.J., and Gingold, R.A., "Shock Simulation by the Particle Method of SPH," Journal of Computational Physics, **52**, 374-381 (1983).

Murray Y.D., "Modeling Rate Effects in Rock and Concrete", Proceedings of the 8th International Symposium on Interaction of the Effects of Munitions with Structures, Defense Special Weapons Agency, McLean VA, USA, (1997).

Murray, Y. D., "Modeling Rate Effects in Rock and Concrete," Proceedings of the 8th International Symposium on Interaction of the Effects of Munitions with Structures, Defense Special Weapons Agency, McLean, VA, USA, (1997).

Murray, Y.D., Users Manual for Transversely Isotropic Wood Model**,** APTEK technical report to FHWA (to be published), (2002).

Murray, Y.D., and J. Reid, Evaluation of Wood Model for Roadside Safety Applications, APTEK technical report to FHWA (to be puslished), (2002).

Nagtegaal, J.C., Parks, D.M., and Rice J.R., "On Numerically Accurate Finite Element Solution in the Fully Plastic Range", Computer Methods in Applied Mechanics and Engineering, **4**, 153 (1974).

Newman, W.M., and Sproull, R.F., Principles of Interactive Computer Graphics, McGraw-Hill, New York (1979).

Newmark, N., "A Method of Computation for Structural Dynamics," Journal of the Engineering Mechanics Division, Proceedings of the American Society of Civil Engineers, 67-94 (1959).

Neilsen, M.K., Morgan, H.S., and Krieg, R.D., "A Phenomenological Constitutive Model for Low Density Polyurethane Foams," Rept. SAND86-2927, Sandia National Laboratories, Albuquerque, NM (1987).

Noh, W.F., Meth. Comp. Phys. **3**, (Academic Press) (1964).

Noh, W.F., "Numerical Methods in Hydrodynamic Calculations," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-52112 (1976).

Oden, J.T, P. Devloo, and T. Strouboulis, "Adaptive Finite Element Methods for the Analysis of Inviscid Compressible Flow: Part I. Fast Refinement/Unrefinement and Moving Mesh Methods for Unstructured Meshes," *Computer Methods in Applied Mechanics and Engineering*, **59**, 327-362 (1986).

Ogden, R.W., *Non-Linear Elastic Deformations*, Ellis Horwood Ltd., Chichester, Great Britian (1984).

Okuda T., Yamamae Y. and Yasuki T., Request for MAT126 Modification, Microsoft Power Point presentation, Toyota Communication Systems and Toyota Motor Corporation, 2003.

Oliver, J., "A Consistent Characteristic Length of Smeared Cracking Models," *International Journal for Numerical Methods in Engineering*, **28**, 461-474 (1989).

Rajendran, A.M.,"Material Failure Models", in Material Behavior at High Strain Rates, Course Notes, Computational Mechanics Associates, Monterey, CA, 424 (1989).

Papadrakakis, M., "A Method for the Automated Evaluation of the Dynamic Relaxation Parameters," Comp. Meth. Appl. Mech. Eng. **25**, 35-48 (1981).

Pian, T.H.H., and Sumihara, K., "Rational Approach for Assumed Stress Elements," Int. J. Num. Meth. Eng., **20**, 1685-1695 (1984).

Prokic, A., "New Warping Function for Thin-Walled Beams. I: Theory," *Journal of Structural Engineering*, 122, 1437-1442, (1994).

Przemieniecki, J. S., Theory of Matrix Structural Analysis, McGraw-Hill Book Company, New York, NY (1986).

Reid, S.R. and C. Peng, "Dynamic Uniaxial Crushing of Wood," Int. J. Impact Engineering, Vol. 19, No. 5-6, pp. 531-570, (1997).

Reyes, A., O.S. Hopperstad, T. Berstad, and M. Langseth, "Implementation of a Material Model for Aluminium Foam in LS-DYNA", Report R-01-02, Restricted, Department of Structural Engineering, Norwegian University of Science and Technology, (2002).

Richards, G.T., "Derivation of a Generalized von Neumann Pseudo-Viscosity with Directional Properties," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-14244 (1965).

Richtmyer, R.D., and Morton, K.W., Difference Equations for Initial-Value Problems, Interscience Publishers, New York (1967).

Sandler, I.S., and D. Rubin, "An Algorithm and a modular subroutine for the cap model," Int. J. Numer. Analy. Meth. Geomech., **3,** 173-186 (1979).

Schwer, L.E., Cheva, W., and Hallquist, J.O., "A Simple Viscoelastic Model for Energy Absorbers Used in Vehicle-Barrier Impacts," In Computational Aspects of Contact Impact and Penetration, Kulak, R.F., and Schwer, L.E., Editors, Elmepress International, Lausanne, Switzerland, 99-117 (1991).

Shapiro, A.B., "TOPAZ3D - A Three Dimensional Finite Element Heat Transfer Code," University of California, Lawrence Livermore National Laboratory, Report UCID-20481 (1985).

Shapiro, A.B., "REMAP: a Computer Code That Transfers Node Information Between Dissimilar Grids," Lawrence Livermore National Laboratory, UCRL-ID-104090, (1990).

Shkolnikov, M.B. Private Communication (1990-1991).

Simo, J.C., and Govindjee, S., ``Exact Closed-Form Solution of the Return Mapping Algorithm in Plane Stress Elasto-Viscoplasticity," *Engineering Computations*, **5**, 254-258 (1988).

Simo J.C., J. W. Ju, "Stress and Strain Based Continuum Damage Models", Parts I & II, Int. J. of Solids and Structures, Vol 23, No 7, (1987).

Simo, J. C., J.W. Ju, K.S. Pister, and R.L. Taylor, "An Assessment of the Cap Model: Consistent Return Algorithms and Rate-Dependent Extension," J. Eng. Mech., **114**, No. 2, 191-218 (1988a).

Simo, J. C., J.W. Ju, K.S. Pister, and R.L. Taylor, "Softening Response, Completeness Condition, and Numerical Algorithms for the Cap Model," Int. J. Numer. Analy. Meth. Eng. (1990).

Simo, J.C. and Taylor, R.L., "A Return Mapping Algorithm for Plane Stress Elastoplasticity," *International Journal for Numerical Methods in Engineering*, **22**, 649-670 (1986).

Steinberg, D.J. and Guinan, M.W., "A High-Strain-Rate Constitutive Model for Metals," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-80465 (1978).

Stillman, D. W., and Hallquist, J. O., "LS-INGRID: A Pre-Processor and Three-Dimensional Mesh Generator for the Programs LS-DYNA, LS-NIKE3D, and TOPAZ3D," Version 3.1, Livermore Software Technology Corporation (1992).

Stojko, S., privated communication, NNC Limited, Engineering Development Center (1990).

Stolarski, H., and Belytschko, T., "Shear and Membrane Locking in Curved Elements," Comput. Meths. Appl. Mech. Engrg., **41**, 279–296 (1983).

Stone, C.M., Krieg, R.D., and Beisinger, Z.E., "Sancho, A Finite Element Computer Program for the Quasistatic, Large Deformation, Inelastic Response of Two-Dimensional Solids," Sandia Report, SAND 84-2618, UC-32, Albuquerque, NM (1985).

Storakers, B., "On Material Representation and Constitutive Branching in Finite Compressible Elasticity", Royal Institute of Technology, Stockholm Sweden, (1985).

Sturt, R.M.V., and B.D. Walker, J.C. Miles, A. Giles, and N. Grew, "Modelling the Occupant in a Vehicle Context-An Integrated Approach," 13th International ESV Conference, Paris, November 4-7 (1991).

Sussman, T. and Bathe, K.J., "A Finite Element Formulation for Nonlinear Incompressible Elastic and Inelastic Analysis," *Computers & Structures*, **26**, Number 1/2, 357-409 (1987).

Tarver, C.M., and Hallquist, J.O., "Modeling of Two-Dimensional Shock Initiation and Detonation Wave Phenomena in PBX 9404 and LX-17," University of California, Lawrence Livermore National Laboratory, Rept. UCID84990 (1981).

Taylor, R.L. and Simo, J.C. Bending and membrane elements for the analysis of thick and thin shells, *Proc. of NUMETA Conference*, Swansea, 1985.

Taylor, R.L., Finite element anlysis of linear shell problems, in Whiteman, J.R. (ed.), *Proc. of the Mathematics in Finite Elements and Applications*, Academic Press, New York, 191-203 (1987).

Taylor, L.M., and Flanagan, D.P., "PRONTO3D A Three-Dimensional Transient Solid Dynamics Program," Sandia Report: SAND87-1912, UC-32 (1989).

Thompson, S. L., "CSQ -- A Two Dimensional Hydrodynamic Program with Energy Flow and Material Strength," Sandia Laboratories, SAND74-0122 (1975).

Thompson, R., L., and Maffeo, R. L., "A Computer Analysis Program for Interfacing Thermal and Structural Codes," NASA Lewis Research Center, Report NASA-TM-87021 (1985).

Trefethen, L.N., "Group Velocity in Finite Difference Schemes," SIAM Review, **24**, No. 2 (1982).

Tsai, S.W., and Wu, E.M., "A General Theory of Strength for Anisotropic Materials," Journal of Composite Materials, 58-80 (1971).

Tuler, F.R. and B.M. Butcher, "A Criterion for the Time Dependence of Dynamic Fracture," *The International Journal of Fracture Mechanics*, **4**, No. 4 (1968).

Turkalj, G. Brnic, J., and Prpic-Orsic J., "Large Rotation Analysis of Elastic Thin-Walled Beam-Type Structures Using ESA Approach," *Computers & Structures*, 81, 1851-1864, (2003).

Underwood, P., "Dynamic Relaxation," Computational Method for Transient Analysis, Belytschko, T., and Hughes, T.J.R., Eds., **1**, 245-263, (1986).

Van Leer, B., "Towards the Ultimate Conservative Difference Scheme. IV. A New Approach to Numerical Convection," Journal of Computational Physics, **23**, 276-299 (1977).

Vawter, D., "A Finite Element Model for Macroscopic Deformation of the Lung," published in the Journal of Biomechanical Engineering, **102**, 1-7, (1980).

von Neumann, J., and Richtmyer, R.D., "A Method for the Numerical Calculation of Hydrodynamical Shocks," J. Appl. Phys., **21**, 232 (1950).

Walker, B.D., and P.R.B. Dallard, "An integrated Approach to Vehicle Crashworthiness and Occupant Protection Systems", SAE International Congress and Exposition, Detroit, Michigan, (910148), February 25-March 1 (1991).

Walker, H.F., Numerical Solution of Nonlinear Equations, University of California, Lawrence Livermore National Laboratory, Rept. UCID-18285 (1979).

Wang, J. T., and O. J. Nefske, "A New CAL3D Airbag Inflation Model," SAE paper 880654, (1988).

Wang, J. T., private communication (1992).

Wang, J.T., "An Analytical Model for an Airbag with a Hybrid Inflator", Publication R&D 8332, General Motors Development Center, Warren, MI (1995).

Wang, J.T., "An Analytical Model for an Airbag with a Hybrid Inflator", AMD-Vol. 210, BED **30**, ASME, 467-497 (1995).

Warsi, Z. U. A., "Basic Differential Models for Coordinate Generation," in Symposium on the Numerical Generation of Curvilinear Coordinate Systems, Nashville, Tenn. (1982).

Whirley, R. G., Hallquist, J. O., and Goudreau, G. L., "An Assessment of Numerical Algorithms for Plane Stress and Shell Elastoplasticity on Supercomputers," *Engineering Computations*, **6**, 116-126 (June, 1989).

Wilkins, M.L., "Calculations of Elastic Plastic Flow," Meth. Comp. Phys., **3**, (Academic Press), 211-263 (1964).

Wilkins, M. L., "Calculation of Elastic-Plastic Flow," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-7322, Rev. I (1969).

Wilkins, M.L., "Use of Artificial Viscosity in Multidimensional Fluid Dynamic Calculations," J. Comp. Phys. **36**, 281 (1980).

Wilkins, M.L., Blum, R.E., Cronshagen, E., and Grantham, P., "A Method for Computer Simulation of Problems in Solid Mechanics and Gas Dynamics in Three Dimensions and Time," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-51574 (1974).

Wilson, E.L., Three Dimensional Static and Dynamic Analysis of Structures, A publication of Computers and Structures, Inc., Berkeley, California, [1996-2000].

Winslow, A. M., "Equipotential Zoning of Two-Dimensional Meshes," Lawrence Radiation Laboratory, UCRL-7312 (1963).

Winslow, A. M., "Equipotential Zoning of The Interior of a Three-Dimensional Mesh," Report to LSTC (1990).

Winters, J.M., "Hill-based muscle models: A systems engineering perspective," In Multiple Muscle Systems: Biomechanics and Movement Organization, JM Winters and SL-Y Woo eds, Springer-Verlag (1990).

Winters J.M. and Stark L., "Estimated mechanical properties of synergistic muscles involved in movements of a variety of human joints,": J Biomechanics 21:1027-1042, (1988).

Woodruff, J.P., "KOVEC User's Manual," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-51079 (1973).

Yen, C.F. and Caiazzo, A, "Innovative Processing of Multifunctional Composite Armor for Ground Vehicles," ARL-CR-484, U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, (2001).

Yunus, S.M., Pawlak, T.P., and Cook, R.D., " Solid Elements with Rotational Degrees of Freedom, Part I-Hexahedron Elements," To be published, (1989).

Zajac F.E., "Muscle and tendon: Properties, models, scaling, and application to biomechanics and motor control, "CRC Critical Reviews in Biomedical Engineering 17(4):359-411, (1989).