Nonlinear Finite Element Modeling and Simulation

CE 264

Dhafer Marzougui
Pradeep Mohan

Week 1

- General FEA Process
- LS-DYNA Background
  - History
  - Capabilities
  - Application
- FEM Analysis Procedures
  - Pre-process : Model Setup
  - Solution-procedure: Solver
  - Post-process - Analysis
- LS-DYNA Input Format
  - Keyword Format
  - Structural Format
- Example of an Input Deck
- Computer Session

General FEA Process

- Model Development - Pre-processing
  - Discretize Geometry: Nodes/Elements
  - Geometry properties: Thickness/Cross-section
  - Material properties
  - Loading conditions
  - Constraints
  - Boundary conditions
- Solver - Solution processing
  - Numerical solution of equations of motion

General FEA Process

- Post-processing: Results Analysis
  - Deformed geometry
  - Displacements, velocities, accelerations
  - Stress and strain
  - Reaction forces
  - Energies
- FE Model Improvement
  - Update Model based on the analysis results
  - Iterative process until objectives achieved

General FEA Process

- Model Development - Pre-processing
  - LS-INGRID, FEM-B
  - I-DEAS, True-Grid, EasiCrash
  - PATRAN, HyperMesh
- Solver - Solution processing
  - LS-DYNA, PamCrash, RADIOSS
  - NASTRAN, ANSYS, Algor
- Results Analysis - Post-processing
  - LS-TAURUS, LS-POST
  - HyperMesh

Background and History of LS-DYNA

- 1976
  - DYNA3D developed at Lawrence Livermore National Laboratory by John Hallquist
  - Low velocity impact of heavy, solid structures, military applications
- 1979
  - DYNA3D ported on Cray-1
  - Improved sliding interface
  - Order of magnitude faster
- 1981
  - New material models - *Explosive-structure*, *Soil-structure*
  - Impacts of penetration projectiles

Background and History of LS-DYNA

- 1986
  - Beams, Shells, Rigid Bodies
  - Single Surface Contact
  - Support for Multiple Computer Platforms
- 1988
  - Automotive Applications Support
  - LS-DYNA
- 1989
  - Full Commercial Version
  - LSTC

Background and History of LS-DYNA

- 1993
  - Keyword Format
  - Automatic Single Surface Contact
  - 1st International LS-DYNA User Conference
- 1995
  - Training Lab Established at West Coast - LSTC
- 1997
  - Training Class Started at East Coast - NCAC/GWU
- Today
  - Release of Version LS970, Many New Features

General Capabilities

- Transient dynamics
- Quasi-static simulations
- Flexible and rigid bodies
- Nonlinear material behavior
- More than 80 constitutive relationships
- More than 40 element formulation
- Finite strain and finite rotation
- General contact algorithm
- Thermal Analysis
- Explicit and implicit analyses

General Capabilities

- Pre-stress and Post-stress (LS-NIKE3D)
- Interactive graphics
- Preprocessor - LS-INGRID
- Third party interfaces
- Postprocessor - LS-Taurus, LS-Post
- Other rigid body program coupling
- CAD data interface

Applications

- Automotive, train, ship, and aerospace crashworthiness
- Sheet and bulk forming process simulation
- Engine blade containment and bird strike analysis
- Seismic safety simulation
- Weapons design and explosive detonation simulation
- Biomechanics simulation
- Industrial accidents simulation
- Drop and impact analysis of consumer product
- Roadside Hardware Analysis
- Virtual proving ground simulation

LS-DYNA Input File Format

- Structured Input Format
  - Original Format
  - Organized by Entities
  - Fixed Format
- Keyword Input Format
  - Started 1993
  - More Flexible
  - Easy to Modify Input Deck

Keyword Format

- Keyword format input file
- List of Keyword options
- Examples

Keyword Format Input File

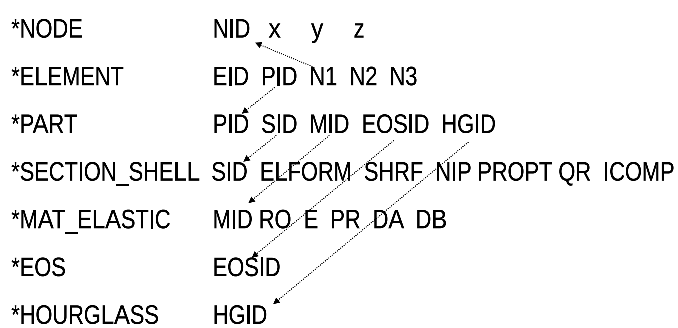
- Sections
  - Control, Material, Equation of State, Element, Parts, etc.
- The "*" followed by keyword indicate beginning of a section block.
- The "$" used for Comment Cards
- Data blocks begin with keyword followed by data pertaining to the keyword
- Multiple Blocks with the same keyword are permissible
- Material and Contact types are defined by name
- Keywords are alphabetically organized in manual

Keyword Format Input File

```
*KEYWORD
*TITLE
SAMPLE INPUT FILE
*CONTROL_TERMINATION
 0.1000000         0 0.0000000         0 0.0000000
*DATABASE_BINARY_D3PLOT
 1.00000-3                 0
*DATABASE_BINARY_D3THDT
 1.00000-3
*MAT_ELASTIC
        1 7.89000-9 2.00000+5 0.3000000
*SECTION_SOLID
        1        0
*SECTION_SHELL
        1        2
 1.0000000 1.0000000 1.0000000 1.0000000 0.0000000
*PART
PART NAME 1
        1        1        1        0        0        0        0        0
```

```
*NODE
       1 0.000000000E+00 0.000000000E+00 0.000000000E+00
       2 7.000000000E+00 0.000000000E+00 0.000000000E+00
       3 0.000000000E+00 7.000000000E+00 0.000000000E+00
       4 7.000000000E+00 7.000000000E+00 0.000000000E+00
       5 0.000000000E+00 0.000000000E+00 7.000000000E+00
       6 7.000000000E+00 0.000000000E+00 7.000000000E+00
       7 0.000000000E+00 7.000000000E+00 7.000000000E+00
       8 7.000000000E+00 7.000000000E+00 7.000000000E+00
*ELEMENT_SOLID
       1       1       1       2       4       3       5       6       8       7
*PART
PART NAME 1
        2        2        2        0        0        0        0        0
*ELEMENT_SHELL
       1       2       1       2       4       3
*END
```


Keyword Format Input File



LS-DYNA Execution

- Command Line

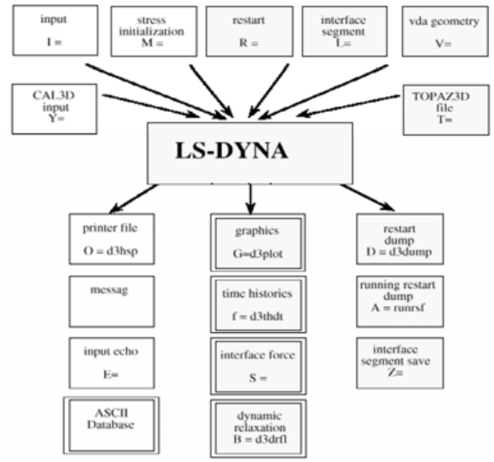
LS-DYNA **I=inf** O=**otf** G=**ptf** D=**dpf** F=**thf** U=**xtf** T=**tpf** A=**rrd** M=**sif** J=**jif** S=**iff** Z=**isf1** L=**isf2** B=**rlf** W=**root** E=**efl** X=**scl** C=**cpu** K=**kill** V=**vda** Y=**c3d** **{KEYWORD}** **{THERMAL}** **{COUPLE}** **MEMORY=nwds** where

- Example

  ls-dyna i=inputfile

  ls940 r=d3dump01 memory=12000000

LS-DYNA Execution



LS-DYNA Output Files

- d3hsp
- message
- d3plot,d3plot01,…
- d3thdt,d3thdt01, …
- d3dump01, …
- runrsf
- Ascii files (glstat, nodout, deforc, ..etc)

Week 2

- Detail Capabilities of Keyword Format
- Explicit FEM Theory
- Computer session

*AIRBAG

```
$
$
*AIRBAG_SIMPLE_AIRBAG_MODEL_1
       108         1         0 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
 7.17000+8 1.00400+9 300.00000         1 0.7000000 0.0000000 0.1000000 1.2040-12
         0
```


*AIRBAG

- Control volumes
- Thermodynamic properties for the airbag inflator models
- Tires
- Pneumatic dampers
- Biomechanic parts

*BOUNDARY

- Fixed (SPC’s)
- Prescribed motion
- Thermal

*CONSTRAINTED

Constraints within a structure between structural parts

- Nodal rigid bodies
- Rivets
- Welds
- Linear constraints
- Tying a shell edge to a shell edge with failure
- Merging rigid bodies
- Adding extra nodes to rigid bodies
- Rigid body joints

*CONTACT

- 24 different contact types
- Deformable to deformable bodies
- Single surface contact in deformable bodies
- Deformable body to a rigid body
- Tying deformable structures (strain failure)
- Modeling rebar in concrete structures (*CONTACT_ID)

*CONTACT_ENTITY

Analytical rigid surface to deformable structure

- Metal forming
  - The punch and die surface geometry can be input as VDA surfaces which are treated as rigid.
- Occupant Modeling
  - Treat contact between rigid body occupant dummy hyper ellipsoids and deformable structures such as airbags and instrument panels.
- Motion governed by rigid body mechanics or prescribed translation and/or rotation (6 DOF)

*CONTROL

- Termination time
- Hourglass type
- Contact penalty scale factor
- Shell element formulation
- Numerical damping

*DATABASE

- Controlling output
- ASCII files
- Binary files

*DEFINE

- Curves
- Boxes to limit geometric extent
- Local coordinate systems
- Vectors
- Inputs to other options

*DEFORMABLE_TO_RIGID

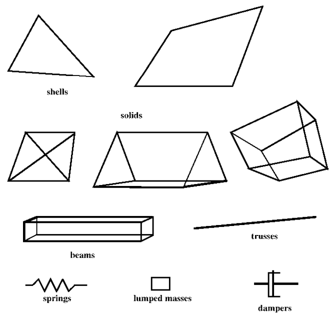
- Switch materials that are defined as deformable to rigid at the start of the analysis
- Cost effective method for simulating events such as vehicle rollover

*ELEMENT

- Beams
- Concentrated masses
- Dampers
- Seat belts
- Shells
- Solids
- Springs
- Thick Shells

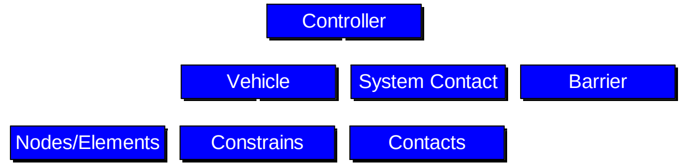
Basic Elements

- Shells
- Solids
- Beams/Trusses
- Discrete Elements



*INCLUDE

- Split files into subfiles
- Split subfiles into sub-subfiles, and so on



*INITIAL

- Initial velocities
- Detonation and Momentum
- Initial stresses
- Initial temperatures

*LOAD

- Concentrated point loads
- Distributed pressures
- Body force loads
- Variety of thermal loadings

*MAT

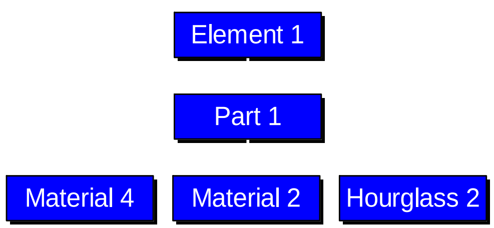
- Constitutive constants for all material models
- 80+ structural materials
  - stress-strain relations
- 8 spring/damping materials
  - F-d and F-v relationships
- 1 seat belt material
- 6 thermal materials

*NODE

- Nodal point identifiers
- x, y, z coordinates
- Translational constraint
- Rotational constraint

*PART

- Relates parts ID between elements, sections, material and hourglass control
- For rigid material, rigid body inertia properties and initial conditions can be specified through the *PART command



*RIGIDWALL

- Planar
- Rectangular prism
- Cylindrical prism
- Spherical
- Stationary or moving
- Finite or infinite
- Multiple walls can be defined to model combinations of geometric shapes
- Friction

*SECTION

- Element type dependent
  - element formulation
  - integration rule
  - thickness or cross-section properties
- Used in *PART

*SET

- Concept of grouping nodes, elements, parts, etc.
- Examples
  - output acceleration, velocity, displacement for a set of nodes
  - set of shell elements as slaves for a contact definition
  - define a cross section with a set of nodes and a set of shells
  - single surface contract for all parts specified in a set of parts

Other Keywords

*DAMPING

*HOURGLASS

*INTEGRATION

*TITLE

*TRANSLATE

*USER_INTERFACE

Keyword Examples

- Control Output

```
$ Termination Time
$
*CONTROL_TERMINATION
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$   endtim    endcyc     dtmin    endneg    endmas
      6.01         0       0.0       0.0       0.0
$
$ Energy Computation
$
*CONTROL_ENERGY
$        i         i         i         i
$     hgen      rwen    slnten     rylen
         2         2
$
$
```

Keyword Examples

◆ Control Output

```
$
$ Time interval between state dumps (D3PLOT)
$
*DATABASE_BINARY_D3PLOT
$  DT/CYCL       LCDT
       1.0
$
*DATABASE_EXTENT_BINARY
$        i         i         i         i         i         i         i         i
$    neiph     neips    maxint    strflg    sigflg    epsflg    rltflg

$        i         i         i
$   cmpflg    ieverp    beamip
                   1
$
$ Time history data interval (D3THDT)
$
*DATABASE_BINARY_D3THDT
$  DT/CYCL       LCDT
    999999
$
```


Keyword Examples

◆ Control Output

```
*DATABASE_RWFORC
$  DT/CYCL
       0.1
$
*DATABASE_HISTORY_NODE
$   Define nodes that output into NODOUT
$      id1       id2       id3       id4       id5       id6       id7       id8
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
     99999       414       486
$
*DATABASE_NODOUT
$  DT/CYCL
       0.1
*DATABASE_GLSTAT
$  DT/CYCL
       0.1
*DATABASE_MATSUM
$  DT/CYCL
       0.1
*DATABASE_SLEOUT
$  DT/CYCL
       0.1
```


Keyword Examples

◆ Contact Definitions

```
$
$ Define Contacts - Sliding Interfaces
$
*CONTACT_AUTOMATIC_SINGLE_SURFACE
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$     ssid      msid     sstyp     mstyp    sboxid    mboxid       spr       mpr
         0
$  Equating ssid to zero means that all segments are included in the contact
$
$       fs        fd        dc        vc       vdc    penchk        bt        dt
      0.08      0.08
$
$      sfs       sfm       sst       mst      sfst      sfmt       fsf       vsf

$
```


Keyword Examples

◆ Material and Parts

```
$
$ Define Materials and Parts
$
*MAT_PIECEWISE_LINEAR_PLASTICITY
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      mid        ro         e        pr      sigy      etan      eppf      tdel
         1 7.830E-06     200.0       0.3     0.207               0.750
$
$        c         p      lcss      lcsr
        40         5
$ PLASTIC STRESS/STRAIN CURVES
$     eps1      eps2      eps3      eps4      eps5      eps6      eps7      eps8
$      es1       es2       es3       es4       es5       es6       es7       es8
     0.000     0.080     0.160     0.400     0.750
     0.207     0.250     0.275     0.290     0.300
$
```

Keyword Examples

- Material and Parts

```
$ PART DEFENITION
$
*PART
$   heading
corner1
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      pid       sid       mid     eosid      hgid     igrav    adpopt
         1         1         1
$
$
$ SHELL ELEMENT CROSS-SECTIONAL PROPERTIES
$
*SECTION_SHELL
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      sid    elform      shrf       nip     propt   qr/irid     icomp
         1         2              3.0000
$       t1        t2        t3        t4      nloc
      2.00      2.00      2.00      2.00
$
```


Keyword Examples

- Nodes and Elements

```
$
$ NODAL POINT CARDS
$
*NODE
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$    NID               X               Y               Z      TC      RC
   99999             0.0             0.0           274.0       0       0
       1 -5.00000000E+01 -4.80000000E+01 0.000000000E+00       0       0
       2 -4.16667000E+01 -4.80000000E+01 0.000000000E+00       0       0
     715 -5.80000000E+01 -2.40000000E+01 2.724830000E+02       0       0
     716 -5.80000000E+01 -3.20000000E+01 2.724830000E+02       0       0
$
$ SHELL ELEMENTS
$
*ELEMENT_SHELL
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$    EID     PID      N1      N2      N3      N4
       1       1       1       2       9       8
       2       1       2       3      10       9
     640       1     710     711     716     715
     641       1     711     485     487     716
$
```


Keyword Examples

- Constraints, Boundary, and Initial Conditions

```
$
$ Define Constraints, Boundary & Initial Conditions
$
$
*BOUNDARY_SPC_NODE
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      NID CID X Y Z RX RY RZ
         1,  0,1,1,1, 1, 1, 1
         2,  0,1,1,1, 1, 1, 1
         3,  0,1,1,1, 1, 1, 1
       252,  0,1,0,0, 0, 1, 1
       259,  0,1,0,0, 0, 1, 1
$
```


Keyword Examples

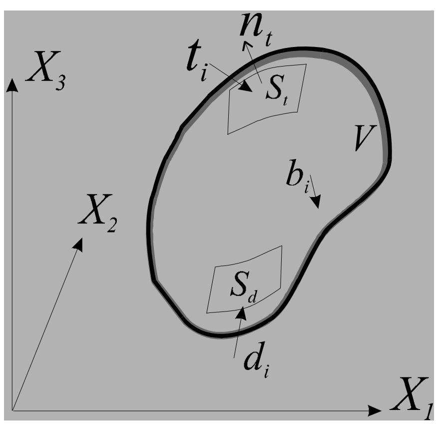
- Rigid Walls

```
$
$  Define Stone Walls
$
*RIGIDWALL_PLANAR_MOVING_FORCES
$.../..../..../..../..../..../..../..../....1..../....2..../....3..../....4..../....5..../....6..../....7..../....8
$     nsid    nsidex     boxid
         0         0         0
$
$       xt        yt        zt        xh        yh        zh      fric
       0.0       0.0     274.0       0.0       0.0       0.0       1.0
$
$  SW MASS    SW VEL
   800.000   8.94000
$
$     SOFT      SSID     NODE1     NODE2     NODE3     NODE4
         0         0     99999
$
```

Finite Element Method Basic Theory

- Equations of Equilibrium
- Time Integration Loop
- Element Formulation
  - Solid
  - Shell
  - Beam
- Time Integration Schemes
  - Implicit
  - Explicit
- Nonlinear Problems

Equations of Equilibrium



Equations of Equilibrium

$\sigma_{ij,j} + \rho f_i = \rho \ddot{x}_i$	Momentum Equation
$\sigma_{ij} n_i = t_i(t)$	Boundary Condition - Traction
$x_i(X_\alpha, t) = D_i(t)$	Boundary Condition - Displacement
$(\sigma_{ij}^+ - \sigma_{ij}^-) n_i = 0$	Boundary Condition - Contact

Equation of Equilibrium

- Implicit Formulation

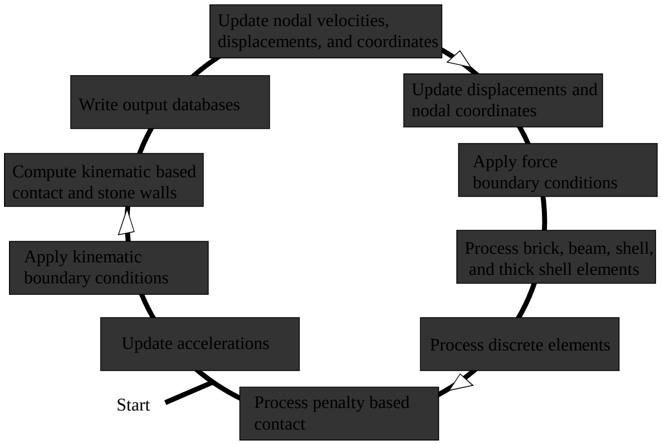
$$[M]\{\ddot{x}\} + [C]\{\dot{x}\} + [K]\{x\} = [F_{external}]$$

- Explicit Formulation

$$[M]\{\ddot{x}\} = [F_{external}] - [F_{internal}]$$

  - Internal forces include the damping, stiffness, contact forces, ..etc.

Time Integration Loop



Week 3

- Initial Conditions
- Boundary Conditions
- Loads
- Rigid Walls
- Constraints

Week 3

- Assume we have a model with the following defined:
  - nodes, elements,materials,properties,parts
- What can we do to the model?
  - Apply initial conditions, boundaries conditions, loads, constrains
  - We need to define: boxes, curves, sets, vectors
- What if parts collide or collapse on themselves?
  - use rigid walls , contacts
- How do we “debug” and/or analyze the model?
  - Output files

*SET - Nodes

- Define a group (set) of nodes
  - assign a set identification number (SID)
  - define the nodes to be included (NID)
- *SET_NODE_LIST
  - define the nodes 8 per line
- *SET_NODE_COLUMN
  - define the nodes 1 per line

*DEFINE_BOX

- Define a box-shaped volume
  - everything inside the box can be used as input
- Assign a box identification number (BOXID)
- Define two extreme corners of the box
  - *Xmin - Xmax*
  - *Ymin - Ymax*
  - *Zmin - Zmax*

*DEFINE_CURVE

- Define a (load) curve
- Assign a load curve identification number (CLID)
- Define the points of curve in pairs
  - Abscissa (x) - Ordinate (y)
- Scaling
- Offset
- Examples
  - Force vs Time
  - Velocity vs Time

*DEFINE_COORDINATE

- Define a local coordinate system
- Assign a coordinate system identification number (CID)
- *DEFINE_COORDINATE_NODES
  - 3 nodes: local origin, along local x-axis, in local x-y plane
- *DEFINE_COORDINATE_SYSTEM
  - x, y, z of three points (same as NODES)
- *DEFINE_COORDINATE_VECTOR
  - 2 vectors: local x-axis, local in-plane vector

*DEFINE_VECTOR

- Define a vector
- Assign a vector identification number (VID)
- Define tail (*xt*, *yt*, *zt*) and head (*xh*, *yh*, *zh*)

Initial Conditions

- Purpose - To set initial conditions
  - Detonation and momentum
  - Stresses
  - Temperature
  - Velocity
- Initial stresses, temperatures and velocities are equal ZERO by default
- Boundary conditions override initial conditions

*INITIAL_DETONATION and *INITIAL_MOMENTUM

- Simulation an impulsive type of loading
- Used for solid elements
- Detonation - lighting explosive materials (parts)

$$p(t) = p_0 e^{-t/tau}$$

- Momentum - depositing an initial momentum on an element

Initial Velocity

- Assign initial translational and rotational velocities to nodes and bodies
- *INITIAL_VELOCITY
  - set of nodes
  - exclude a set of nodes
  - all nodes within a defined box
- *INITIAL_VELOCITY_NODE
  - individual nodes

Initial Velocity

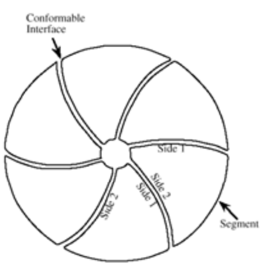
- *INITIAL_VELOCITY_GENERATION
  - for rotating and translating bodies
    - parts
    - set of parts
    - set of nodes
  - must be mutually exclusive from setting nodal initial velocities

Boundary Conditions

- Purpose - To define imposed motions on boundary Nodes
  - Convection, Flux, Radiation, Temperature
  - Cyclic
  - Non-reflecting, Sliding, Symmetry with Failure (Solids)
  - Prescribed Motion
  - SPC

*BOUNDARY_CYCLIC

- Cyclic symmetry
- Define axis of rotation vector
  - x, y and z vectors
  - vectors must be global
- Define 2 boundary planes
- (using node sets)



Solid Elements Only

- *BOUNDARY_NON_REFLECTING
  - stress gradient at boundary equals zero
  - boundary moves with shock wave
- *BOUNDARY_SLIDING_PLANE
  - constrain a set of nodes to move on an arbitrary orientated plane or line
- *BOUNDARY_SYMMETRY_FAILURE
  - symmetry plane fails upon defined tensile failure stress

*BOUNDARY_PRESCRIBED_MOTION

- Impose nodal motion on a node, set of nodes, or a rigid body
- Applicable to one degree-of-freedom
- Motion
  - displacement
  - velocity
  - acceleration (nodes only)
- Motion prescribed by a curve

*BOUNDARY_SPC
Single Point Constraints

- Fix (constrain) one or more degrees-of-freedom of a node
- Individual nodes or a set of nodes
  - *BOUNDARY_SPC_NODE
  - *BOUNDARY_SPC_SET
- May be defined in a local coordinate system

Loads

- Purpose - To Define Applied “Forces”
  - Beam
  - Body
  - Heat, Temperature
  - Nodal and Rigid Body
  - Shell (pressure)

Loads

- Avoid single concentrated loads
  - physically unrealistic
  - induce hourglass modes
- Avoid instantaneous loading
- Require a load curve
- Loads can be scaled

*LOAD_BEAM

- Distributed traction load along any local axis (*r, s, t*)
- Beam or a set of beams
- Force per unit length

*LOAD_BODY

- Body force load due to a prescribed
  - base acceleration
  - angular velocity
- A single degree-of-freedom: *X*, *Y*, *Z*, *RX*, *RY*, *RZ*
- All nodes or a set of parts

*LOAD_NODE and *LOAD_RIGID_BODY

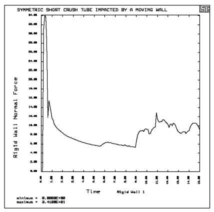
- Apply a load to a node, a set of nodes, or a rigid body
- x, y, or z force
- x, y, or z moment
- Follower force
  - force acts normal to a plane
- Global or local coordinate system

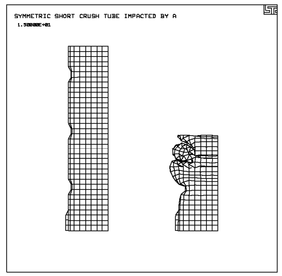
Distributed Pressure Load

- Apply a distributed pressure to a
  - segment (*LOAD_SEGMENT)
  - set of segments (*LOAD_SEGMENT_SET)
  - shell (*LOAD_SHELL_ELEMENT)
  - set of shells (*LOAD_SHELL_SET)
- Positive pressure acts in the negative normal direction of the shell/segment
- Arrival time of pressure

Rigid Walls

- Purpose - To Define Rigid Surfaces
- Simulate barriers, pendulums, crushers, etc.
- Nodes are prevented from penetrating a surface
- Wall energy in GLSTAT
- Wall forces in RWFORC



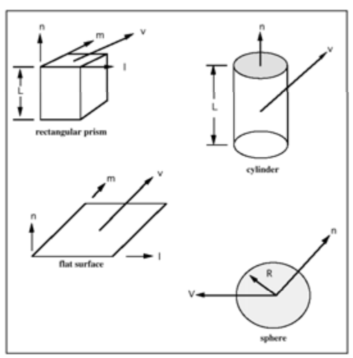


*RIGIDWALL_PLANAR

- Finite or infinite
- Motion condition
  - fixed
  - moving: mass and velocity
- Soft wall option
  - number of cycles to zero velocity
- Wall tracking with extra nodes
- Ortho
  - two separate friction coefficients normal to each other
  - example: rolling object - higher friction in transverse direction

*RIGIDWALL_GEOMETRIC

- Multiple geometric walls can be defined to model combinations of available geometric shapes



Constraints

- Purpose - Constrain Degree-of-Freedom Between Parts
  - rivets and welts
  - tying shells to shells
  - tying shells to solids
  - various rigid body constraints (will discuss later)
- Nodes must have mass
- Nodes cannot be subjected to multiple, independent, and possible conflicting constraints
- SPC's cannot conflict with constraints

*CONSTRAINED_RIVET

- Two-node rivet
- Rigid massless truss
- Acts “similar” to a pair of ball and socket joints
- Nodes can *not* be coincident

*CONSTRAINED_SPOTWELD

- Two-node spotweld
- Rigid massless beam
- No normal rotational stiffness transmitted from shells
- Nodes can not be coincident
- Failure criteria

  S_n = normal force at failure f_n = normal interface force

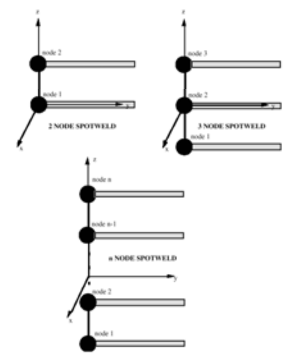
  S_s = shear force at failure f_s = shear interface force

*CONSTRAINED_GENERALIZED_WELD

- Options: SPOT, FILLET, BUTT
- Nodes may be coincident
- Output can be specified in a local coordinate system
- Failure criteria

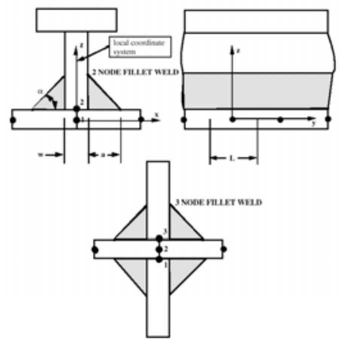
Spotweld

- Nodal ordering and orientation of the local coordinate system is important for determining spotweld failure

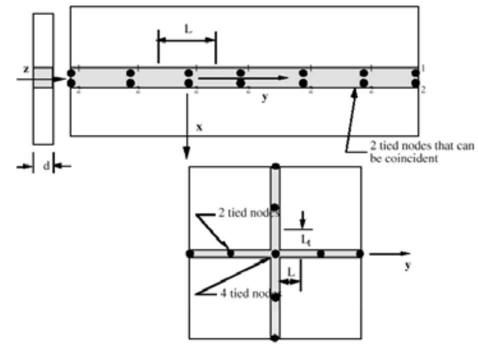


Fillet Weld

- Nodal ordering and orientation of the local coordinate system is shown for fillet weld



Butt Weld



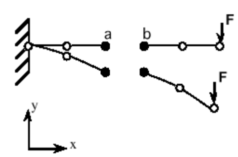
- Orientation of the local coordinate system and nodal ordering is shown

Weld Failure

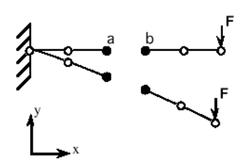
- ◆ Failure time
  - ❑ automatic failure at a specified time
- ◆ Ductile failure
  - ❑ due to plastic strain
  - ❑ effective nodal plastic strain > ε_{fail}^{p}
- ◆ Brittle failure
  - ❑ spotweld
  - ❑ fillet

Part Joining

- ◆ *CONSTRAINED_NODE_SET
  - ❑ translation constraints for 2 or more nodes
  - ❑ x, y, z, or any combination



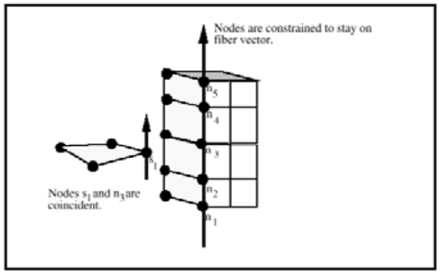
*CONSTRAINT_NODE_SET



*CONSTRAINT_NODAL_RIGID_BODY
*CONSTRAIN_SPOTWELD

Part Joining

- *CONSTRAINED_SHELL_TO_SOLID
  - define a tie between a shell edges and solid elements
  - shell nodes can be constrained to stay on fiber vector
  - node rigid bodies can perform the same function



Fracturing Elements

- *CONSTRAINED_TIE-BREAK
  - shell edge to shell edge interface
  - releases locally as a function of plastic strin
- *CONSTRAINED_TIED_NODES_FAILURE
  - tie nodes set (nodes must be coincident)
  - multiple nodes allowed (I.e., shells)
  - thin shells only
  - failure based on plastic strain

Other Part Joining Techniques

- Coincident Nodes (merging Nodes)
  - distorts geometry
  - no failure criteria
  - can not easily separate parts (e.g., for manipulation, re-meshing)
  - contact thickness violated
- Beams
  - more complex definition
  - effects time step calculation

Other Part Joining Techniques

- Nodal Rigid Bodies (more later)
  - motion governed by equations of dynamics
  - no failure criteria
  - rotations are allowed
- Contacts (more later)
  - tied (surface to surface, nodes to surface)
  - tiebreak (surface to surface, nodes to surface)

Week 4

- Time Integration, Time Step
- Computer Session

Execution Time Control

- For simple problems (mostly academic exercises) time control is relatively unimportant. For simulations that take hours or more, time step control become a significant factor
  - Explicit and Implicit Integration
  - Time Step Calculation
  - Execution Time
  - Controlling Time Step and Execution

Explicit and Implicit Integration

- A simple example: Spring-mass system
- Equation of motion

  $ma(t) + kx(t) = F(t)$

  $v(t) = dx(t)/dt$

  $a(t) = dv(t)/dt = d^2x(t)/dt$
- Numerical integration discretizes the differential equation into a step-by-step solution procedure

  at time t_n

  known: x_n, v_n, and F_n

  find: x_{n+1}

  Step forward: once values at t_{n+1} are known, calculate x_{n+2} and so on

Explicit Integration

- $\underline{M}a_n = \underline{P}_n - \underline{F}_n + \underline{H}_n$
  $\underline{M}$ = diagonal mass matrix
  $\underline{P}$ = external loads + body force
  $\underline{F}$ = internal force (stress divergence vector)
  $\underline{H}$ = hourglass resistance
- $a_n = \underline{M}^{-1}(\underline{P}_n - \underline{F}_n + \underline{H}_n)$
- $v_{n+1/2} = v_{n-1/2} + a_n \Delta t_n$
- $\Delta t_{n+1/2} = (\Delta t_n + \Delta t_{n+1})/2$
- $x_{n+1} = x_n + v_{n+1/2} \Delta t_{n+1/2}$

Explicit vs Implicit

Time Integration Method	Explicit	Implicit
Matrix Inversion	No	Yes
Computation	CPU Bound	I/O Bound
Convergence Criteria	Energy + Deformation	Residual
Time Integration	Central Difference	Newmark β Method
Time Step Size	Small (1E-6 s)	Larger (1E-3 s)
Duration	Millisecond	Second
Mass Matrix	Diagonal	
Numerical Stability	Very Good	Problem Dependent
Material Non-linearity	Linear to Highly NL	Linear to Moderate NL
Strain Rate	High (1E-1 to 1E+6)	Low (0 to 1E+1)
Rotation / Time Step	Wave Propagation Quasi-Static	Vibration Static
Filtering	No	Remove High Frequency
Mesh	Uniform	Arbitrary

Time Step Calculation

- Beam and Truss

$$\Delta t_c = \frac{\Delta x}{c} \qquad c = \sqrt{\frac{E}{\rho}}$$

- Shell

$$\Delta t_c = \frac{\Delta x}{c} \qquad c = \sqrt{\frac{E}{\rho(1-\nu^2)}}$$

- Solid

$$\Delta t_c = \frac{\Delta x}{\sqrt{Q + \left(Q^2 + c^2\right)}} \qquad c = \sqrt{\frac{4G}{2\rho_0} + \frac{\partial p}{\partial \rho}}$$

- Spring

$$\Delta t_c = 2\sqrt{\frac{2M_1 M_2}{k\left(M_1 + M_2\right)}}$$

Material Wave Speed

MEDIUM	Meters/Second
Steel	5240
Aluminum	5328
Titanium	5220
Plexiglass	2598
Water	1478
Air	331

Characteristic Length

- Δx for solid element
  - volume/area $_{max\text{-}side}$
- Δx for shell element
  - area/length $_{max\text{-}edge}$ (default)
  - area/length $_{diagonal}$
  - area/length $_{min\text{-}side}$
- Δx for beam element
  - length

Time Step Calculation

- Discrete spring
  - Independent of distance
- Time step scale factor
  - The above is based on linear analysis, for nonlinear analysis, we build in a factor of safety

    0.9 (default)

    0.67 (high velocity)

Execution Time

- Execution time primarily depends on:
  - material properties
  - mesh size
  - number of elements
  - contacts
  - speed of computer
- CPU estimation
  - Time step Δt = minimum $\Delta x/c$
  - number of cycles = termination time / Δt
  - CPU time = (# cycles)(# elements)(time per zone cycle)
  - correction is needed for time step reduction
  - correction is needed for number and size of contacts

Controlling Execution Time

- Avoid bad elements: very small, high stiffness, low density
- Increase mesh size to the limits of accuracy required
- Monitor d3hsp file for 100 smallest time step elements
- Erode elements based on a percentage of initial time step size
- Scale density of shell elements to maintain a minimum time step
- Stop simulation based on a percentage of the initial time step size
- Stop simulation based on a percentage change of energy
- Stop simulation based on a percentage change of mass
- Alternatively the maximum step threshold may be a function of time
- *CONTROL_TIMESTEP and *CONTROL_TERMINATION and *CONTROL_GLSTAT

Time Scaling

- *CONTROL_TIMESTEP
- Cut off for time step size (minimum time step)
  - Element material properties (moduli not masses) are modified to limit time step size
  - Applicable to shells and materials 3, 18, 19, 24
- A maximum time step can be specified
  - Example: discrete elements
- Scale time step to possibly increase accuracy (<1.0)

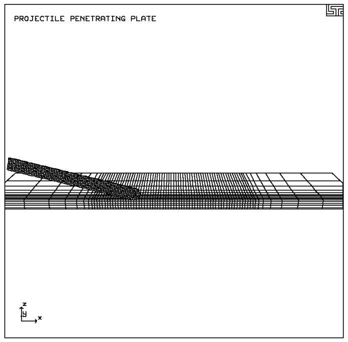
Mass Scaling

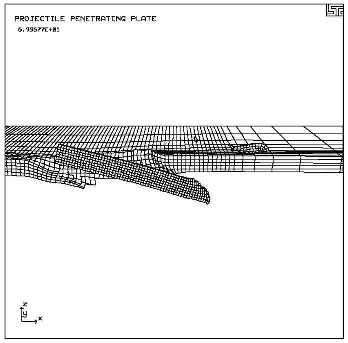
- Quasi-static simulation: metal forming, roof crush
- Compressed elements often have less inertia
- Provide faster solution
- Specify time step size for mass scaling
  - *CONTROL_TIMESTEP
- Time step is controlled by adjusting density
  - $\rho = C^2E$
- Optionally adjust density on first cycle only

Week 5

- Contacts
  - Algorithms
  - Types
  - Guidelines
- Computer Session

Contacts - Sliding interfaces





Contact - General

- Purpose: To Prevent Penetration and/or separation
  - deformable to deformable bodies
  - single surface contact in deformable bodies
  - deformable body to rigid body contact
  - tying deformable structures (strain failure)

CONTACT - Features

- Input options
  - segment set
  - shell set
  - part
  - part set
  - node set
  - within a defined box
  - include all
- Computes solid exterior surfaces
- Static and dynamic coefficients of friction
- Small penetrations
- Damping
- Thickness overrides
- Birth and death time

Contacts

- Methods
  - Penalty
  - Kinematic constraint
  - Distributed parameter
- Types
  - Node to surface
  - Surface to surface
  - Single surface
  - Tied
  - Sliding
  - Rigid body

Penalty Method

- Normal interface springs between penetrating node and contact surface
- Tends to excite very little mesh hourglassing
- Stiffness is prescribed as follows:

  Contact Stiffness= $(\alpha K A^2) / V$

  - α is the penalty scale factor
  - K is the material bulk modulus
  - A is the segment area
  - V is the element volume

Contact

- Discrete Nodes Impacting a Surface
  - NODES_TO_SURFACE (5)
  - AUTOMATIC_NODES_TO_SURFACE (a5)

    No segment orientation
  - ERODING_NODES_TO_SURFACE (16)

    Contact is set free if element fail

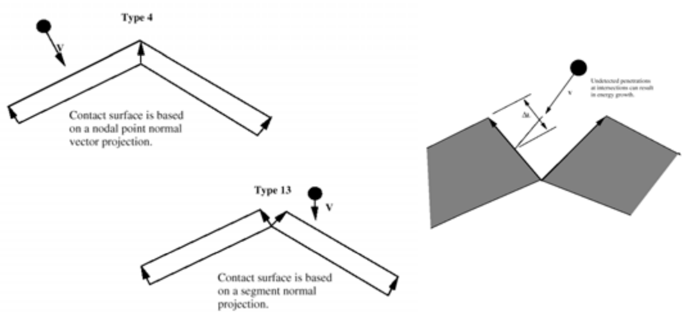
Contact

- Surface to Surface Contact
  - SURFACE_TO_SURFACE (3)
  - AUTOMATIC_SURFACE_TO_SURFACE (a3)
  - ONE_WAY_SURFACE_TO_SURFACE (10)
  - AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE (a10)
  - ERODING_SURFACE_TO_SURFACE (14)
  - SINGLE_EDGE (22)

Contact

- Single Surface Contact:
  - SINGLE_SURFACE (4)
  - AUTOMATIC_SINGLE_SURFACE (13)
  - AIRBAG_SINGLE_SURFACE (a13)
  - ERODING_SINGLE_SURFACE (15)
  - AUTOMATIC_GENERAL (26)
- Type 4 contact requires uniform normal orientation
- Type 13 contact
  - Normal orientation may be random
  - Tied interfaces are allowed

Single Surface Projection



Rigid Body Contact

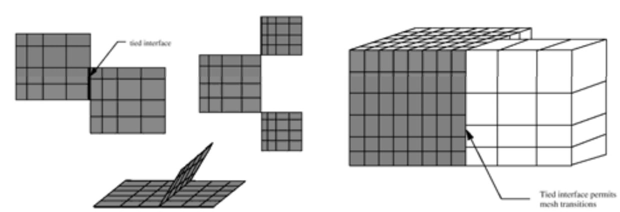
- May be used with deformable bodies
- Arbitrary force-deflection curve
- Keywords
  - RIGID_BODY_TWO_WAY_TO_RIGID_BODY (19)
  - RIGID_NODES_TO_RIGID_BODY (20)
  - RIGID_BODY_ONE_WAY_TO_RIGID_BODY (21)
- Special Case
  - DRAWBEAD (23)

Kinematic Constraint Method

- Based on impact and release condition of Hughes et al, 1976
- Momentum conservation is insured
- Constraints are placed on the nodal displacements of the slave nodes
- Slave surface should be the fine mesh (to prevent kinks)
- Used for tied interfaces

Tied Interfaces

- Constraints are imposed on the slave nodes
  - use coarsely meshed side as master surface
- Good for mesh transitions
- Good for tying parts together
- See also *CONSTRAINED options



Tied Surfaces

- TIED_SURFACE_TO_SURFACE (2)
  - tying surfaces with translational degree-of-freedom (DOF)
- TIED_NODES_TO_SURFACE (6)
  - tying translational DOF of nodes to surface
  - does not transmit moments
- TIED_SHELL_EDGE_TO_SURFACE (7)
  - tying both translational and rotational DOF
- TIEBREAK_NODES_TO_SURFACE (8)
  - normal and shear failure forces
- TIRBREAK_SURFACE_TO_SURFACE (9)
  - normal and shear failure stress

Surface to Surface Constraint Algorithm

- Taylor and Flanagan constraint algorithm (1989)
  - interface nodes remain on or very close to the surface
  - elastic vibrations are insignificant
  - generally not applicable to rigid bodies
  - additional nodal constraints cause problems (e.g., spot welds)
  - can only subject a surface to this constraint from one side
- CONSTRAINT_SURFACE_TO_SURFACE (17)
- CONSTRAINT_NODES_TO_SURFACE (18)

Distributed Parameter Method

- Distribution of mass and pressure over the contact area
- Constraints imposed on slave node acceleration and velocity to insure movement along the master surface
- Used for sliding only
  - fluid structure
  - gas to structure
- SLIDING_ONLY (1)
- SLIDING_ONLY_PENALTY (p1)

Viscous Contact Damping

- Damp oscillations normal to the contact surfaces
- Damping as a percentage of critical ($2m\omega$)

  twenty percent damping = 20, not 0.20

  $m = \min\{m_{slave}, m_{master}\}$
- Natural frequency of interface is computed using the contact stiffness

*CONTROL_CONTACT

Change Defaults for Contact Computation

- Global scale factor for sliding interface penalties (Default=0.10)
- Scale factor for rigid body with fixed rigid wall interaction
- Initial penetration check
- Consider shell thickness for
  - surface to surface and node to surface
- Consider shell thickness changes for single surface
  - Flag: membrane straining produces shell thickness changes
- Penalty stiffness calculation method
- Reorient contact segment normals
- Contact searching frequency (Default = 10)

Other Contact

- *CONTROL_ENERGY
  - compute sliding interface energy dissipation
- *DATABASE_OPTION
- ASCII
  - GLSTAT: global statistics
  - RCFORC: resultant contact forces
  - SLEOUT: contact energy
- Binary
  - INTFOR: contact interface data

*CONTACT_ENTITY

- Treat contact between deformable bodies & geometric rigid body
- Geometric Entities
  - cylinder, plane, sphere, toroid, ellipsoid, VDA
- Improves performance
  - closed from contact calculation
- Improves accuracy
  - surface is independent of finite element mesh refinement

Applications for Contact Entities

- Metal Forming: The punch and die surface geometries can be input as geometric surfaces which are treated as rigid
- Treating contact between rigid body occupant dummy hyper ellipsoids and deformable structures such as airbag and instrument panels
- Coupling with the rigid body occupant modeling codes, such as MADYMO and CAL3D
- Airbag into steering wheel

Contact Guideline I

- Contact type 13 is recommended
  - Perhaps the most efficient and reliable contact
  - 1 large contact zone is not more expensive than several small ones
  - Automatic contact input simplifies problem translation
- Contact type 5 is simple and 100% reliable if the master surface is closed surface, the same goes for type 3 if both master and slave surface are closed
  - advantage: contact forces can be monitored
  - advantage: individual contacts can be controlled

Contact Guidelines II

- Uniform meshes improve results
- Make master side with coarser mesh for one way treatment
- Contacts work "best" when master and slave sides have similar mesh sizes and material properties
- The "soft constraint option" may be appropriate for the case of objects with highly different properties
- Avoid sharp corners

Contact Guidelines III

- Avoid initial penetration at all cost!
  - They may cause stresses that exceed the yield stress and will initiate buckling immediately
- Default values are good reference values
- Contact normals must point to the opposing surface except when noted otherwise
- Undeformable or very stiff parts, whose kinematics are determined by contact forces, must be modeled very fine since distribution over many nodes is important to obtain realistic results

Week 6

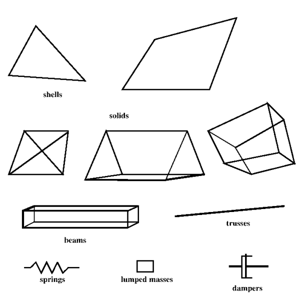
- Element Formulation
  - Solid
  - Shell
  - Beam
  - Discrete
- Hourglass Control
- Computer Session

Element Selection Criteria

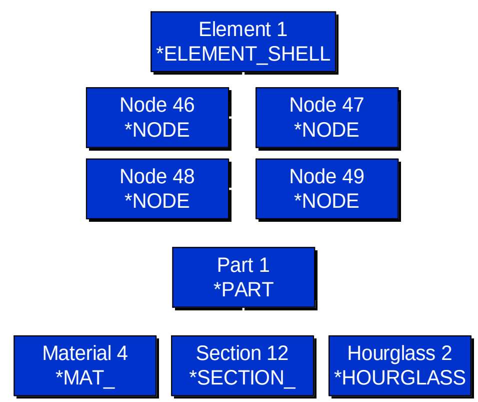
- Structural Geometry
- Loading Conditions
- Model Assumptions
- Economics

*ELEMENT

- Define elements using nodes
- To organize and specify how elements behave, elements are assigned to a part
- Types
  - Concentrated masses
  - Springs
  - Dampers
  - Seat belts
  - Beams
  - Shells
  - Solids
  - Thick Shells



Elements and Parts



*PART

- Relates part ID between elements, sections and materials
- Organize elements into meaningful groups
- *SECTION
  - Specify mathematical (element) formulation
  - Specify integration rule
  - Specify geometric properties not defined explicitly by the element
- *MATERIAL
  - Specify material behavior (and properties)
- Elements that require part ID's
  - beam discrete, seatbelt, shell, solid, tshell
- Elements that do not require part ID's
  - mass, seatbelt, accessories

*SECTION

- Element type dependent
  - element formulation
  - integration rule
  - thickness or cross-section properties
- Used in *PART

*ELEMENT_MASS

- Assign a lumped mass to a node
- Required input
  - mass element ID (EID)
  - node ID
  - mass value

*ELEMENT_DISCRETE

- Define springs and dampers
  - mounts, locks, hinges, simplified components away from design area, lumped parameter modeling
- Massless
- Rotations are in radians
- Required input
  - element ID
  - part ID
  - 2 nodal ID's (one can be ground)
  - orientation (N1 to N2 x-direction only, etc.)
  - scale factor on force

Discrete Elements

- Force behavior is defined using material options
- *MAT_SPRING
  - Elastic stiffness, f (displacement)
    - elastic and nonlinear elastic
    - inelastic
    - general nonlinear
    - Maxwell (exponential decay of stiffness)
- *MAT_DAMPER
  - Damping constant, F(velocity)
    - viscous
    - nonlinear viscous

*SECTION_DISCRETE

- Translation or rotation
- Dynamic magnification factor
- Clearance
- Tension/compression deflection limits

*ELEMENT_BEAM

- Model long slender objects (10:1 ratios)
  - steering columns, suspension components, building frames, rebar
- Required input
  - element ID
  - part ID
  - 3 nodal point ID's

*SECTION_BEAM

- Element Formulation
  - 6 degree-of-freedom
    - Hughes-Liu (default)
    - Belytschko-Schwer resultant
    - Belytschko-Schwer with full integration
    - Belytschko-Schwer tubular beam
  - 3 degree-of-freedom
    - truss
    - cable
- Cross Section
  - rectangular, tubular, I, C, T, Z, arbitrary
  - areas or inertias (2nd moment and polar)

*ELEMENT_SOLID

- Model components that are discretized relatively similar in size in three orthogonal directions
  - castings, forgings, radiators, belts
- Proper mass and inertia representation
- Required input:
  - element ID
  - part ID
  - nodal ID's (4, 6, 8 - tetrahedron, wedge, brick)
- Tetrahedrons can control Hourglassing but timestep becomes more difficult to control

Solid Element Formulation

- Constant stress solid (default)
  - 8-node brick
  - hourglass control with $1 \times 1 \times 1$ integration
  - also valid for wedge and tetrahedron
- Fully integrated S/R solid
  - 8-node brick
  - $2 \times 2 \times 2$ integration (no hourglassing)
  - no locking due to selectively reduced integration

Solid Element Formulation (continued)

- Fully integrated quadric with nodal rotations
  - 8-node brick
  - 14 integration points
  - rotational degrees of freedom
- S/R quadratic tetrahedron with nodal rotations
  - 4-node brick
  - 5 integration points
  - rotational degrees of freedom

*ELEMENT_SHELL

- Model components that are relatively thin in one direction
  - sheet metal, thin-walled structures, engines blades, cams
  - crashworthiness, occupant simulation, sheet metal stamping, impacts on aircraft, impulsive loading or missiles
- Required input:
  - element ID
  - part ID
  - 4 nodes for a quad, 3 nodes for a tri
  - override default thickness at each node

Element Formulation

1. Hughes-Liu
2. Belytschko-Tsay (default)
3. BCIZ triangular shell
4. C_0 triangular shell (recommended)
5. Belytschko-Tsay membrane
6. S/R Hughes-Liu
7. S/R co-rotational Hughes-Liu
8. Belytschko-Leviathan shell
9. Fully integrated Belytschko-Tsay membrane
10. Belytschko-Wong-Chiang
11. Fast co-rotational Hughes-Liu
16. Fast Fully integrated

- Can be set globally or for each part
  - *CONTROL_SHELL, *SECTION_SHELL

Shell Element Parameters

- *SECTION_SHELL
  - number of through shell thickness integration points (default = 2)
  - thickness at each node
  - reference surfaces - top, mid, bottom surface (Hughes-Liu only)
- *CONTROL_SHELL
  - treat degenerated quads as C_0 tri's
  - membrane straining causes thickness change
  - B-W-C warping stiffness for B-T
  - element warpage warning

Shell Features

- Finite strain are treated
- Arbitrary and fixed through thickness integration
- Shell element thickness update
- Geometric properties are optionally specified on the element card for complete generality
- Fully vectorized and parallelized for SGI, Cray, HP
- Constitutive subroutines are shared by all shell elements
- Common local coordinates systems are used
- Hourglass control available to control zero energy modes

Shell Technology

Why only three and four-noded shell?

- High frequency content in higher order shells drives the time step size down
- Contact algorithm are not set up to run with higher order surfaces
- Mesh generation and post-processing would have to be further developed
- Less robust than simpler elements under large distortion

Shell Technology

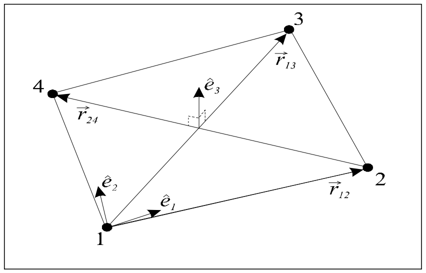
- Why multiple shell formulation?
  - Fully integrated for elasticity, metal forming applications, airbags, or whenever accuracy is concern
  - Triangular elements for mesh grading since collapsed quad are too stiff. Autosorting of tri's in LS-DYNA
  - Belytschko-Tsay for speed!
  - Membran elements without bending or transverse shear for very thin sheets

Belytschko-Tsay Shell

- The B-T shell element was developed by Belytschko and Tsay in 1981, and improved by Belytschko, Lin and Tsay in 1984
- Based on a combined co-rotational and velocity-strain formulation
- Co-rotational portion of the formulation avoids the complexities of nonlinear mechanics by an embedded coordinate system in the element
- The conjugate stress to velocity strain is the Cauchy stress
- Shell kinematics assumes that nodes are co-planar

Co-Rotational Coordinates

- Construction of element coordinate system



Belytschko-Tsay Shell

- This shell was implemented as a computationally efficient alternative to the Hughes-Liu shell
  - With 5 integration points, the B-T shell requires 725 mathematical operations, whereas the under-integrated H-L shell requires 4066
  - Selective reduced (S/R) integration of H-L shell requires 35,367 mathematical operations
- Because of its computational efficiency, the B-T shell element is usually the element formulation of choice. For this reason, it has become the default 4-node shell element formulation

Belytschko-Tsay Shell Lacks of Accuracy

- B-T shell is fast but simplifications required for speed affects accuracy
- Two problems that illustrate its shortcomings
  - hemispherical shell problem with corner forces
  - twisted beam problem with end load
- The B-T shell ignores warpage in geometry
- Determining when and if the simplification are important is nearly impossible unless another shell is available for making comparisons
- The B-T shell will eventually be phased out as new shells gain acceptance

Belytschko-Wong-Chiang Shell

- Advantages
  - improved treatment of transverse shear
  - more accurate for warped element configurations
- Disadvantages
  - more costly than B-T
  - does not degenerate into a triangular shell
- Alternative: B-W-C warping stiffness for B-T (*CONTROL_SHELL)

Triangular Shells

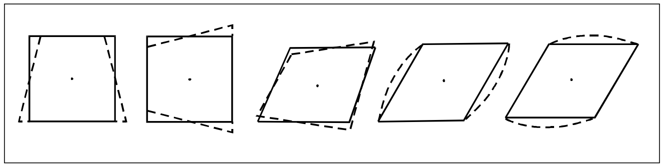
- Tri's are stiffer
- Tri's are more costly
  - reduction in time step
  - increased number of elements
- Tri's do not hourglass (advantage)
- Tri's are used for mesh transition regions
- Tri's are good for eliminating warped quads
- Tri's are good for curved geometry
- Avoid using stiff degenerated quad's (use C_0 tri's)

Shell Technology Cost Comparisons

- Operation counts do not translate directly into increased cost
  - Gather-scatter costs are identical for each formulation (30%)
  - Constitutive models are identical for each formulation (30%)
  - Considerable overhead in contact, rigid bodies, constraints, and other elements, results in speed differences on real problems being typical 15%
  - Main issue is whether the improved results in some applications justify the added extra cost
- Operation Count - example
- Relative Cost - example

Hourglassing
Zero Energy Modes

- Examples
  - shells
  - solids



Department of Civil and Environmental Engineering   The George Washington University

Hourglassing

- Hourglass modes are a results of rank of deficiency in the element stiffness matrix caused by insufficient integration points
  - Zero Energy Modes
- These modes result in mathematical sates that are not physically possible
- Hourglassing can be controlled under certain circumstances
- One point integration is much faster
  - so we accept the risk
    - but always check energy balances to be safe
    - general rule: Hourglass Energy < 10% of Internal Energy

Department of Civil and Environmental Engineering   The George Washington University

Hourglassing Control
Types and Their Limitations

- Viscous forms
  - standard LS-DYNA (default)
  - Flanagan-Belytschko (2)
  - Flanagan-Belytschko with exact volume integration (3)
- Stiffness forms
  - Flanagan-Belytschko (4)
  - Flanagan-Belytschko with exact volume integration (5)
- The stiffness form may result in stiff response
- Flanagan-Belytschko behaves better for large rotations

Department of Civil and Environmental Engineering   The George Washington University

Hourglassing Control

- Stiffness form
  - more stable in many applications
  - preferred for vehicle crash and sheet stamping
- Viscous form sometimes works better
- If hourglassing occurs in an area where it does not influence the design area of concern, then it is admissible
- Fully integrated elements have no hourglassing
- Hourglass modes are orthogonal to the real deformation
- Work done by hourglass control does not appear in energy equation
- Total energy will reduces slightly
- Hourglass energy dissipation appears in GLSTAT and MATSUM

Department of Civil and Environmental Engineering   The George Washington University

Hourglassing - Keywords

- *CONTROL_ENERGY
  - switch to have hourglass energy calculation (10% penalty)
- *CONTROL_HOURGLASS
  - set hourglass type (default is viscous)
  - *can* modify hourglass coefficient
- *HOURGLASS
  - set hourglass type and parameters to use for specific parts
- *PART
  - change global hourglass type and parameters for a specific part by identifying a specific hourglass ID

Hourglass Prevention

- The Flanagan-Belytschko (‘shape vector’) formulation is preferred since the increased cost is small and the default ‘base vector’ formulation interferes with the rigid body modes of an element
- Choice between viscous and stiffness force calculation is not a real issue
  - stiffness formulation may reinforce the structure and low coefficient should be used (0.01 to 0.02)
- Hourglass modes are better avoided by mesh-refinement
- If mesh refinement does not work, switch formulation rather than tweak hourglass parameters

Week 7

- Material Models
  - Metals
  - Rubber
  - Foam
- Rigid Bodies
- Computer Session

Materials
Material Behavior and Properties

- Material behavior and properties are possibly the most difficult portion in developing useful simulation results
- Nonlinear material behavior is constantly being updated through new research
- Nonlinear material properties are not easily obtained
- Components often need to be modeled with simplified geometry

Basic Material Behavior

Behavior	Hardening	Ideal	Softening
Stability	yes	yes	no
Uniqueness	yes	yes	yes
Application	metals, concrete	crude steel, plastics	dense sand, concrete large def.

Softening

- As soon as 1 element reaches yield, all other elements will unload elastically as the yielding element proceeds forward (and downward) on the stress-strain curve
- Simple example - LPM with two springs

Elastoplastic Complexities

- Failure
  - minimum time step
  - plastic strain
  - failure stain
- Unloading/Re-loading
- Strain Rate Effects

Strain Rate Effects

- Strain Rate Effect (dynamic effect) - SRE
  - quick loading of materials can cause changes in material properties. Most notable in steels
  - strain rate = rate at which material deforms
  - laboratory testing is done quasi-statically, actual applications are dynamic
  - SRE is most dominate at low strains (up to 5%) and for mild steels

Strain Rate Effects

- Cowper-Symonds
  - scale the yield stress by stain rate dependent factor
- Johnson-Cook
  - scale the flow stress by an effective plastic strain rate factor
- General
  - scale the yield stress by a strain rate dependent factor
    - curve: scale factor versus strain rate
- Strain rate dependent plasticity
  - curve: yield stress versus effective strain rate

Other Materials - Increased Complexity

- Anisotropic
- Multi-Layers

Material Types

- Composites
- Ceramics
- Fabric
- Foam
- Glass
- Metal
- Plastic
- Rubber
- Soil/Concrete

Common Materials

- Metals
- Rubber
- Foam

Metals

1. *MAT_ELASTIC
3 *MAT_PLASTIC_KINEMATIC
24 *MAT_PIECEWISE_LINEAR_PLASTICITY

1. *MAT_ELASTIC

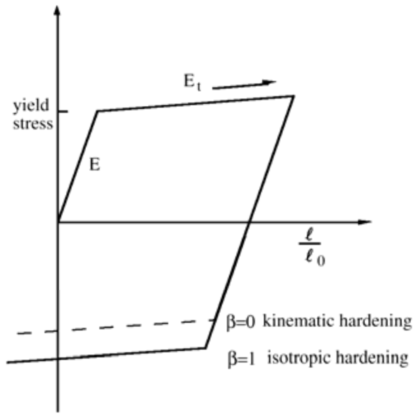
- Define liner material
- Beams, Shells, Solids, Thick Shells
- Input
  - density
  - Young's modulus
  - Poisson's ratio
- Axial an bending damping for Bel-Schwer beam

3. *MAT_PLASTIC_KINEMATIC

- Defines a bilinear constitutive law
- Beams, Shells, Solids, Thick Shells
- Input
  - density yield stress
  - Young's modulus tangent modulus
  - Poisson's ratio hardening
- Cowper-Symonds strain rate effect
- Element deletion based on failure strain

3. *MAT_PLASTIC_KINEMATIC

Hardening



24 *MAT_PIECEWISE_LINEAR_PLASTICITY

- Defines a bilinear constitutive law or an arbitrary stress versus strain curve
- Beams, Shells, Solids, Thick Shells
- Input
  - density — yield stress
  - Young's modulus — tangent modulus
  - Poisson's ratio — stress-strain curve
- Cowper-Symonds SRE or arbitrary strain rate dependency
- Element deletion based on plastic strain or minimum time step

Rubber

- Hyperelastic material
- Response is path independent
- Generally considered to be incompressible since the bulk modulus greatly exceeds the shear modulus in magnitude

7 *MAT_BLATZ-KO_RUBBER
27 *MAT_MOONEY-RIVLIN_RUBBER
31 *MAT_FRAZER-NASH_RUBBER
38 *MAT_BLATZ-KO_FOAM
77 *MAT_HYPERELASTIC_RUBBER
77 *MAT_OGDEN_RUBBER
87 *MAT_CELLULAR_RUBBER

7. *MAT_BLATZ-KO_RUBBER

- Solids and Shells
- Nearly incompressible continuum rubber
- Poisson's ratio is at 0.463
- Input
  - density
  - shear modulus
- Suitable for polyurethane rubber
- Second Piola-Kirchoff stress which is transformed to Cauchy stress

27. *MAT_MOONEY-RIVLIN_RUBBER

- Solid and Shells
- Reduction of the Frazer-Nash rubber model
- Poisson's ratio (> 0.49 recommended)
- Two constants: *A* and *B*
- Strain energy density function
  - $W = A(I-3) + B(II-3) + C(III2-1) + D(III-1)2$
    - *I, II, III* invariants of right Cauchy-Green tensor
    - $C = 0.5A + B$
    - $D = [A(5\nu - 2)+B(11\nu - 5)]/2(1 - 2\nu)$
    - $2(A + B)$ = shear modulus of linear elasticity
- If $A = B = 0.0$, then they are calculated using a least square fir from uniaxial data via a load curve (A and B will be printed in d3hsp file)

31. *MAT_FRAZER-NASH_RUBBER

- Solid only
- Modified from the hyperelastic constitutive law described by Kedington (1988)
- Poisson's ratio: $0.49 < \nu < 0.50$
- Strain energy functional

  $U = C_{100}I_1 + C_{200}I_1^2 + C_{300}I_1^3 + C_{400}I_1^4 + C_{110}I_1I_2 + C_{210}\, I_1^2I_2 + C_{010}I^2 + C_{020}I_2^2 + f(J)$
  - *I, J* stress invariants
- Input either constants or force versus change in gauge length

38. *MAT_BLATZ-KO_FOAM

- Solids and Shells
- Compressible foam
- Poisson's ratio is fixed at 0.25
- Suitable for rubber like foams of polyurethane
- Input
  - density
  - shear modulus

77 *MAT_HYPERELASTIC_RUBBER
77. *MAT_OGDEN_RUBBER

- Solids only
- General hyperelastic rubber combined with linear viscoelasticity (Ogden 1984, Christensen 1980)
- Similar behavior but provide different parameter options
- Poisson's ratio > 0.49 recommended
- Effectively a Maxwell fluid which consists of dampers and springs in series
- Results are nearly identical to Mooney-Rivlin (Mat 27) for large values of Poisson's ratio

Crushable Foams

- Energy dissipative
- Response of these materials are path dependent
- Soils and crushable foams represent materials that stiffen as they compress or compact
  - Soil, for example, has a specific stiffness while the empty space exists between the grains. As the grains bridge in compression, the stiffness of the material increases. Soils tend to unload linearly. Soils do not hold tensile loads
- Porous forms have similar behavior where the stiffness is lower until the internal voids are compressed. Foams tend to unload along the loading path

Crushable Foams

5 *MAT_SOIL_AND_FOAM
14 *MAT_SOIL_AND_FOAM_FAILURE
26 *MAT_HONEYCOMB
53 *MAT_CLOSED_CELL_FOAM
57 *MAT_LOW_DENSITY_FOAM
62 *MAT_VISCOUS_FOAM
63 *MAT_CRUSHABLE_FOAM
75 *MAT_BILKHU/DUBOIS_FOAM

Crushable Foams

- Foam components are usually very soft compared to surrounding structure
- Consequently, contact forces are likely to cause hourglass in foam components, especially if they are modeled coarsely
- Therefore, fully integrated brick elements are best used, eliminating hourglass problems from the start

Crushable Foams

- It is important to check both uni-axial and tri-axial behavior of the model before using it to represent a certain foam
- Some of the more commonly used foams :

  26 *MAT_HONEYCOMB
  - radiators, moving deformable barriers

  57 *MAT_LOW_DENSITY_FOAM
  - seat cushions, padding on side impact dummies

  62 *MAT_VISCOUS_FOAM
  - energy absorbing foam found on certain crash dummies

  75 *MAT_BILKHU/DUBOIS_FOAM
  - good general isotropic crushable foam
  - considers uni-axial and tri-axial test data
  - unloading is elastic (Poisson’s ratio set to zero)

Rigid Bodies

- RB has six-degree-of-freedom
- RB boundary conditions act on CG
- Defining control to the nodes of a RB is risky and not recommended
  - nodal constraints
  - prescribed motion
- D3HSP contains mass calculation for RB
- RBOUT contains ASCII information

Rigid Bodies

Extensive capabilities to model rigid bodies (RB):

- *MAT_RIGID (material 20) can be used to define a RB from defined shell, solid or beam elements
- Two RB can be merged into a single RB
- Extra nodes for RB
- A RB can be defined by a set of nodes
- Joints connect RB
- Material can be switched between rigid and deformable
- Multiple contact treatments available
- Inertial properties and initial conditions can be defined

Rigid Bodies

- *MAT_RIGID (material 20) can be used to define a RB from defined shell, solid or beam elements
  - material properties are used by contact algorithms
  - center of mass can be constrained (displacement and rotation)
  - inertial properties (CG, mass I's, initial velocities) can be redefined through the *PARTS command
- Two RB can be merged into a single RB
- *CONSTRAINED_RIGID_BODIES

Rigid Bodies

- Extra nodes for RB
  - *CONSTRAINED_EXTRA_NODES
- Nodal Rigid Body
  - CONSTRAINED_NODAL_RIGID_BODY
  - a RB can be defined by a set of nodes
  - inertial properties
    - computed automatically from nodal masses and coordinates
    - user specified: CG, mass, I's, initial velocities

Joints Connecting RB

- 6 type of joint definitions
  - spherical, revolute, universal
  - planar, translational
  - cylindrical
- Nodal pairs (1,2), (3,4) and (5,6) should coincide with the exception of cylindrical and translational joints
- Joints only apply to RB
- Keywords
  - *CONSTRAINED_JOINT_option
  - *CONSTRAINED_JOINT_STIFFNESS

Deformable Switching

- Materials can be switched between rigid and deformable
- *DEFORMABLE_TO_RIGID
  - switch deformable materials to rigid at the start
- In a Restart file:
  - *RIGID_DEFORMABLE_R2D
  - *RIGID_DEFORMABLE_D2R

Rigid Bodies - Contact

- Sliding Interfaces
  - may be used with deformable bodies
  - arbitrary force-deflection curves
  - 19 *CONTACT_RIGID_BODY_TWO_WAY_TO_RIGID_BODY
  - 20 *CONTACT_RIGID_NODES_TO_RIGID_BODY
  - 21*CONTACT_RIGID_BODY_ONE_WAY_TO_RIGID_BODY
- Rigidwall
  - specify penalty for treating rigid wall to rigid body contact in *CONTACT_CONTACT
- Geometric Entities - *CONTACT_ENTITIES
  - Improve performance and accuracy

Week 8

- Analysis Tools
- Output Options
- Quasi-Static Analysis
- Dynamic Relaxation
- Damping
- Restart
- Cross-Section Analysis

Analysis Tools

- There are many things that could go wrong within a model
- LS-DYNA3D tends to provide results even in cases where the results are nonphysical
  - Some errors are simple: incorrect format
  - Some errors are subtle: duplicate nodes in a small region
  - Some errors are indirect: slight modification in one option effects another
  - Some errors are complex: shooting nodes in contact region
  - Some errors are frustrating: Floating Point Exception- core dump
- ☑ It helps to have guidelines and strategies for uncovering such errors

Analysis Tools

- Critical Files: Input file and D3HSP file
- Common Errors
- Consistent Units
- Ctrl+C Sense Switches
- Interactive Graphics
- The Post Processor
- Colleagues or LSTC Hotline (925)-449-2500

The Input File

- Everything *wrong* must be in the input file!
- A brief scan through portions of this file can reveal much
- Things to look for include:
  - ********’s being written where numbers should be
  - existence of material properties
  - incorrect material numbers for various element types
  - lack of boundary conditions on the node cards
  - all of the other major input sections
- When experiencing a read error on DYNA3D startup, the first thing to do is to make sure that the section where the error occurred as well as the one before are defined correctly and the appropriate control flags are set

D3HSP File

- The d3hsp file echoes the input file and is also interesting reading
- When something goes wrong, scan through the d3hsp file making sure that all of the various options are exactly as you expected
- d3hsp file also contains other useful information:
  - material and system mass properties
  - latest options that are not yet in the manual
  - 100 smallest time step controlling elements
  - when an element fails
  - most error termination statements
  - CPU usage
- When nothing goes wrong, scan through the d3hsp file anyway

Common Errors

- Most error terminations provide info on the cause of the problem
  - input formatted incorrectly
  - odd inertial properties
  - initial contact penetration
  - load curve definitions
  - massless nodes
- Floating Point Exception can be caused by several things
  - parts with zero density
  - parts with zero thickness
  - over-constrained nodes
  - constrained nodes, contacts and rigid walls all occurring on the same node at one time
  - ill-defined load curves

Common Errors

- Sometimes runs terminate normally but still have problems
  - Units
  - Material properties
  - Loads, boundary conditions, and initial conditions
  - Contact segment normals
  - Problem time and cycle number may explain a lack of interesting output
  - element aspect ratios, angles, and warpage
  - Duplicate nodes and elements
  - Cracks or holes
  - Material numbers

Consistent Units

Mass	Length	Time	Force
kg	m	s	N
kg	mm	ms	k
ton (1000kg)	mm	s	N
slug	ft	s	lbf
lbf-s^2/in	in	s	lbf

The Sense Switches

- Ctrl+C interrupts execution and prompts for a sense switch
- sw1 A restart file is written and LS-DYNA3D terminates
- sw2 LS-DYNA3D responds with time and cycle info
- sw3 A restart file is written and LS-DYNA3D continues
- sw4 A plot state is written and LS-DYNA3D continues
- sw5 Interactive graphics
- sw6 Stop Sequencing Interactive Graphics

The Sense Switch

The items which are printed include:

- Kinetic Energy
- External Work
- Internal Energy
- Total Energy
- X, Y, and Z Momentum
- Controlling Element Number and Type
- Current Time Step and Controlling Element

The Sense Switch

- If the time step is too small, then the mesh may contain a disproportionally small elements. And with a minor modification to the input, it can be eliminated, allowing an order-of-magnitude increase in time step
- A rapidly decreasing time step can be the result of a badly applied load or boundary condition. It can also be the result of mesh pattern that is unfavorable for the deformation or bad material data. Most of the remaining causes are signaled by the energy conservation printout
- For most of impact problems, simulations are started with an initial kinetic energy. Normally, no external work is applied. The kinetic energy will decrease, the internal energy will increase, and the total energy should remain constant. If the total energy takes some big jumps, the model has an error. It is time to recheck everything, but especially contact and fracture. Use SW4, to output a graphic state before the problem crashes.

The Sense Switch 2

- Another piece of good debug information is the momentum printout. For impact problems, this tells you immediately if the projectile is going in the right direction, Dividing by mass of the moving body tells you if the object has the correct velocity. The changes in the momentum vector should be compatible with what is expected of the system

Analysis Tools - Conclusions

- You can never be too thorough
- When all of the above fails, there is the possibility of an LS-DYNA3D code error
- Mathematics can play marvelous tricks on physics

ASCII Output Files

- Obtain specialized output in ASCII format for x-y plot
- Platting can be done in using
  - LS-TAURUS (phase 3)
  - LS-POST (ASCII)
- Desired output must be specified
  - *DATABASE_option
    - Option = desired output type
    - requires time interval between output
- Some options require additional data

ASCII Output Files

- Airbag Statistics *DATABASE_ABSTAT
  - volume
  - pressure
  - internal energy
  - mass flow
  - density
  - temperature
  - output mass flow rate
  - mass
- Boundary Nodal Forces *DATABASE_BNDOUT
  - boundary condition nodal forces and energies when discrete forces are applied at a boundary

ASCII Output Files

- Discrete Element Data *DATABASE_DEFORC
  - forces and moments for discrete elements: springs and dampers
  - global x,y,z
  - resultant
- Element Data *DATABASE_ELOUT
  - requires *DATABASE_HISTORY_option
    - beam or a set of beams
    - shell or a set of shells
    - solid or a set of solids

ASCII Output Files

- Element Data (continued)
  - beams
    - axial resultant force
    - resultant s-shear and t-shear
    - resultant s-moment and t-moment
    - toroidal resultant
  - solids (bricks)
    - global stress
    - global shear
    - effective shear
    - effective stress
    - yield function
  - shells
    - strain
      - global strain
      - global shear strains
      - lower and upper surface strain
    - stress
      - global stress
      - global shear stress
      - plastic strain
      - integration points

ASCII Output Files

- Global Statistic *DATABASE_GLSTAT
  - global energy information
  - total energy
  - initial energy/total energy
  - kinetic energy
  - internal energy
  - hourglass energy
  - stonewall energy
  - spring and damping energy
  - system damping energy
  - sliding interface energy
  - external work
  - global x, y, z velocity
  - time step
  - *CONTROL_ENERGY required to get hourglass, stonewall, sliding

ASCII Output Files

- Geometric Contact Entities *DATABASE_GCEOUT
- Joint Forces *DATABASE_JNTFORC
  - global forces and moments
  - resultant forces and moments
- Material Energies *DATABASE_MATSUM
  - material information for each part
  - kinetic energy
  - internal energy
  - hourglass energy
  - global momentum
  - global velocity

ASCII Output Files

- Nodal Contact Forces *DATABASE_NCFORC
  - global forces
  - component analysis
- Nodal Forces *DATABASE_NODFOR
  - global forces
  - requires *DATABASE_NODAL_FORCE_GROUP
- Nodal Point Data *DATABASE_NODOUT
  - displacements and rotations
  - velocities and angular velocities
  - acceleration and and angular accelerations
  - requires *DATABASE_HISTORY_(nodes or a set of nodes)

ASCII Output Files

- Rigid Body Data *DATABASE_RBDOUT
  - displacement and rotation
  - velocity and angular velocity
  - acceleration and angular acceleration
- Resultant Interface Forces *DATABASE_RCFORC
  - global forces of defined contacts
- Rigid Wall Forces *DATABASE_RWFORC
  - normal forces
  - global forces
- Seat Belt Output *DATABASE_SBTOUT

ASCII Output Files

- Cross Section Forces *DATABASE_SECFORC
  - global forces and moments
  - resultant forces and moments
  - dimensional center
  - requires *DATABASE_CROSS_SECTION
    - specify element to element nodes on the cross section
  - applies to beams, shells, solids springs and dampers
- Sliding Interface Energy *DATABASE_SLEOUT

ASCII Output Files

- SPC Reaction Forces *DATABASE_SPCFORC
  - global forces and momnets
- Spotweld/Rivet Forces *DATABASE_SWFORC
  - axial force
  - shear force
  - applies to all rigid nodal constraints

ASCII Output Files

- Specialized output for various post-processing software
- AVS Database *DATABASE_AVSFLT
- Deformed Geometry *DATABASE_DEFGEO
  - this option also creates a Nastran Bulk Data File (NASBDF)
    - can be read into many pre-processors
- MOVIE *DATABASE_MOVIE
- MPGS *DATABASE_MPGS

* AVS, MOVIE and MPGS require *DATABASE_EXTENT specifications

Quasi-Static Analysis

- Inertial forces are insignificant
- Material properties are independent of time
- Implicit solvers do not converge for large systems of equations
- Convergence is not an issue with explicit solvers
  - Possible time duration is inherently small with explicit solvers
    0.001 seconds to 0.100 seconds
- Several methods for quasi-static solutions

Quasi-Static Analysis

- Dynamic Relaxation
  - prescribed geometry
  - Initial loading
- Damping
- Time Scaling
- Mass Scaling

Dynamic Relaxation Prescribed Geometry

- Nodal x, y, z displacements and rotations
- Initialization
- Linear analysis
- Specify stress initialization file on the excution line
  - m = sif
- File format is I8,6E15
- *CONTACT_DYNAMIC_RELAXATION/*CONTROL_DAMPING
  - dynamic relaxation flag = 2

Dynamic Relaxation Loading

- Specify initial loading (*DEFINE_CURVE)
- Apply load curve to initialization
- Apply load curve to initialization and analysis
- *CONTACT_DYNAMIC_RELAXATION/*CONTROL_DAMPING
  - dynamic relaxation flag = 1

Dynamic Relaxation Options

- Iterations between convergence check (default=250)
- Dynamic relaxiation factor (default = 0.995)
- Optional termination time (default = infinity)
- Scale factor for computed time step
- Convergence tolerance

Dynamic Relaxation Comments

- The computed velocity is multiplied by the dynamic relaxation factor
- No solution exists when kinetic energy is prescribed
  - discounts motion control
  - allow pressure, force, and thermal loads
- Kinetic energy limit
- Papadrakakis automatic control option is based on critical damping

Damping

- System damping or mass damping

  $a_n = M^{-1}(P_n - F_n + H_n - F_n^{damp})$

  equivalent to putting structure in a viscous environment
- Raleigh damping or stiffness damping
  - pulls energy out from within an element
  - analogy: a structure heating up
- Critical damping = 2 * ω_{min}

Damping

- *CONTROL_DAMPING
- *CONTROL_GLOBAL
  - define mass weighted nodal damping that applies globally to the nodes of deformable bodies
- *DAMPING_PART_MASS
  - mass weighted damping to specified parts
  - damps all motions including rigid body motions
  - preferred for low frequencies
- *DAMPING_PART_STIFFNESS
  - Rayleigh stiffness damping
  - orthogonal to rigid body motion
  - preferred for high frequencies

Restart

- Restart a simulation
- Allowable changes:
  - termination time
  - output intervals
  - add nodal boundary conditions
  - delete contacts, parts, elements
  - switch from rigid bodies to deformable
  - switch from deformable bodies to rigid

Restart

- The ability to stop and restart a simulation is extremely useful
- reset output intervals
- delete contact surfaces
- delete elements and parts
- change boundary conditions
- switch between deformable bodies and rigid bodies
- control time step and termination
- change damping options

Restart

- The restart LS-DYNA file is similar to the original input file
- ls-dyna3d r=d3dump I=restart.k
  - ls-dyna3d (ls940) execution command for LS-DYNA
  - d3dump complete state dump of the simulation at t he time you want to restart
  - restart.k keyword format restart deck
- For keyword format, some versions of LS-DYNA require all output to be re-specified (*DATABASE commands)

Cross Section Analysis

- Define cross sections through various parts of a structure
  - transmission forces (interface force or section forces)
  - moments
  - centroid location
  - area

  All above quantities are functions of time
- Overall view of how a structure is performing
- Individual components can be analyzed to see how and when they transmit loads, crush and absorb energy throughout an event
- Verify model accuracy

Defining Cross Sections

- Define cross sections by specifying:
  - the nodes of the cross section
  - the element to be used to calculate the forces at those nodes

    The sign of the forces transmitted are determined by which side of the nodes the elements lie on
- Automatically
  - *DATABASE_CROSS_SECTION_PLANE
- User specified
  - *DATABASE_CROSS_SECTION_SET
- Specify desired output frequency
  - *DATABASE_SECFORC

Force Calculation

- In general, the load is comprised of two components: element stiffness and mass inertia. The equilibrium equation of a node on a cross section can be written as Newton's 2nd Law:

  F + **f** = Mass * Acceleration

  F = nodal force due to stress in the cross section elements

  f = interface forces
- This equation is a vector, resulting in global x, y, and z forces
- Mass is allocated to ensure equal and opposite forces when choosing elements on one side of the nodes versus the other side
- Accelerations for each node and stresses for each element are calculated throughout a simulation, thus computing cross section information is relatively inexpensive