

*CONTROL

The keyword control cards are optional and can be used to change defaults. They can also activate solution options, such as mass scaling, adaptive remeshing, and the implicit solver (see the *CONTROL_IMPLICIT section for details). We do, however, suggest defining the *CONTROL_TERMINATION card. *The order of the control cards in the input file is arbitrary. To avoid ambiguities, define no more than one control card of each type.*

The available control cards follow in alphabetical order:

- *CONTROL_ACCURACY
- *CONTROL_ACOUSTIC
- *CONTROL_ACOUSTIC_COUPLING
- *CONTROL_ACOUSTIC_SPECTRAL
- *CONTROL_ADAPSTEP
- *CONTROL_ADAPTIVE
- *CONTROL_ADAPTIVE_CURVE
- *CONTROL_AIRBAG
- *CONTROL_ALE
- *CONTROL_BULK_VISCOSITY
- *CONTROL_CHECK_SHELL
- *CONTROL_COARSEN
- *CONTROL_CONSTRAINED
- *CONTROL_CONTACT
- *CONTROL_COUPLING
- *CONTROL_CPG
- *CONTROL_CPM
- *CONTROL_CPU
- *CONTROL_DEBUG

*CONTROL

*CONTROL_DISCRETE_ELEMENT
*CONTROL_DYNAMIC_RELAXATION
*CONTROL_EFG
*CONTROL_ENERGY
*CONTROL_EOS_USER_LIBRARY
*CONTROL_EXPLICIT_THERMAL_ALE_COUPLING
*CONTROL_EXPLICIT_THERMAL_BOUNDARY
*CONTROL_EXPLICIT_THERMAL_CONTACT
*CONTROL_EXPLICIT_THERMAL_INITIAL
*CONTROL_EXPLICIT_THERMAL_OUTPUT
*CONTROL_EXPLICIT_THERMAL_PROPERTIES
*CONTROL_EXPLICIT_THERMAL_SOLVER
*CONTROL_EXPLOSIVE_SHADOW
*CONTROL_FORMING_AUTO_NET
*CONTROL_FORMING_AUTOCHECK
*CONTROL_FORMING_AUTOPOSITION_PARAMETER
*CONTROL_FORMING_BESTFIT
*CONTROL_FORMING_HOME_GAP
*CONTROL_FORMING_INITIAL_THICKNESS
*CONTROL_FORMING_MAXID
*CONTROL_FORMING_ONESTEP
*CONTROL_FORMING_OUTPUT
*CONTROL_FORMING_PARAMETER_READ
*CONTROL_FORMING_POSITION
*CONTROL_FORMING_PRE_BENDING
*CONTROL_FORMING_PROJECTION

*CONTROL

*CONTROL_FORMING_REMOVE_ADAPTIVE_CONSTRAINTS
*CONTROL_FORMING_SCRAP_FALL
*CONTROL_FORMING_SHELL_TO_TSHELL
*CONTROL_FORMING_STONING
*CONTROL_FORMING_STRAIN_RATIO_SMOOTH
*CONTROL_FORMING_TEMPLATE
*CONTROL_FORMING_TIPPING
*CONTROL_FORMING_TRAVEL
*CONTROL_FORMING_TRIM_MERGE
*CONTROL_FORMING_TRIM_SOLID_REFINEMENT
*CONTROL_FORMING_TRIMMING
*CONTROL_FORMING_UNFLANGING
*CONTROL_FORMING_USER
*CONTROL_FREQUENCY_DOMAIN
*CONTROL_HOURGLASS
*CONTROL_IMPLICIT_AUTO
*CONTROL_IMPLICIT_BUCKLE
*CONTROL_IMPLICIT_CONSISTENT_MASS
*CONTROL_IMPLICIT_DYNAMICS
*CONTROL_IMPLICIT_EIGENVALUE
*CONTROL_IMPLICIT_FORMING
*CONTROL_IMPLICIT_GENERAL
*CONTROL_IMPLICIT_INTERA_RELIEF
*CONTROL_IMPLICIT_JOINTS
*CONTROL_IMPLICIT_MODAL_DYNAMIC
*CONTROL_IMPLICIT_MODAL_DYNAMIC_DAMPING

*CONTROL

*CONTROL_IMPLICIT_MODAL_DYNAMIC_MODE
*CONTROL_IMPLICIT_MODES
*CONTROL_IMPLICIT_ORDERING
*CONTROL_IMPLICIT_RESIDUAL_VECTOR
*CONTROL_IMPLICIT_ROTATIONAL_DYNAMICS
*CONTROL_IMPLICIT_SOLUTION
*CONTROL_IMPLICIT_SOLVER
*CONTROL_IMPLICIT_SSD_DIRECT
*CONTROL_IMPLICIT_STABILIZATION
*CONTROL_IMPLICIT_STATIC_CONDENSATION
*CONTROL_IMPLICIT_TERMINATION
*CONTROL_LSDA
*CONTROL_MAPPING_SHIFT_BY_DT
*CONTROL_MAT
*CONTROL_MPP_CONTACT_GROUPABLE
*CONTROL_MPP_DECOMPOSITION_ADJUST_MTYPE_COST
*CONTROL_MPP_DECOMPOSITION_ADJUST_PART_COST
*CONTROL_MPP_DECOMPOSITION_ARRANGE_PARTS
*CONTROL_MPP_DECOMPOSITION_AUTOMATIC
*CONTROL_MPP_DECOMPOSITION_BAGREF
*CONTROL_MPP_DECOMPOSITION_CHECK_SPEED
*CONTROL_MPP_DECOMPOSITION_CONTACT_DISTRIBUTE
*CONTROL_MPP_DECOMPOSITION_CONTACT_ISOLATE
*CONTACT_MPP_DECOMPOSITION_DEFORDED_GEOMETRY
*CONTROL_MPP_DECOMPOSITION_DISABLE_UNREF_CURVES
*CONTROL_MPP_DECOMPOSITION_DISTRIBUTE_ALE_ELEMENTS

*CONTROL

*CONTROL_MPP_DECOMPOSITION_DISTRIBUTE_SPH_ELEMENTS
*CONTROL_MPP_DECOMPOSITION_ELCOST
*CONTROL_MPP_DECOMPOSITION_FILE
*CONTROL_MPP_DECOMPOSITION_FLAG_STRESS_STRAIN_CURVE
*CONTROL_MPP_DECOMPOSITION_METHOD
*CONTROL_MPP_DECOMPOSITION_NODISTRIBUTE_DES_ELEMENTS
*CONTROL_MPP_DECOMPOSITION_NUMPROC
*CONTROL_MPP_DECOMPOSITION_OUTDECOMP
*CONTROL_MPP_DECOMPOSITION_PARTS_DISTRIBUTE
*CONTROL_MPP_DECOMPOSITION_PARTSET_DISTRIBUTE
*CONTROL_MPP_DECOMPOSITION_RCBLOG
*CONTROL_MPP_DECOMPOSITION_REDECOMPOSITION
*CONTROL_MPP_DECOMPOSITION_SCALE_CONTACT_COST
*CONTROL_MPP_DECOMPOSITION_SCALE_FACTOR_SPH
*CONTROL_MPP_DECOMPOSITION_SHOW
*CONTROL_MPP_DECOMPOSITION_TRANSFORMATION
*CONTROL_MPP_IO_LSTC_REDUCE
*CONTROL_MPP_IO_NOBEAMOUT
*CONTROL_MPP_IO_NOD3DUMP
*CONTROL_MPP_IO_NODUMP
*CONTROL_MPP_IO_NOFULL
CONTROL_MPP_IO_NOTIEDIO
*CONTROL_MPP_IO_SWAPBYTES
*CONTROL_MPP_MATERIAL_MODEL_DRIVER
*CONTROL_MPP_PFILE
*CONTROL_MPP_REBALANCE

***CONTROL**

*CONTROL_NONLOCAL
*CONTROL_OUTPUT
*CONTROL_PARALLEL
*CONTROL_PORE_AIR
*CONTROL_PORE_FLUID
*CONTROL_PZELECTRIC
*CONTROL_REFERENCE_CONFIGURATION
*CONTROL_REFINE_ALE
*CONTROL_REFINE_ALE2D
*CONTROL_REFINE_MPP_DISTRIBUTION
*CONTROL_REFINE_SHELL
*CONTROL_REFINE_SOLID
*CONTROL_REMESHING
*CONTROL_REQUIRE_REVISION
*CONTROL_RIGID
*CONTROL_SEGMENTS_IN_ALE_COUPLING
*CONTROL_SHELL
*CONTROL_SOLID
*CONTROL SOLUTION
*CONTROL_SPH
*CONTROL_SPH_INCOMPRESSIBLE
*CONTROL_SPOTWELD_BEAM
*CONTROL_STAGED_CONSTRUCTION
*CONTROL_START
*CONTROL_STEADY_STATE_ROLLING
*CONTROL_STRUCTURED

***CONTROL**

*CONTROL_SUBCYCLE
*CONTROL_TERMINATION
*CONTROL_THERMAL_EIGENVALUE
*CONTROL_THERMAL_FORMING
*CONTROL_THERMAL_NONLINEAR
*CONTROL_THERMAL_SOLVER
*CONTROL_THERMAL_TIMESTEP
*CONTROL_TIMESTEP
*CONTROL_UNITS
*CONTROL_2D_REMESHING_REGION

*CONTROL

*CONTROL_ACCURACY

*CONTROL_ACCURACY

Purpose: Define control parameters that can improve the accuracy of the calculation.

Card 1	1	2	3	4	5	6	7	8
Variable	OSU	INN	PIDOSU	IACC	EXACC	SRTFLG		
Type	I	I	I	I	F	I		
Default	0	↓	optional	0	0.0	0		

VARIABLE	DESCRIPTION
OSU	Global flag for 2nd order objective stress updates (see Remark 1 below). Generally, for explicit calculations only those parts undergoing large rotations, such as rolling tires, need this option. Objective stress updates can be activated for a subset of part IDs by defining the part set in columns 21-30. EQ.0: Off (default) EQ.1: On
INN	Invariant node numbering for shell and solid elements. (See Remarks 2 and 3). EQ.-4: On for both shell and solid elements except triangular shells EQ.-2: On for shell elements except triangular shells EQ.1: Off (default for explicit) EQ.2: On for shell and thick shell elements (default for implicit) EQ.3: On for solid elements EQ.4: On for shell, thick shell, and solid elements
PIDOSU	Part set ID for objective stress updates. If this part set ID is given only those part IDs listed will use the objective stress update; therefore, OSU is ignored.
IACC	Implicit accuracy flag, turns on some specific accuracy considerations in implicit analysis at an extra CPU cost. See Remark 4 . EQ.-1: Off

VARIABLE	DESCRIPTION
	EQ.0: On (only for implicit) EQ.1: On (only for implicit) EQ.2: On (partially also for explicit, for compatibility when switching between implicit and explicit)
EXACC	Explicit accuracy parameter: EQ.0.0: Off (default) GT.0.0: On (see Remark 5)
SRTFLG	Flag to process parts, contacts, nodal rigid bodies, and elements in a sequence sorted by their respective user IDs, regardless of the order they appear in the input files. When turned on, it ensures consistent results if the user modifies the order of these entities in the input files, such as by changing the order of include files between runs of the same model. See Remark 6 . EQ.0: Off (default) EQ.1: On

Remarks:

1. **Second-order objective stress update.** Second-order objective stress updates are occasionally necessary. Some examples include spinning bodies such as turbine blades in a jet engine, high velocity impacts generating large strains in a few time steps, and large time step sizes due to mass scaling in metal forming. There is a significantly added cost, which is due in part to the added cost of the second-order terms in the stress update when the Jaumann rate is used and the need to compute the strain-displacement matrix at the mid-point geometry. This option is available for the following element types:
 - a) one-point brick elements
 - b) the selective-reduced integrated brick element which uses eight integration points
 - c) fully integrated plane strain and axisymmetric volume-weighted (type 15) 2D solid elements
 - d) the thick shell elements
 - e) Belytschko-Tsay shell elements

- f) Belytschko-Tsay shell elements with warping stiffness
 - g) Belytschko-Chiang-Wong shell elements
 - h) S/R Hughes-Liu shell elements
 - i) type 16 fully integrated shell element
2. **Invariant node numbering for shell elements.** Invariant node numbering for shell and thick shell elements affects the choice of the local element shell coordinate system. The orientation of the default local coordinate system is based on the shell normal vector and the direction of the 1-2 side of the element. If the element numbering is permuted, the results will change in irregularly shaped elements. With invariant node numbering, permuting the nodes shifts the local system by an exact multiple of 90 degrees. In spite of its higher costs [<5%], the invariant local system is recommended for several reasons. First, element forces are nearly independent of node sequencing; secondly, the hourglass modes will not substantially affect the material directions; and, finally, stable calculations over long time periods are achievable. The INN parameter has no effect on thick shell form 2, which is always invariant, and thick shell form 3, which is never invariant.
3. **Invariant node numbering for solid elements.** Invariant node numbering for solid elements is available for anisotropic materials only. This option has no effect on solid elements of isotropic material. This option is recommended when solid elements of anisotropic material undergo significant deformation.
4. **Implicit calculations.** All other things being equal, a single time step of an implicit analysis usually involves a larger time increment and deformation than an explicit analysis. Many of the algorithms in LS-DYNA have been heavily optimized for explicit analysis in ways that are inappropriate for implicit analysis. While an implicit analysis, by default, invokes many measures to ensure accuracy, certain corrections associated with unusual applications or with large computational expense are invoked only by setting IACC = 1. A list of features that are included with this option follows at the end of this remarks section. The explicit column indicates features that are also active when IACC = 2. IACC = 2 is primarily intended for switching between implicit and explicit and maintaining compatibility between features. As of release R16, the corrections are on by default for implicit calculations. To turn off the corrections, set IACC = -1.
5. **EXACC.** The EXACC option is developed to improve the numerical accuracy for an explicit analysis. Currently, nodal coordinates are computed and stored in double precision in all versions. In most cases, this is sufficiently accurate and EXACC is recommended to be off. However, in some cases, particularly when initial coordinates are large and displacements are small, EXACC can increase the accuracy of computations. To use this option, the EXACC parameter should

be set to a positive value which is a characteristic element length for the mesh. For example, if the typical edge length in a model is 5.0 mm and the length units are millimeters, then set EXACC = 5.0.

To translate a model or a component of a model, keyword *INCLUDE_TRANSFORM is recommended because the 10-digit or 20-digit formats in keyword *NODE are not enough to represent a double precision value without round off error. Using *INCLUDE_TRANSFORM together with EXACC can yield a solution that is more consistent with the original result before the transformation.

Element features activated for IACC = 1	Implicit	Explicit
Strong objectivity, meaning that large rotations do not induce spurious strains and stresses, enforced for <ul style="list-style-type: none"> ◦ 1D seatbelts ◦ 2nd order shells (types 23 and 24) ◦ 1st order shells (types -16, 4 and 16) ◦ 1st order solids (types -2, -1, 1, 2, 13, 15, and 16) ◦ cohesive solids (types 19 and 20) ◦ beam elements (types 1, 2, and 9) 	Yes	No
Assumed strain formulation of 2 nd order triangular shell (type 24)	Yes	No
Thickness is involved in solution process for shell thickness update (see ISTUPD on *CONTROL_SHELL) and not imposed as a post-processing step. This is for shell type 16 and allows contact to act on updated geometry, thus avoiding large contact stresses at beginning of steps.	Yes	No
Allow beam types 1, 2, and 11 in implicit linear analyses	Yes	No
Linear element formulations in linear implicit, meaning that results are scaled linearly with respect to scaling of boundary conditions, of <ul style="list-style-type: none"> ◦ 1st order shells (types 13, 15, 16) ◦ 1st order solids (types -2, -1, 1, 2). 	Yes	No
Switch low-order quadrilateral shell elements (types 1, 2, 6, 7, 8, 10, 11) to type 16, solid element type 1 to type 2 for material 83, and beam types 4 and 5 to type 1. ESORT on *CONTROL_SOLID and *CONTROL_SHELL are still honored, and a nonzero ISOLID/ISHELL/IBEAM on *CONTROL_IMPLICIT_EIGENVALUE has precedence and thus overrides the automatic switch mentioned here.	Yes	No
Activate solid element 13 for all materials	Yes	Yes
Improved representation of torsional and rotational modes in eigenvalue analysis for beams (types 1, 2, and 13). See ITORM = 2	Yes	No

*CONTROL

*CONTROL_ACCURACY

Element features activated for IACC = 1 on *SECTION_BEAM.	Implicit	Explicit
Contact features activated for IACC = 1 Strong objectivity in tied contacts listed on *CONTACT, meaning that large rotations will not induce contact stresses. These contacts also include bending and torsional constraints whenever those are physically justified.	Implicit	Explicit
Material features activated for IACC = 1 Large strain/temperature accuracy in some materials (types 4 shells/solids, 60 solids, 106 solids). Stiffness smoothing of tension/compression transition in material 83, for enhanced implicit convergence characteristics. Accurate Jacobi iterations in some hyperelastic material models (type 30 solids, 77 solids, 83 solids), for better strain assessment and implicit convergence characteristics. Fully iterative plasticity in some metallic material models (type 3 solids, 24 shells/solids, 123 shells/solids), for enhanced accuracy and implicit convergence. Consistent tangent modulus in material 24, accounting for relatively large plastic strains. Switch FORM to 14 for fabric material, *MAT_FABRIC, when not set to either 14 or -14. Switch material 57 to an equivalent material 83 whenever applicable, due to better implicit treatment.	Yes	Yes
Miscellaneous feature activated for IACC=1 Steady state thermal solution allowed in coupled simulations, for compatibility with implicit mechanical statics; see ATYPE on *CONTROL_THERMAL_SOLVER. Consistent nodal forces from body loads for high order elements, accounting for isoparametric shape functions; see *LOAD_BODY. Allow point constraints on rigid body nodes, without transferring it to rigid body center of mass. Use with care, so as to not overconstrain the system; see *BOUNDARY_SPC.	Yes	No

Miscellaneous feature activated for IACC=1	Implicit	Explicit
Strongly objective joint stiffness formulation; see *CONSTRAINED_JOINT_STIFFNESS.	Yes	No
Sparse matrix treatment of control volume airbags for efficiency; see *AIRBAG.	Yes	No

6. **Enabling/disabling SRTFLG with the command line.** The command line option `srtflg=` activates and deactivates SRTFLG by setting `srtflg=1` and `srtflg=0`, respectively. This command line option overrides the value specified in ***CONTROL_ACCURACY**

*CONTROL

*CONTROL_ACOUSTIC

*CONTROL_ACOUSTIC

Purpose: Define control parameters for transient acoustic solutions.

Card 1	1	2	3	4	5	6	7	8
Variable	MACDVP							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
MACDVP	Calculate the nodal displacements and velocities of *MAT_ACOUSTIC volume elements for inclusion in d3plot and time-history files. EQ.0: Acoustic nodal motions will not be calculated EQ.1: Acoustic nodal motions will be calculated

Remarks:

- Solution Costs.** Acoustic simulations using *MAT_ACOUSTIC volume elements (ELFORM = 8 and ELFORM = 14) solve for the displacement potential. The infinitesimal motions of the acoustic nodes can then be found from the gradient of the displacement and velocity potentials. This is purely a post-processing endeavor and has no effect on the predicted pressures and structural response. Calculating these motions will, however, roughly double the cost of the acoustic solution, so it is not done by default.
- Model Validity.** The acoustic theory underpinning *MAT_ACOUSTIC volume elements presumes infinitesimal motions. In the presence of larger motions the pressure calculations will proceed regardless, but the calculation of acoustic nodal motions can then be unreliable.

CONTROL_ACOUSTIC_COUPLING**CONTROL*****CONTROL_ACOUSTIC_COUPLING**

Purpose: Alter default parameters for keywords *BOUNDARY_ACOUSTIC_COUPLING_MISMATCH and *BOUNDARY_ACOUSTIC_COUPLING_SPECTRAL. Changing these parameters is not generally recommended.

Card 1	1	2	3	4	5	6	7	8
Variable	MACCPL	ACECF1	ACECF2	ACECF3	ACECF4			
Type	I	F	F	I	F			
Default	3	1.5	0.79	0.5	0.95			

VARIABLE**DESCRIPTION**

MACCPL	Coupling method: EQ.3: Projection with areal equilibration to enhance enforcement of the zero moments. The equilibration test examines the moments generated by the coupling matrix when the acoustic pressure is constant. If these moments are not zero (rare), then an attempt is made to adjust the coupling coefficients, so the moments are minimized. EQ.4: Projection with no areal equilibration
ACECF1	Multiplier on proximity test
ACECF2	Angle between normal vectors in an orientation test
ACECF3	Multiplier on ceiling test
ACECF4	Area equilibration threshold. The equilibration test is skipped when the accumulated area of the coupling matrix is less than the area of the structural face. This normally occurs with partial coverage.

*CONTROL

*CONTROL_ACOUSTIC_SPECTRAL

*CONTROL_ACOUSTIC_SPECTRAL

Purpose: Request an acoustic spectral element analysis instead of the default isoparametric, acoustic finite element analysis. This keyword is only available for double precision. See Appendix W for a list of keywords this feature supports.

Card 1	1	2	3	4	5	6	7	8
Variable	MASEORD	MASEHRF	MASEKFL	MASEIGX	MASEPLT			
Type	I	I	I	I	I			
Default	none	0	0	1	0			

VARIABLE	DESCRIPTION
MASEORD	Spectral element integration order ($2 \leq \text{MASEORD} \leq 15$). See Remark 2 .
MASEHRF	Optional <i>h</i> -refinement (see Remark 3): EQ.0: No splitting unless tetrahedra or pentahedra are present EQ.1: Split all elements once into hexahedra EQ.2: Split each element a second time into 8 hexahedra EQ.3: Split each element a second time into 27 hexahedra
MASEKFL	Dump flag for <i>h</i> -refined and spectral element meshes: EQ.1: Dump keyword deck of acoustic mesh after <i>h</i> -refinement EQ.10: Dump keyword deck of spectral acoustic element mesh (generated by LS-DYNA) EQ.11: Dump both meshes for review
MASEIGX	Approach to element time step calculation (see Remark 4): EQ.1: Gershgorin theorem EQ.2: Maximum element eigenvalue
MASEPLT	Flag to output a high-resolution plot state format: EQ.0: Not output EQ.4: Ensight binary

Remarks:

1. **Elements Types.** This keyword applies to all elements with ELFORM = 8 in the model. Those elements may be hexahedra, tetrahedra, or pentahedra. No acoustic pyramids may be used in spectral element solutions.
2. **Integration Order.** Spectral elements have extra degrees of freedom. Second-order integration results in 27 degrees of freedom per element. 15th order integration results in 4096 degrees of freedom per element. Only the degrees of freedom at the corner nodes are visible to you for d3plot visualization. One element per wavelength with 8th order integration typically gives a very accurate solution over hundreds of cycles of time.
3. ***h*-refinement.** If pentahedra or tetrahedra are used anywhere in the acoustic fluid mesh, then all acoustic fluid elements are split once into hexahedra. This feature supports additional splitting. For example, you may use it to accommodate the extreme mesh refinement that is often required in ultrasonic wave propagation problems. With this field you do not have to generate and manipulate an extremely large and unwieldy mesh.
4. **Critical Time Step.** The Gerschgorin theorem is a faster estimation method and will yield a more conservative time step. Typically, the conservative time step is also less dispersive and more accurate.

*CONTROL

*CONTROL_ADAPSTEP

*CONTROL_ADAPSTEP

Purpose: Define control parameters for contact interface force update during each adaptive cycle.

Card 1	1	2	3	4	5	6	7	8
Variable	FACTIN	DFACTR						
Type	F	F						
Default	1.0	0.01						

VARIABLE

DESCRIPTION

- FACTIN Initial relaxation factor for contact force during each adaptive remesh. To turn this option off set FACTIN = 1.0. Unless stability problems occur in the contact, FACTIN = 1.0 is recommended since this option can create some numerical noise in the resultant tooling forces. A typical value for this parameter is 0.10.
- DFACTR Incremental increase of FACTIN during each time step after the adaptive step. FACTIN is not allowed to exceed unity. A typical value might be 0.01.

Remarks:

1. **Contact Types.** This command applies to contact with thickness offsets including contact types:

*CONTACT_FORMING_..._

*CONTACT_NODES_TO_SURFACE_

*CONTACT_SURFACE_TO_SURFACE

*CONTACT_ONE WAY_SURFACE_TO_SURFACE.

***CONTROL_ADAPTIVE**

Purpose: Activate adaptive meshing for applications, such as sheet metal forming or bulk metal forming. The field ADPOPT in [*PART](#) identifies the part(s) whose mesh is to be adapted and, to some extent, the type of adaptivity applied. The field ADPTYP in this keyword, ***CONTROL_ADAPTIVE**, also helps determine the type of mesh adaptivity to perform.

Available adaptivity types:

- Shell splitting or shell h -adaptivity
- Solid r -adaptivity in which a new mesh of tetrahedrons is created
- r -adaptivity of continuum shells in which a new mesh of quads is created
- Refinement of a composite sandwich, comprised of a solid core merged to shell face plates, by setting the fields IFSAND to 1 and ADPTYP to 1 or 2.
- 3D axisymmetric (or orbital) adaptivity of axisymmetric geometries comprised of hexahedral and/or pentahedral solids.
- Shell mesh fusion (which has long been available in SMP but only became available in MPP starting with R11.0).

Related keywords include:

1. [*CONTROL_ADAPTIVE_CURVE](#): refines mesh along a curve. This feature tends to generate too many elements along the curve and therefore, negatively affects the computational speed.
2. [*DEFINE_CURVE_TRIM](#): when used together with [*CONTROL_ADAPTIVE_CURVE](#), pre-refines mesh in the area within a specific distance (TCTOL) from both sides of a curve. With this feature you have more control over how many elements will be generated along the curve.
3. [*DEFINE_BOX_ADAPTIVE](#): uses one box for fission and another box for fusion. Boxes can translate or remain stationary. Applies to both shell h -adaptivity and tet r -adaptivity.
4. [*DEFINE_BOX_NODES_ADAPTIVE](#): defines a moving tube along a tool path for mesh fission in front of the tool and fusion behind the tool. This is useful in cases where the tool path is curved, such as in roller hemming simulations.
5. [*DEFINE_CURVE_BOX_ADAPTIVITY](#): defines a polygon adaptive box within which you can control mesh refinement level. This feature is useful when deformation is concentrated in a localized region (such as in a line die simulation).

*CONTROL

*CONTROL_ADAPTIVE

Remeshing hyperelastic materials or material models based on a total Lagrangian formulation may lead to numerical instabilities or inaccurate results.

For alternative forms of adaptivity based solely on mesh refinement, see *CONTROL_REFINE_OPTION.

Card Summary:

Card 1a. Include this card if ADPTYP = 1, 2 or 4 (*h*-adaptivity for shells).

ADPFREQ	ADPTOL	ADPTYP	MAXLVL	TBIRTH	TDEATH	LCADP	IOFLAG
---------	--------	--------	--------	--------	--------	-------	--------

Card 1b. Include this card if ADPTYP = 7 (3D *r*-adaptive remeshing of solid elements).

ADPFREQ		ADPTYP		TBIRTH	TDEATH	LCADP	ADPD3P
---------	--	--------	--	--------	--------	-------	--------

Card 1c. Include this card if |ADPTYP| = 8 (2D *r*-adaptive remeshing for plane stress, plane strain, and axisymmetric continuum elements).

ADPFREQ	ADPTOL	ADPTYP	MAXLVL	TBIRTH	TDEATH	LCADP	
---------	--------	--------	--------	--------	--------	-------	--

Card 2a. This card is optional. LS-DYNA reads it if ADPTYP = 1, 2, or 4.

ADPSIZE	ADPASS	IREFLG	ADPENE	ADPTH	MEMORY	ORIENT	MAXEL
---------	--------	--------	--------	-------	--------	--------	-------

Card 2b. This card is optional. LS-DYNA reads it if ADPTYP = 7.

ADPSCL			ADPENE	ADPDAM	MEMORY		
--------	--	--	--------	--------	--------	--	--

Card 2b.1. Include this card if ADPDAM > 1 on Card 2b.

DAM1							
------	--	--	--	--	--	--	--

Card 2c. This card is optional. LS-DYNA reads it if |ADPTYP| = 8

ADPSIZE	ADPASS				MEMORY		MAXEL
---------	--------	--	--	--	--------	--	-------

Card 3a. This card is optional. LS-DYNA reads it if ADPTYP = 1, 2, or 4.

IADPN90	IADPGH	NCFREQ	IADPCL	ADPCTL	CBIRTH	CDEATH	LCLVL
---------	--------	--------	--------	--------	--------	--------	-------

Card 3b. This card is optional and blank. LS-DYNA reads it if ADPTYP = 7 (for ADPDAM < 2), -8, or 8. Do not include this card if ADPDAM > 1 on Card 2b.

--	--	--	--	--	--	--	--

Card 4a. This card is optional. LS-DYNA reads it if ADPTYP = 1 or 2.

					D3TRACE		IFSAND
--	--	--	--	--	---------	--	--------

Card 4b. This card is optional. LS-DYNA reads it if ADPTYP = 4.

				ADPERR	D3TRACE		
--	--	--	--	--------	---------	--	--

Card 4c. This card is optional. LS-DYNA reads it if ADPTYP = 7.

					D3TRACE	IADPCF	
--	--	--	--	--	---------	--------	--

Card 4d. This card is optional. LS-DYNA reads it if |ADPTYP| = 8.

CNLA			MMM2D		D3TRACE		
------	--	--	-------	--	---------	--	--

Card 5. This card is optional.

INMEMORY							
----------	--	--	--	--	--	--	--

Data Card Definitions:

This card is included if ADPTYP = 1, 2, or 4.

Card 1a	1	2	3	4	5	6	7	8
Variable	ADPFREQ	ADPTOL	ADPTYP	MAXLVL	TBIRTH	TDEATH	LCADP	IOFLAG
Type	F	F	I	I	F	F	I	I
Default	none	10^{20}	1	3	0.0	10^{20}	0	0

VARIABLE	DESCRIPTION
ADPFREQ	Time interval between adaptive refinements; see Figures 12-1 and 12-2 .
ADPTOL	Adaptive error tolerance; ADPTOL is in degrees for ADPTYP set to 1 or 2
ADPTYP	Adaptive options. ADPTYP = 1, 2 and 4 refer to <i>h</i> -adaptivity for shells. ADPTYP = 1 and 2 refer to <i>h</i> -adaptivity for shell / solid / shell sandwich composites.

EQ.1: Angle change in degrees per adaptive refinement relative

VARIABLE	DESCRIPTION
	to the surrounding shells for each shell to be refined
	EQ.2: Total angle change in degrees relative to the surrounding shells for each shell to be refined. For example, if APDTOL = 5 degrees, the shell will be refined to the second level when the total angle change reaches 5 degrees. When the angle change is 10 degrees, the shell will be refined to the third level.
	EQ.4: Adapts when the shell error in the energy norm, Δe , exceeds ADPTOL/100 times the mean energy norm within the part, which is estimated as:
	$\Delta e = \left(\int_{\Omega_e} \frac{\ \Delta \sigma\ ^2}{E} d\Omega \right)^{1/2} .$
	Here E is the Young's modulus. The error of the stresses, $\Delta \sigma$, is defined as the difference between the recovered solution σ^* and the numerical solution, σ^h , that is, $\Delta \sigma \equiv \sigma^* - \sigma^h$. Various recovery techniques for σ^* and error estimators for Δe are defined by ADPERR. This option works for shell types 2, 4, 16, 18, and 20.
MAXLVL	Maximum number of refinement levels. Values of 1, 2, 3, 4, ... allow a maximum of 1, 4, 16, 64, ... shells, respectively, to be created for each original shell. The refinement level can be overridden by *DEFINE_BOX_ADAPTIVE or *DEFINE_SET_ADAPTIVE .
TBIRTH	Birth time at which the adaptive remeshing begins; see Figures 12-1 and 12-2 .
TDEATH	Death time at which the adaptive remeshing ends; see Figures 12-1 and 12-2 .
LCADP	Load curve ID, defining how the adaptive interval is changed as a function of time. If this option is nonzero, the ADPFREQ will be replaced by LCADP. The x -axis is time, and the y -axis is the varied adaptive time interval.
IOFLAG	Flag to generate adaptive mesh at exit, including *NODE , *ELEMENT_SHELL_THICKNESS , *BOUNDARY_OPTION , and *CONSTRAINED_ADAPTIVITY , which is to be saved in the file, adapt.msh:
	EQ.1: Generate h -adapted mesh.

Include this card if ADPTYP = 7.

Card 1b	1	2	3	4	5	6	7	8
Variable	ADPFREQ		ADPTYP		TBIRTH	TDEATH	LCADP	ADPD3P
Type	F		I		F	F	I	I
Default	none		1		0.0	10^{20}	0	0

VARIABLE	DESCRIPTION
ADPFREQ	Time interval between adaptive refinements; see Figures 12-1 and 12-2 .
ADPTYP	Adaptive options: EQ.7: 3D r -adaptive remeshing for solid elements. The adaptive remeshing process uses tetrahedrons (solid formulation 10 or 13, or if EFG, formulation 42), or in the case of 3D axisymmetry (orbital) adaptivity, hexahedral and pentahedral elements. The process generates a completely new mesh initialized from the old mesh using a least squares approximation. The algorithm bases the mesh size on the minimum and maximum edge lengths defined on the input for *CONTROL_REMESHING . This option remains under development, and we are improving its reliability on complex geometries.
TBIRTH	Birth time at which the adaptive remeshing begins; see Figures 12-1 and 12-2 .
TDEATH	Death time at which the adaptive remeshing ends; see Figures 12-1 and 12-2 .
LCADP	Load curve ID, defining how the adaptive time interval is changed as a function of time. If this option is nonzero, LCADP replaces ADPFREQ. The x -axis is time, and the y -axis is the varied adaptive time interval.
ADPD3P	Flag to control the output of an adaptive mesh and d3plot in 3D r -adaptivity: EQ.0: Do not output an adaptive mesh output at exit and do

*CONTROL

*CONTROL_ADAPTIVE

VARIABLE	DESCRIPTION							
	not write d3plot files before and after each adaptive step.							
EQ.1:	Output the adaptive mesh at exit, but do not write d3plot files before and after each adaptive step.							
EQ.2:	Output d3plot files after each adaptive step, but do not output an adaptive mesh at exit.							
EQ.3:	Output d3plot files before and after each adaptive step, but do not output an adaptive mesh at exit. Use this option for metal flow lines.							

Include this card if |ADPTYP| = 8.

Card 1c	1	2	3	4	5	6	7	8
Variable	ADPFREQ	ADPTOL	ADPTYP	MAXLVL	TBIRTH	TDEATH	LCADP	
Type	F	F	I	I	F	F	I	
Default	none	10^{20}	1	3	0.0	10^{20}	0	

VARIABLE	DESCRIPTION
ADPFREQ	Time interval between adaptive refinements; see Figures 12-1 and 12-2 .
ADPTOL	Characteristic element size
ADPTYP	Adaptive options: EQ.±8: 2D r -adaptive remeshing for plane stress, plane strain, and axisymmetric continuum elements, that is, shell formulations 12 through 15. A completely new mesh is generated which is initialized from the old mesh using a least squares approximation. The mesh size is currently based on the value, ADPTOL, which gives the characteristic element size. This option is based on earlier work by Dick and Harris [1992]. If ADPTYP is negative, then self-contacting material will not be merged together. The self-merging is often preferred since it eliminates sharp folds in the boundary; however, if the sharp fold is being simulated, unexpected results are generated.

VARIABLE	DESCRIPTION
MAXLVL	Maximum number of refinement levels. Values of 1, 2, 3, 4, ... allow a maximum of 1, 4, 16, 64, ... shells, respectively, to be created for each original shell. The refinement level can be overridden by *DEFINE_BOX_ADAPTIVE or *DEFINE_SET_ADAPTIVE .
TBIRTH	Birth time at which the adaptive remeshing begins; see Figures 12-1 and 12-2 .
TDEATH	Death time at which the adaptive remeshing ends; see Figures 12-1 and 12-2 .
LCADP	Load curve ID, defining how the adaptive interval is changed as a function of time. If this option is nonzero, LCADP replaces ADPFREQ. The <i>x</i> -axis is time, and the <i>y</i> -axis is the varied adaptive time interval.

This card is optional. LS-DYNA reads it if ADPTYP = 1, 2, or 4.

Card 2a	1	2	3	4	5	6	7	8
Variable	ADPSIZE	ADPASS	IREFLG	ADPENE	ADPTH	MEMORY	ORIENT	MAXEL
Type	F	I	I	F	F	I	I	I
Default	inactive	0	0	0.0	inactive	inactive	0	inactive

VARIABLE	DESCRIPTION
ADPSIZE	Minimum shell size to be adapted based on element edge length. If undefined, the edge length limit is ignored. LT.0: Absolute value defines the minimum characteristic element length to be adapted based on square root of the element area, that is, instead of comparing the shortest element edge with ADPSIZE, it compares the square root of the element area with ADPSIZE , whenever ADPSIZE is defined by a negative value.
ADPASS	One or two pass flag for <i>h</i> -adaptivity: EQ.0: Two pass adaptivity as shown in Figure 12-1 EQ.1: One pass adaptivity as shown in Figure 12-2

VARIABLE	DESCRIPTION
IREFLG	If positive, the mesh is refined uniformly by IREFLG levels at time = TBIRTH. A value of 1, 2, 3, ... creates 4, 16, 64, ... shells, respectively, for each original shell. MAXLVL must be greater than or equal to IREFLG for this to work.
	If negative, $ IREFLG $ is taken as a curve ID. The curve specifies the minimum element size as a function of time. If the ordinate values (minimum element size) are positive, those values will override other element size criteria. If the ordinate values are negative, the absolute value of the ordinate is the element size used for refinement.
ADPENE	For shells, <i>h</i> -adapt the mesh when the FORMING contact surfaces approach or penetrate the tooling surface depending on whether the value of ADPENE is positive (<i>approach</i>) or negative (<i>penetrates</i>), respectively. The tooling adaptive refinement is based on the curvature of the tooling. If ADPENE is positive the refinement generally occurs before contact takes place; consequently, it is possible that the parameter ADPASS can be set to 1 to invoke the one pass adaptivity.
ADPTH	Thickness below which adaptive remeshing begins: EQ.0.0: This parameter is ignored. GT.0.0: Absolute shell thickness level below which adaptive remeshing should begin. LT.0.0: $ ADPTH $ is the element thickness reduction ratio. If the ratio of the element thickness to the original element thickness is less than $1.0 + ADPTHK$, the element will be refined. This option works only if ADPTOL is nonzero. If thickness based adaptive remeshing is desired without angle changes, then set ADPTOL to a large angle for ADPTYP = 1 or 2.
MEMORY	This flag can have two meanings depending on whether the memory environmental variable is or is not set. The command "setenv LSTC_MEMORY auto" (or for bourne shell "export LSTC_MEMORY=auto") sets the memory environmental variable which causes LS-DYNA to expand memory automatically. Note that automatic memory expansion is not always 100% reliable depending on the machine and operating system level; consequently, it is not yet the default. To see if this is set on a particular machine type the command "env". If the environmental variable is not set

VARIABLE	DESCRIPTION
	then when memory usage reaches this percentage, MEMORY, further adaptivity is prevented to avoid exceeding the memory specified at execution time. Caution is necessary since memory usage is checked after each adaptive step, and, if the memory usage increases by more than the residual percentage, 100-PERCENT, the calculation will terminate.
	If the memory environmental variable <i>is set</i> then when the number of words of memory allocated reaches or exceeds this value, MEMORY, further adaptivity is stopped.
ORIENT	This option applies to the FORMING contact option only. If this flag is set to one (1), the user orientation for the contact interface is used. If this flag is set to zero (0), LS-DYNA sets the global orientation of the contact surface the first time a potential contact is observed after the birth time. If tracked nodes are found on both sides of the contact surface, the orientation is set based on the principle of “majority rules.” Experience has shown that this principle is not always reliable.
MAXEL	If this number of shells is exceeded, adaptivity is stopped.

This card is optional. It is read if ADPTYP = 7.

Card 2b	1	2	3	4	5	6	7	8
Variable	ADPSCL			ADPENE	ADPDAM	MEMORY		
Type	F			F	I	I		
Default	0.0			0.0	0	inactive		

VARIABLE	DESCRIPTION
ADPSCL	Strain rate scale factor. See Remark 12 . EQ.0.0: No strain rate scaling GT.0.0: Strain rate scale factor
ADPENE	For three-dimensional <i>r</i> -adaptive solid remeshing (ADPOPT = 2 in *PART), the mesh refinement is based on the curvature of the tooling when ADPENE is positive. See Remark 9 .

*CONTROL

*CONTROL_ADAPTIVE

VARIABLE	DESCRIPTION
ADPDAM	Type of damage accumulation in the workpiece. See Remark 11. EQ.0: No damage accumulation EQ.1: Ratio of effective plastic strain to failure plastic strain EQ.2: Cockcroft-Latham damage
MEMORY	This flag can have two meanings depending on whether the memory environmental variable is or is not set. The command "setenv LSTC_MEMORY auto" (or for bourne shell "export LSTC_MEMORY=auto") sets the memory environmental variable which causes LS-DYNA to expand memory automatically. Note that automatic memory expansion is not always 100% reliable depending on the machine and operating system level; consequently, it is not yet the default. To see if this is set on a particular machine type the command "env". If the environmental variable <i>is not set</i> then when memory usage reaches this percentage, MEMORY, further adaptivity is prevented to avoid exceeding the memory specified at execution time. Caution is necessary since memory usage is checked after each adaptive step, and, if the memory usage increases by more than the residual percentage, 100-PERCENT, the calculation will terminate.

If the memory environmental variable *is set* then when the number of words of memory allocated reaches or exceeds this value, MEMORY, further adaptivity is stopped.

Include this card ADPTYP = 7 if ADPDAM > 1.

Card 2b.1	1	2	3	4	5	6	7	8
Variable	DAM1							
Type	F							
Default	none							

VARIABLE	DESCRIPTION
DAM1	Critical Cockcroft-Latham damage value (include if ADPDAM = 2)

This card is optional. It is read if |ADPTYP| = 8.

Card 2c	1	2	3	4	5	6	7	8
Variable	ADPSIZE	ADPASS				MEMORY		MAXEL
Type	F	I				I		I
Default	inactive	0				inactive		inactive

VARIABLE	DESCRIPTION
ADPSIZE	<p>Minimum shell size to be adapted based on element edge length. If undefined, the edge length limit is ignored.</p> <p>LT.0: Absolute value defines the minimum characteristic element length to be adapted based on square root of the element area, that is, instead of comparing the shortest element edge with ADPSIZE, it compares the square root of the element area with ADPSIZE , whenever ADPSIZE is defined by a negative value.</p>
ADPASS	<p>One or two pass flag for <i>h</i>-adaptivity:</p> <p>EQ.0: Two pass adaptivity as shown in Figure 12-1</p> <p>EQ.1: One pass adaptivity as shown in Figure 12-2</p>
MEMORY	<p>This flag can have two meanings depending on whether the memory environmental variable is or is not set. The command "setenv LSTC_MEMORY auto" (or for bourne shell "export LSTC_MEMORY=auto") sets the memory environmental variable which causes LS-DYNA to expand memory automatically. Note that automatic memory expansion is not always 100% reliable depending on the machine and operating system level; consequently, it is not yet the default. To see if this is set on a particular machine type the command "env". If the environmental variable <i>is not set</i> then when memory usage reaches this percentage, MEMORY, further adaptivity is prevented to avoid exceeding the memory specified at execution time. Caution is necessary since memory usage is checked after each adaptive step, and, if the memory usage increases by more than the residual percentage, 100-PERCENT, the calculation will terminate.</p>

If the memory environmental variable *is set* then when the number

*CONTROL

*CONTROL_ADAPTIVE

VARIABLE	DESCRIPTION
	of words of memory allocated reaches or exceeds this value, MEMORY, further adaptivity is stopped.
MAXEL	If this number of shells is exceeded, adaptivity is stopped.

This card is optional. It is read if ADPTYP = 1, 2, or 4.

Card 3a	1	2	3	4	5	6	7	8
Variable	IADPN90	IADPGH	NCFREQ	IADPCL	ADPCTL	CBIRTH	CDEATH	LCLVL
Type	I	I	I	I	F	F	F	I
Default	0	0	none	1	none	0.0	10^{20}	0

VARIABLE	DESCRIPTION
IADPN90	Fission control flag around radii: GT.0: Maximum number of shells after fission covering the entire radius from starting tangent to ending tangent EQ.-1: This setting works with look-forward adaptivity, making more consistent mesh adaptivity along the radius from starting tangent to ending tangent. The actual number of elements covering the radius, will be controlled by ADP-SIZE and MAXLVL. Note this setting also works to prevent the "kinks" that are likely to happen along the draw wall in the deep drawing scenario, under which the parameter ADPFREQ needs to be set fine enough for fission as the blank draws into the die radius. Also see Remark 5 .
IADPGH	Fission flag for neighbor splitting: EQ.0: Split all neighbor shells EQ.1: Do not split neighbor shells
NCFREQ	Frequency of fission to fusion steps. For example, if NCFREQ = 4, then fusion will occur on the fourth, eighth, twelfth, etc., fission steps, respectively. If this option, is used NCFREQ > 1 is recommended.

VARIABLE	DESCRIPTION
IADPCL	Fusion will not occur until the fission level reaches IADPCL. Therefore, if IADPCL = 2 and MAXLVL = 5, any shell can be split into 256 shells. If the surface flattens out, the number of elements will be reduced if the fusion option is active, i.e., the 256 elements can be fused and reduced to 16.
ADPCTL	Adaptivity error tolerance in degrees for activating fusion. It follows the same rules as ADPTYP as defined in Card 2a.
CBIRTH	Birth time for adaptive fusion. If ADPENE > 0, look-ahead adaptivity is active. In this case, fission, based on local tool curvature, will occur while the blank is still relatively flat. The time value given for CBIRTH should be set to a time later in the simulation after the forming process is well underway.
CDEATH	Death time for adaptive fusion
LCLVL	Load curve ID of a curve that defines the maximum refinement level as a function of time

This card is optional and blank. It is read if ADPTYP = 7 (for ADPDAM < 2), 8, or -8. Include this card if including Cards 4c or 4d. Do not include this card if ADPDAM > 1 for ADPTYP = 7.

Card 3b	1	2	3	4	5	6	7	8
Variable								
Type								

This card is optional. It is read if ADPTYP = 1 or 2.

Card 4a	1	2	3	4	5	6	7	8
Variable					D3TRACE		IFSAND	
Type						I		I
Default						0		0

*CONTROL

*CONTROL_ADAPTIVE

VARIABLE	DESCRIPTION
D3TRACE	Output flag: EQ.0: No additional output states EQ.1: A d3plot state will be output just before and after an adaptive step even though it may not be requested. You may want this output so that the LS-PrePost particle trace algorithm will work in the case of adaptivity.
IFSAND	Set this flag to "1" for forming of sandwich composites. For details, see Remark 6 .

This card is optional. It is read if ADPTYP = 4.

Card 4b	1	2	3	4	5	6	7	8
Variable					ADPERR	D3TRACE		
Type					I	I		
Default					0	0		

VARIABLE	DESCRIPTION
ADPERR	3-digit number, as "XYY", where "X" and "YY" define the options for the recovery techniques and the error estimators, respectively: For X: EQ.0: Superconvergent patch recovery (SPR) (default) EQ.1: The least square fit of the stress to the nodes (Global L2) EQ.2: Error density SPR, as $\Delta\tilde{e} = \Delta e / \text{Area}_{\text{element}}$ EQ.3: Self-weighted SPR, as $\Delta\tilde{e} = \sqrt{\Delta e \times e}$ For YY: EQ.00: Energy norm (default) EQ.01: Cauchy σ_x EQ.02: σ_y EQ.03: σ_z EQ.04: τ_{xy}

VARIABLE	DESCRIPTION
	EQ.05: τ_{yz}
	EQ.06: τ_{zx}
	EQ.07: Effective plastic strain, ε_{ep}
	EQ.08: Pressure
	EQ.09: von Mises
	EQ.10: Principal deviator stress S_{11}
	EQ.11: S_{22}
	EQ.12: S_{33}
	EQ.13: Tresca
	EQ.14: Principal stress σ_{11}
	EQ.15: σ_{22}
	EQ.16: σ_{33}
	EQ.20: User subroutine uadpval to extract the numerical solutions for recovery and uadpnorm to provide an error estimator.
D3TRACE	<p>Output flag:</p> <p>EQ.0: No additional output states</p> <p>EQ.1: A d3plot state will be output just before and after an adaptive step even though it may not be requested. You may want this output so that the LS-PrePost particle trace algorithm will work in the case of adaptivity.</p>

This card is optional. It is read if ADPTYP = 7.

Card 4c	1	2	3	4	5	6	7	8
Variable						D3TRACE	IADPCF	
Type						I	I	
Default						0	0	

*CONTROL

*CONTROL_ADAPTIVE

VARIABLE	DESCRIPTION
D3TRACE	Output flag: EQ.0: No additional output states EQ.1: A d3plot state will be output just before and after an adaptive step even though it may not be requested. You may want this output so that the LS-PrePost particle trace algorithm will work in the case of adaptivity.
IADPCF	Flag to enable adaptive user control files: EQ.0: No user control files EQ.1: Perform run-time control on 3D adaptivity through control files

This card is optional. It is read if |ADPTYP| = 8.

Card 4d	1	2	3	4	5	6	7	8
Variable	CNLA			MMM2D		D3TRACE		
Type	F			I		I		
Default	110.0			0		0		

VARIABLE	DESCRIPTION
CNLA	Limit angle for corner nodes. See Remark 10 . GT.0.0: CNLA is the limit angle. Simplified boundary lines for straight sections are used as remeshing basis. LT.0.0: CNLA is the limit angle and accurate boundary lines are used as remeshing basis (recommended, with CNLA = -110 being a good choice).
MMM2D	If non-zero, common boundaries of all adapted parts will be merged. This is true even if the parts did not share a boundary at the beginning of the calculation but come into contact later.
D3TRACE	Output flag: EQ.0: No additional output states EQ.1: A d3plot state will be output just before and after an

VARIABLE	DESCRIPTION							
	adaptive step even though it may not be requested. You may want this output so that the LS-PrePost particle trace algorithm will work in the case of adaptivity.							

This card is optional.

Card 5	1	2	3	4	5	6	7	8
Variable	INMEMORY							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
INMEMORY	Flag to determine the way shell <i>h</i> -adaptivity is performed (see Remark 8): EQ.0: Traditional out-of-core adaptivity (default). EQ.1: In-core adaptivity (under development). This approach is only supported in MPP and only for ADPTYP = 2. It does not apply to composite sandwich <i>h</i> -adaptivity.

Remarks:

Remarks about *h*-adaptivity

1. **Restarting.** The d3dump and runrsf files contain all information necessary to restart an adaptive run. This did not work in version 936 of LS-DYNA.
2. **Related field in *PART.** For this control card to work, the field ADPOPT = 1 must be set in the [*PART](#) definition. Otherwise, adaptivity will not function.
3. **Contact types and options.** For adaptivity to work optimally, the parameter SNLOG = 1, must be set on [Optional Card B](#) in the [*CONTACT](#) Section. On disjoint tooling meshes the contact option *CONTACT_FORMING_... is recommended.

4. **Root ID (RID) file.** After the adaptive run is completed, a file named adapt.rid is left on disk. This file contains the root ID of all elements created during the calculation, and it does not need to be kept if it is not used in post-processing.
5. **Note about IADPN90 field.** For all metal forming simulations, set IADPN90 to -1.
6. **Mesh adaptivity for sandwiched parts.** Mesh adaptivity can be applied to sandwich composites consisting of layer(s) of solid elements (core) sandwiched by one layer of shell elements each on the top and bottom surfaces of the core. Nodes must be shared at the solid-to-shell interfaces. Prior to R11.0, this mesh adaptivity is limited to only one layer of solid elements with in-plane mesh refinements for both solids and shells. Starting with R11.0 [Zhu et al, 2017], it applies to multiple layers of solid elements; see [Figure 12-3](#).

After adaptive refinement of the shell faces, the solid elements of the core are created by sweeping the shells through the thickness ([Figure 12-4](#)). So, if quad shells are used and refined into smaller quads, matching hexahedral solids are created for the core. Similarly, if triangular shells are used and refined into smaller triangular shells, matching pentahedral solids are created for the core. The number of layers of the solids in the core does not change due to adaptivity. By allowing multiple layers of solids in the thickness direction, different materials may be used through the core thickness. Moreover, this adaptive approach serves to provide better resolution in local areas of interest while keeping the computational cost to a reasonable level.

The adapted sandwich composite may be trimmed by setting ITYP = 1 in [*CONTROL_FORMING_TRIMMING](#) and with [*DEFINE_CURVE_TRIM](#). Trimming of sandwiched parts allows for multiple layers of solids.

In a typical forming setup, the following cards need to be changed to activate the sandwiched part mesh adaptivity:

```
*CONTROL_ADAPTIVE
$# adpfreq    adptol    adptyp    maxlvl    tbirth     tdeath    lcadp    ioflag
  0.00223      4.0       2          4        0.0001.0000E+20      0         0
$# adpsize    adpass    ireflg    adpene    adpth      memory    orient    maxel
  0.90000      1          1        10.00000      0.000        0         0
$# ladpn90    ladpgh    ncfred    ladpcl    adpctl    cbirth     cdeath    lclvl
   -1          0          0          1        0.000      0.0001.0000E+20      0
$                                         IFSAND
                                         1

*PART
Mid-core layer of solid elements
$      PID      SECID      MID      EOSID      HGID      GRAV      ADPOPT      TMID
      1          1          1
Top layer of shell elements
      100        100        1
Bottom layer of shell elements
      101        100        1
```

Note: IFSAND in *CONTROL_ADAPTIVE is set to “1” to activate the composite sandwich adaptivity; each ADPOPT under ***PART** is set to “1” to activate the adaptivity.

7. **Mesh fusion in MPP.** Starting with R11.0, mesh fusion in MPP is available. A mesh fusion example is shown in [Figure 12-5](#), and a partial keyword example is provided below:

```
*CONTROL_ADAPTIVE
$ ADPFREQ    ADPTOL    ADPTYP    MAXLVL    TBIRTH    TDEATH    LCADP    IOFLAG
  0.0024      4.0       2          3          0.0       70.0      0          1
$ ADPSIZE    ADPASS    IREFLG    ADPENE    ADPTH     MEMORY    ORIENT    MAXEL
  0.9         1          0          5.0       0.0
$ IADPN90   IADPGH    NCFREQ    IADPCL    ADPCTL   CBIRTH    CDEATH    LCLVL
  -1          0          2          0          8.0      0.00      70.0
```

In this keyword, NCFREQ defines the fusion frequency, ADPCTL defines the fusion criterion, and CBIRTH and CDEATH define when the fusion starts and ends, respectively.

Based on an extensive study [Fan et al, 2017], the fusion feature reduces the computation time notably (average around 25%) and has little effects on formability analysis, such as thinning and effective strain predictions. The mesh fusion effect on springback prediction is found to be smaller than 10%. Therefore, you can apply fusion extensively in all formability related simulations since the leading formability indicators are little affected while the calculation can be sped-up by a factor of 25%. In springback simulations, however, you should use fusion with caution. Using this feature depends on the required simulation accuracy. You can apply this feature if the springback results are to be used for a quick and rough estimation; however, if the results are to be used for compensating dies and for re-machining, then fusion may not be appropriate

8. **Adaptivity algorithm.** Out-of-core adaptivity requires dumping the mesh out to disk and restarting the program with a newly created input deck at each adaptive step. This method is computationally expensive due to the I/O, keyword processing, and MPP decomposition time. In-core adaptivity defined with IN-MEMRY = 1 does not require exiting the solution loop and can maximize the performance of the CPUs employed. This method is under development, so many features are not yet supported. It is currently supported for basic metal forming. It also maintains the loading condition from ***LOAD_SHELL_SET** upon adaptivity.

Remarks about *r*-adaptivity

9. **Contact and ADPENE.** In three dimensions when ADPENE > 0, the solid part to be adapted is assumed to be on the SURFA side of a contact while the “tooling”, consisting of a shell surface, is assumed to be on the SURFB side of that same contact. ADPENE > 0 represents a distance from the tooling surface within

which the adapted mesh refinement of the SURFA part is influenced by the radius of curvature of the tooling surface. This feature is currently *unavailable* in SMP and for SOFT = 2 in [*CONTACT](#).

10. **CNLA.** In two dimensions r -adaptive remeshing ($|ADPTYP| = 8$), the generated mesh should have a node at each corner so that the corners are not smooth. By default, the mesher will assume a corner wherever the interior angle between adjacent edges is smaller than 110 degrees. Setting CNLA larger than 110 enables angles larger than 110 to be corners. Care should be taken to avoid an unnecessarily large value of CNLA as this may prevent the mesher from generating smooth meshes.
11. **Damage accumulation.** LS-DYNA calculates the damage value in the workpiece undergoing 3D r -adaptivity ($ADPTYP = 7$) with the criterion specified with ADPDAM. The damage is uncoupled from the material response and indicates critical failure spots in the workpiece. Input the material properties needed to determine damage in the data cards for the material model (such as stress triaxiality dependent failure strain) and set additional constants with Card 2b.1 of this keyword. A history variable stores the damage value. The location of the history variable depends on the material type. The d3hsp file provides the location. For example, it is history variable #28 for *MAT_224 and history variable #22 *MAT_106. NEIPH in [*DATABASE_EXTENT_BINARY](#) controls the number of history variables output to d3plot. To output this damage value, set to NEIPH to a value greater than the history variable # for the material under consideration. Currently, this feature is available for *MAT_024, *MAT_106, *MAT_107, and *MAT_224.
12. **Strain rate scaling.** To speed up the forging simulation, the tool velocity is usually increased. For strain-rate-dependent materials, the strain rate needs to be scaled down to obtain the correct stress response in the workpiece. Currently, this feature is available for *MAT_024, *MAT_106, *MAT_107, and *MAT_224.

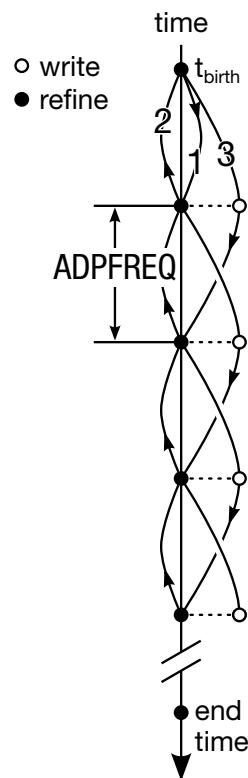
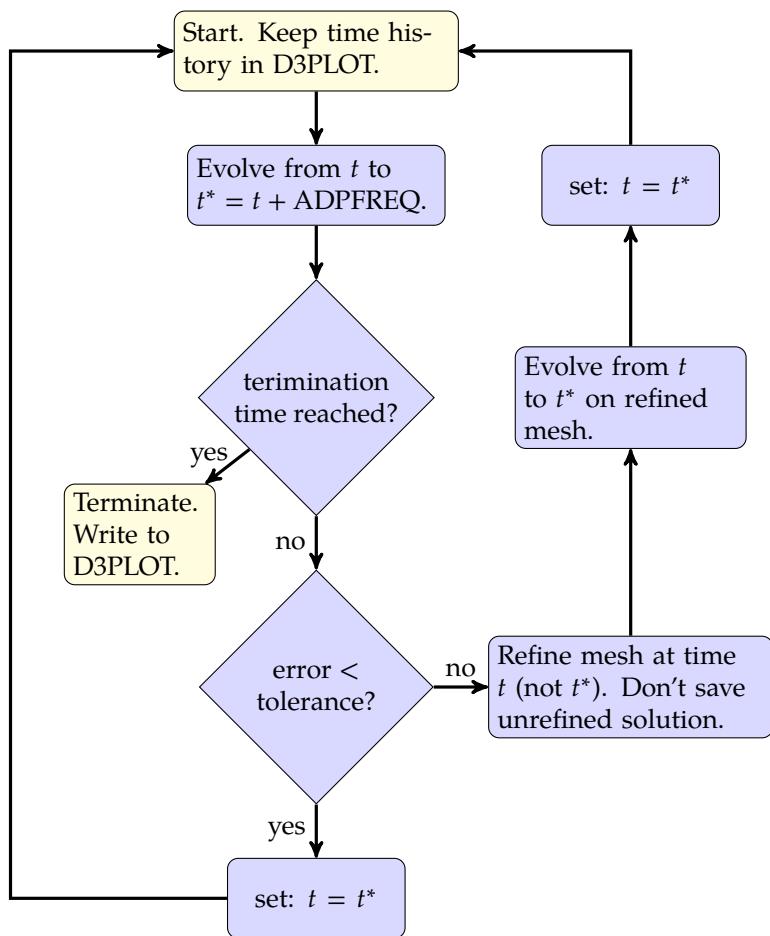


Figure 12-1. Flowchart for ADPASS = 0. While this option is *sometimes* more accurate, ADPASS = 1 is *much* less expensive and recommended when used *with* ADPENE.

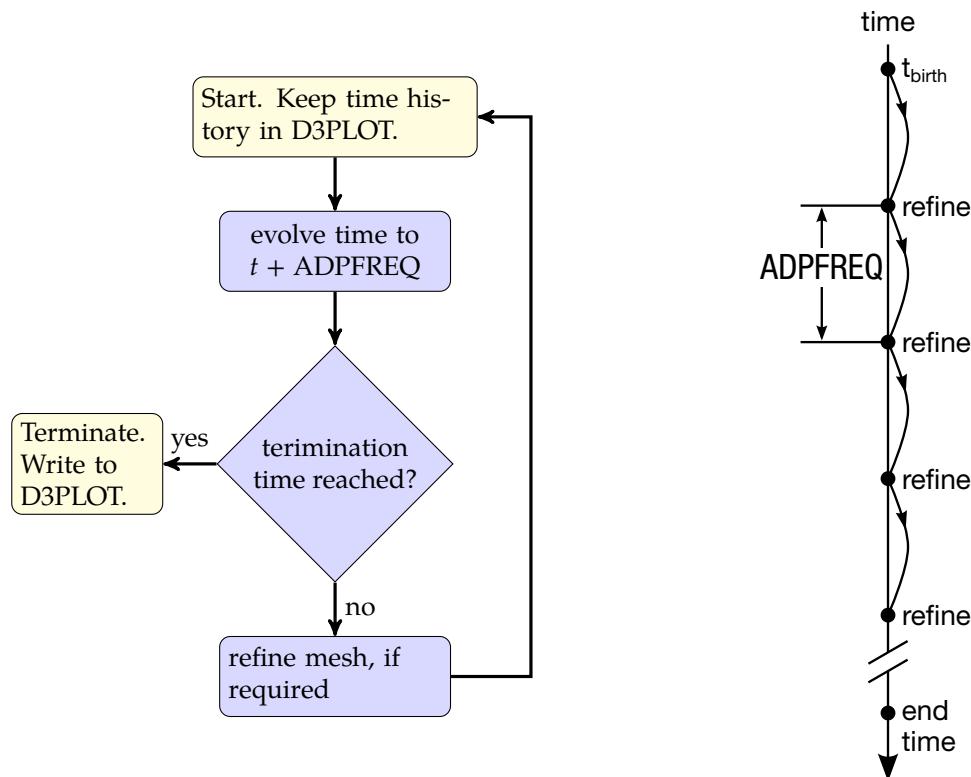


Figure 12-2. Flow chart for ADPASS = 1. This algorithm may be summarized as “periodically refine.” This method is recommended over ADPASS = 0 when used with ADPENE, which implements look ahead.

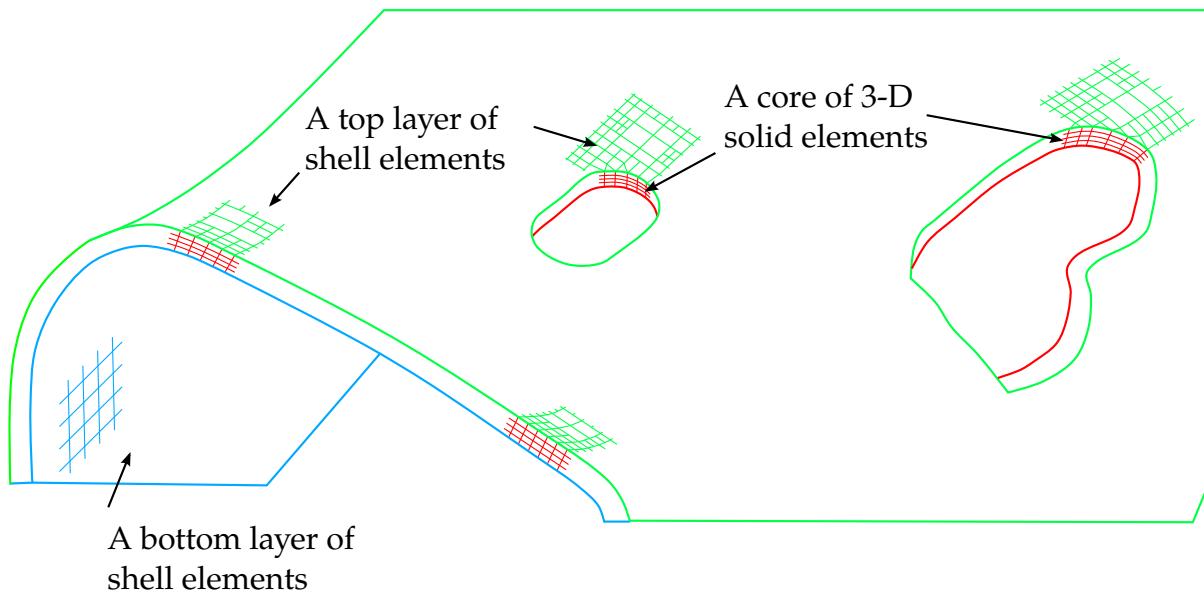


Figure 12-3. Activate sandwich composite mesh adaptivity by setting IF-SAND = 1. Before R11.0, adaptivity is limited to only one layer of solid elements; starting with R11.0 adaptive meshing applies to multiple layers of solid elements.

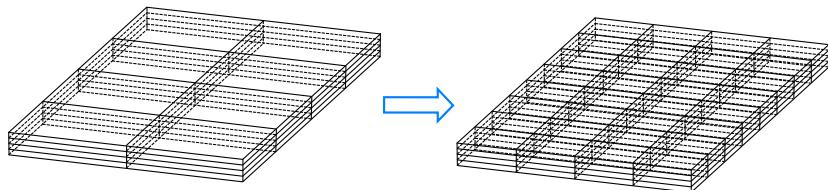


Figure 12-4. Mesh fission for sandwich part. Sweeping the top and bottom adaptive shell elements through the thickness, a total of 32 hexahedral elements are refined into 128 of the same type. The number of layers of the solids always remains the same.

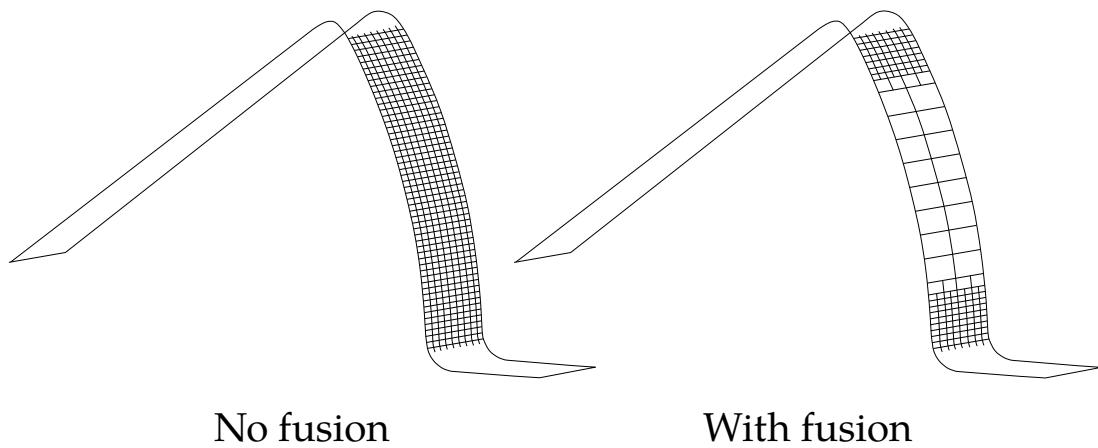


Figure 12-5. Comparison of final mesh pattern between adaptive mesh without fusion and with fusion, on a hat-section draw-bending (half-symmetric model shown).

***CONTROL_ADAPTIVE_CURVE**

Purpose: To refine the element mesh along a curve during or prior to a sheet metal forming simulation. All curves defined by the keyword *DEFINE_CURVE_TRIM are used in the refinement. This option provides additional refinement to that generated by *CONTROL_ADAPTIVE. Additionally, pre-mesh refinement along a curve with specific distance/range on both sides of the curve can be modeled when this keyword is used together with *DEFINE_CURVE_TRIM_3D (by activating the field TCTOL). Lastly, this keyword can be used to refine the mesh along a curve during trimming when used together with the keyword *ELEMENT_TRIM. This feature only applies to shell elements and *h*-adaptivity.

Card 1	1	2	3	4	5	6	7	8
Variable	IDSET	ITYPE	N	SMIN	ITRIOPT			
Type	I	I	I	F	I			

VARIABLE	DESCRIPTION
IDSET	Set ID
ITYPE	Set type: EQ.1: IDSET is shell set ID. EQ.2: IDSET is part set ID.
N	Refinement option: EQ.1: Refine until there are no adaptive constraints remaining in the element mesh around the curve, subjected to the maximum refinement level of 5. GT.1: Refine no more than N levels.
SMIN	If the element dimension is smaller than this value, do not refine.
ITRIOPT	Option to refine an enclosed area of a trim curve. EQ.0: Refine the elements along the trim curve. EQ.1: Refine the elements along the trim curve and enclosed by the trim curve. Under the keyword *DEFINE_CURVE_TRIM_3D, the variable TCTOL must be set to "2" and a seed node NSEED1 must be defined inside the curve loop.

Adaptive mesh refinement along a curve during the beginning of a simulation:

The [Figure 12-6](#) top right mesh refinement illustrates mesh adaptivity along an enclosed curve as done by the partial input example below. Since the mesh refinement is controlled by either the refinement level N or smallest element size SMIN, care should be taken so not too many elements are generated in the model.

The partial keyword input example below refines the mesh by four levels along both sides of the curve defined by the IGES file adpcurves.iges. If an element edge length is shorter than 0.3 mm, then the element is not refined.

```
*INCLUDE
drawn.dynain
*DEFINE_CURVE_TRIM_3D
$    TCID      TCTYPE          TFLG       TDIR       TCTOL
      1           2
adpcurves.iges
*CONTROL_ADAPTIVE_CURVE
$    IDSET      ITYPE          N         SMIN
      1           2           4         0.3
```

Adaptive mesh refinement inside a curve loop during the beginning of a simulation:

[Figure 12-6](#) bottom left mesh refinement illustrates refining the mesh inside the closed curve from the partial input below. The original mesh ([Figure 12-6](#) top left) is refined by either six levels or to no smaller than 3.2 mm element edge length inside the curve loop defined by the IGES file area.iges. TCTYPE may be set to either 1 or 2, but TCTOL must be set to 2, and NSEED1 (106877) must be defined inside the curve loop. Note this feature is available starting in Revision 115142.

```
*KEYWORD
*INCLUDE
incoming.dynain
*CONTROL_TERMINATION
0.0
*parameter
$-----1-----2-----3-----4-----5-----6-----7-----8
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ enter refinement level:
I reflvl      6
$$$$$$$$$$$$$$$$$$$$$$$$$ enter smallest element length:
R minsize     3.2
$-----1-----2-----3-----4-----5-----6-----7-----8
*DEFINE_CURVE_TRIM_3D
$    TCID      TCTYPE          TFLG       TDIR       TCTOL      TRDIS      NSEED1
      1           2                   2           106877
$$$$$ enter IGES curve file name (make sure it's very close to the blank):
area.iges
$-----1-----2-----3-----4-----5-----6-----7-----8
*CONTROL_ADAPTIVE_CURVE
$    PartSET     ITYPE          N         SMIN      ITRIOPT
      1           2     &reflvl   &minsize      1
```

Adaptive mesh along a curve with refinement controlled by a distance during the

beginning of a simulation:

This feature was added to limit the number of elements created.

When *CONTROL_CURVE_TRIM_3D is used with *CONTROL_ADAPTIVE_CURVE, TCTOL is interpreted as the total width the mesh will be refined centered on the defined curve during the beginning of a simulation; in other words, on each side of the curve the mesh will be refined to a distance of half TCTOL. [Figures 12-7](#) and [12-8](#) illustrate this. This feature *only* works with the 3D option.

The curve needs to be sufficiently close to the part. Since the curve is often made from some feature lines of forming tools, the curve must be re-positioned closer to the blank, or better yet, projected onto the blank; otherwise the refinement will not take place. The curve can be moved closer to the part with LS-PrePost 4.0 using *GeoTol* → *Project* → *Closest Proj* → *Project to Element* → *By Part*.

The partial input example below refines the mesh 2.0 mm (TCTOL = 4.0) on each side of the curve defined by the file adpcurves.iges. The maximum refinement level is 4, and the minimum element size allowed is 0.3 mm.

```
*INCLUDE
drawn.dynain
*DEFINE_CURVE_TRIM_3D
$    TCID      TCTYPE        TFLG      TDIR      TCTOL
      1          2            0          0       4.000
adpcurves.iges
*CONTROL_ADAPTIVE_CURVE
$    IDSET      ITYPE        N         SMIN
      1          2            4         0.3
```

Mesh refinement along a curve is very useful during line die simulations. For example, in a flanging simulation, a trimmed blank, that is mostly flat in the flanging break line in draw die, can be refined using a curve generated from the trim post radius. In LS-PrePost 4.0, the curve can be generated using *Curve* → *Spline* → *From Mesh* → *By Edge*, check *Prop*, and defining a large *Ang* to create a continuous curve along element edge. This curve can then be projected onto the blank mesh using *GeoTol* → *Project* feature which will then be used as the curve file adpcurves.iges here. The mesh pre-refinement along curves are implemented in the flanging process starting with LS-PrePost4.0 *eZSetup* for metal forming applications. In LS-PrePost4.3 *eZSetup*, improvements are made so adaptive mesh refinement along a curve can be made without the need to define any tools.

In [Figures 12-9](#), [12-10](#), [12-11](#), [12-12](#)and [12-13](#), mesh pre-refinement along a curve is demonstrated on a fender outer case in which the effect of different TCTOL values can be seen.

The keyword *INCLUDE_TRIM is recommended to be used at all times to include the dynain file from a previous simulation, except when the to-be-adapted sheet blank has no stress and strain information; that is, no *INITIAL_STRESS_SHELL, and *INITIAL_-

STRAIN_SHELL cards present in the sheet blank keyword or dynain file. For this case, the keyword *INCLUDE must be used instead.

Adaptive mesh refinement along a curve during trimming:

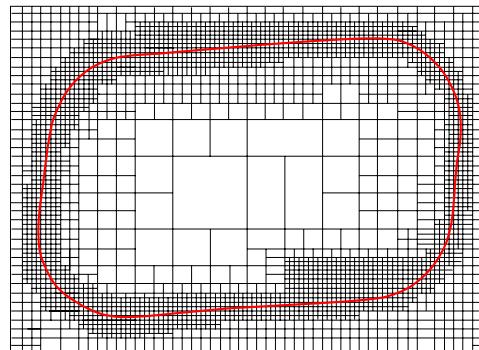
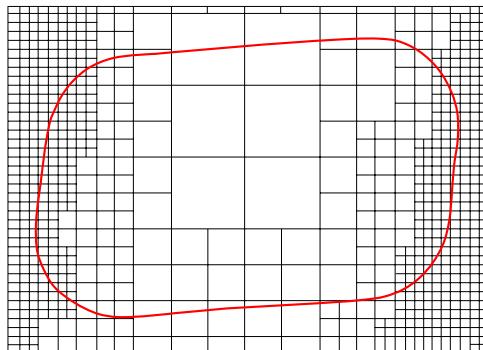
When the keyword *ELEMENT_TRIM is present, this keyword is used to refine meshes during a trimming simulation. Coarse meshes along the trim curve can be refined prior to trimming, causing a more detailed and distinctive trim edge. A partial example input deck is shown below:

```
*INCLUDE_TRIM
drawn.dynain
*ELEMENT_TRIM
    1
*DEFINE_CURVE_TRIM_2D
$#   TCID      TCTYPE        TFLG      TDIR      TCTOL      TOLN      NSEED1      NSEED2
    1          2              0          0.250
doubletrim.iges
*DEFINE_TRIM_SEED_POINT_COORDINATES
$#   NSEED      X1          Y1          Z1          X2          Y2          Z2
    1     -184.565    84.755
*CONTROL_ADAPTIVE_CURVE
$#   IDSET      ITYPE      N      SMIN      ITRIOPT
    1          2          3          3.0          0
*CONTROL_CHECK_SHELL
$#   PSID      IFAUTO      CONVEX      ADPT      ARATIO      ANGLE      SMIN
    1          1          1          1          0.25      150.0      0.18
```

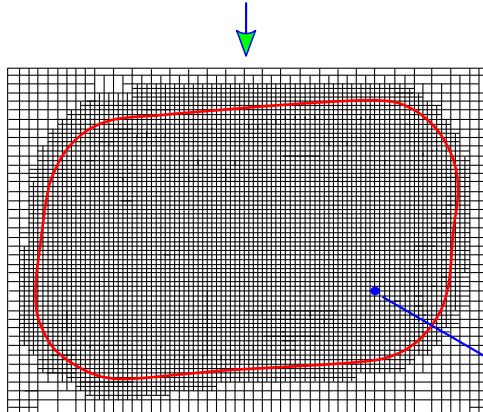
The keyword *ELEMENT_TRIM defines a deformable part set to be trimmed. The keyword *DEFINE_CURVE_TRIM_2D defines the trim curve and type, along with trim tolerance, etc. The keyword *DEFINE_TRIM_SEED_POINT_COORDINATES indicates which side of the part will remain after trimming by specifying a seed node. The keyword *CONTROL_ADAPTIVE_CURVE specifies the adaptive mesh refinement level and minimum element size along the trim curve. Finally, the keyword *CONTROL_CHECK_SHELL repairs and fixes trimmed elements, so they are suitable for next simulation. More details can be found in each of the corresponding keyword manual sections.

Revision Information:

1. Revision 65630: use of TCTOL as a distance for mesh refinement (when used together with *CONTROL_ADAPTIVE_CURVE). TCTOL is the distance from the curve to the edge of the refinement.
2. Revision 113756: some improvements to TCTOL. TCTOL becomes the distance of the entire width of the refinement.
3. Revision 115142: ITROOPT = 1 and TCTOL = 2 are available for mesh refinement along and inside a curve loop.



To refine along a curve, set ITRIOPT=0



To refine along the curve and interior of the curve, set ITRIOPT=1; under the keyword *DEFINE_CURVE_TRIM_3D, set TCTOL=2 and define NSEED1 (a seed node).

NSEED1

Figure 12-6. Mesh refinement along a curve (top right); along a curve and interior of the curve (bottom left).

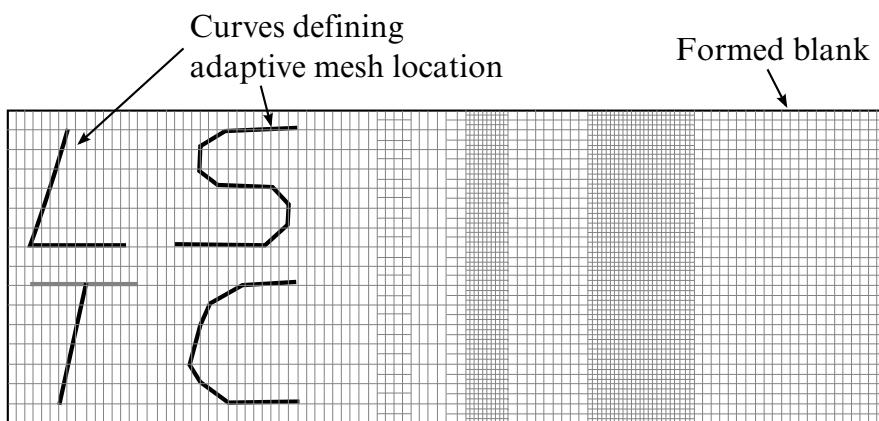


Figure 12-7. Curves can be discontinuous and in one IGES file.

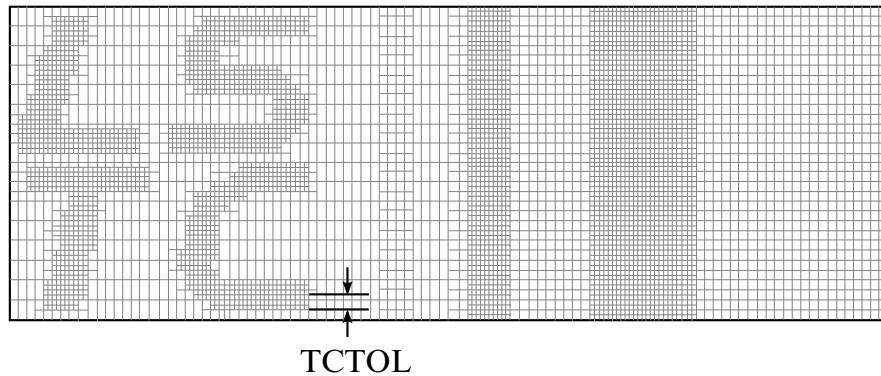


Figure 12-8. Define variable TCTOL to limit the mesh adaptivity area.

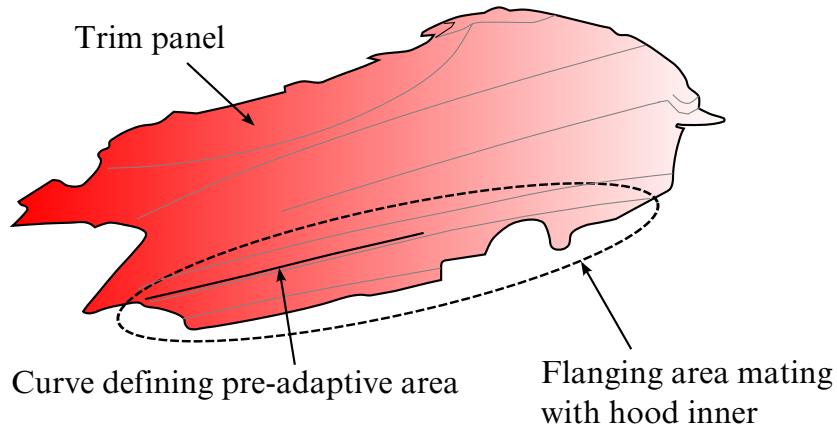


Figure 12-9. A complex mesh refinement example (NUMISHEET2002 Fender).

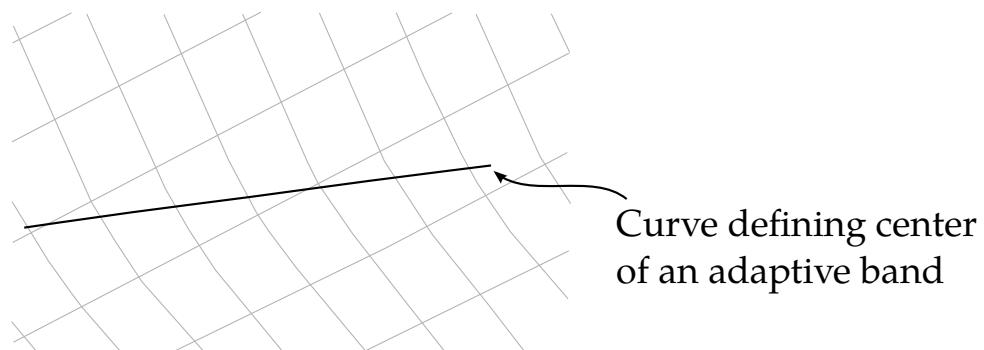


Figure 12-10. Original mesh with target curve defined.

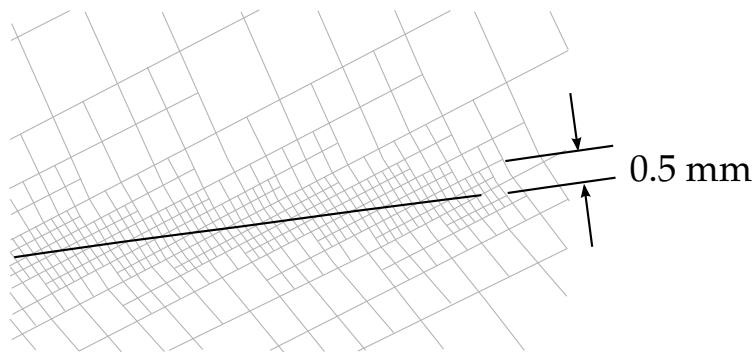


Figure 12-11. Mesh refinement along the target curve with TCTOL = 1.0.

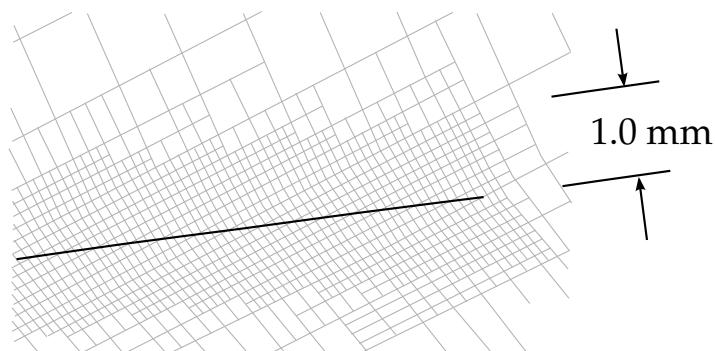


Figure 12-12. Mesh refinement along the target curve with TCTOL = 2.0.

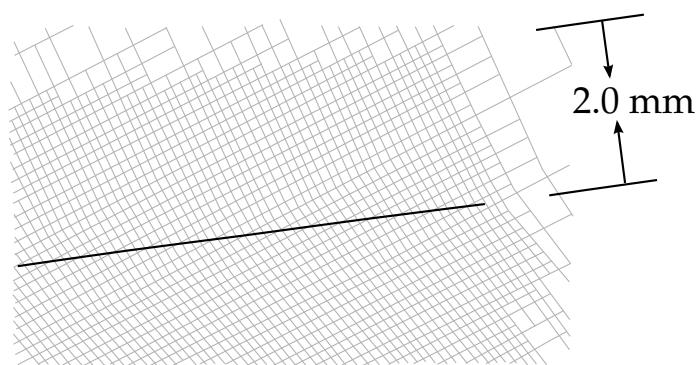


Figure 12-13. Mesh refinement along the target curve with TCTOL = 4.0.

*CONTROL

*CONTROL_AIRBAG

*CONTROL_AIRBAG

Purpose: Global control parameters for CV AIRBAG.

Card 1	1	2	3	4	5	6	7	8
Variable	CKERR							
Type	I							
Default	0							

VARIABLE

DESCRIPTION

CKERR

Flag to check the bag of a CV airbag for (a) open (free) edges and (b) segments not associated with shell or solid elements:

EQ.0: Do not check (default).

EQ.1: Check for free edges and segments not associated with elements. If the airbag surface contains a free edge, LS-DYNA will output the nodes of the free edge to d3hsp, issue a warning, and *continue* the run. If a segment in the segment set defining the airbag is not associated with an element, LS-DYNA will output the segment to d3hsp, issue an error message, and *terminate* the run.

EQ.2: Check for free edges and segments not associated with elements. If the airbag surface contains a free edge, LS-DYNA will output the nodes of the free edge to d3hsp, issue a warning, and *terminate* the run. If a segment in the segment set defining the airbag is not associated with an element, LS-DYNA will output the segment to d3hsp, issue an error message, and *terminate* the run.

Remarks:

CKERR = 1 or 2 causes LS-DYNA to check the integrity of the airbag model. The airbag must be specified as a closed bag (no free edges) using a part set or segments set (SIDTYP of *AIRBAG_OPTION) for correct results. If it has a free edge, then the calculated values for the airbag, such as pressure, volume, and temperature, will be incorrect. If the airbag is defined using a segment set, the segments in the set must be associated with an element. Otherwise, LS-DYNA cannot capture the physics of the airbag.

CONTROL_ALE**CONTROL*****CONTROL_ALE**

Purpose: Set global control parameters for the Arbitrary Lagrangian-Eulerian (ALE) and Eulerian calculations. This command is required when solid element formulation 5, 6, 7, 11, or 12 is used. Parallel processing using SMP is not recommended when using these element formulations; rather, it is better to use MPP for good parallel processing performance. See [*CONTROL_MPP_DECOMPOSITION_DISTRIBUTE_ALE_ELEMENTS](#).

Card 1	1	2	3	4	5	6	7	8
Variable	DCT	NADV	METH	AFAC	BFAC	CFAC	DFAC	EFAC
Type	I	I	I	F	F	F	F	F
Default	1	1	2	0.0	0.0	0.0	0.0	0.0

Card 2	1	2	3	4	5	6	7	8
Variable	START	END	AAFAC	VFACT	PRIT	EBC	PREF	NSIDEBC
Type	F	F	F	F	I	I	F	I
Default	0.0	10^{20}	1.0	10^{-6}	0	0	0.0	none
Remarks			obsolete					

This card is optional.

Card 3	1	2	3	4	5	6	7	8
Variable	NCPL	NBKT	IMASCL	CHECKR	BEAMIN	MMGPREF	PDIFMX	DTMUFAC
Type	I	I	I	F	F	I	F	F
Default	1	50	0	0.0	0.0	0	0.0	0.0

*CONTROL

*CONTROL_ALE

This card is optional.

Card 4	1	2	3	4	5	6	7	8
Variable	OPTIMPP	IALEDR	BNDFLX	MINMAS				
Type	I	I	I	F				
Default	0	0	0	10^{-5}				

VARIABLE	DESCRIPTION
DCT	Flag to invoke alternate advection logic for ALE (see Remark 2): NE.-1: Use default advection logic. EQ.-1: Use alternate (improved) advection logic; generally recommended, especially for simulation of explosives. Note that for S-ALE, DCT is ignored, and the alternative advection option is always used.
NADV	Number of cycles between advects (almost always set to 1).
METH	Advection method: EQ.1: Donor cell with Half Index Shift (HIS), first-order accurate. EQ.2: Van Leer with HIS, second-order accurate (default). EQ.-2: Van Leer with HIS. Additionally, the monotonicity condition is relaxed during the advection process to better preserve *MAT_HIGH_EXPLOSIVE_BURN material interfaces (see Remark 3). EQ.3: Donor cell with HIS modified to conserve total energy over each advection step, in contrast to METH = 1, which conserves internal energy (see Remark 4). EQ.6: Finite volume method with flux-corrected transport. Only supported for ideal gases: the finite volume method is only applied to ALE elements fully filled with materials using *EOS_IDEAL_GAS or *EOS_001 for ideal gases. The advection in mixed ALE elements is handled by a donor cell method.

VARIABLE	DESCRIPTION
AFAC	ALE smoothing weight factor - Simple average EQ.-1.0: Turn smoothing off. See Remark 5 .
BFAC	ALE smoothing weight factor - Volume weighting
CFAC	ALE smoothing weight factor - Isoparametric
DFAC	ALE smoothing weight factor - Equipotential
EFAC	ALE smoothing weight factor - Equilibrium
START	Start time for ALE smoothing or start time for ALE advection if smoothing is not used.
END	End time for ALE smoothing or end time for ALE advection if smoothing is not used. LT.0.0: The ALE mesh is removed after END .
AAFAC	ALE advection factor (donor cell options, default = 1.0). This field is obsolete.
VFACT	Volume fraction limit for stresses in single material and void formulation. All stresses are set to zero for elements with lower volume fraction than VFACT. EQ.0.0: Set to default 10^{-6}
PRIT	A flag to turn on or off the pressure equilibrium iteration option for multi-material elements (see Remark 6): EQ.0: Off (default) EQ.1: On
EBC	Automatic Eulerian boundary condition (see Remark 7): EQ.-2: Generate *ALE_ESSENTIAL_BOUNDARY with slip condition for mesh boundaries that do not already have segment boundary conditions, such as *BOUNDARY_NON_REFLECTING , *LOAD_BLAST_SEGMENT_SET , *BOUNDARY_SPC_SET , or *LOAD_SEGMENT_SET , applied. EQ.0: Off EQ.1: On with stick condition

*CONTROL

*CONTROL_ALE

VARIABLE	DESCRIPTION
	EQ.2: On with slip condition
PREF	A pseudo reference pressure equivalent to an environmental pressure that is applied to the free surfaces of the ALE domain or mesh (see Remark 8)
NSIDEBC	A node set ID (NSID) which is to be excluded from the EBC constraint.
NCPL	Number of Lagrangian cycles between coupling calculations. This is typically done every cycle; therefore, its default is 1.
NBKT	Number of Lagrangian cycles between global bucket-sort searches to locate the position of the Lagrangian structure (mesh) relative to the ALE fluid (mesh). Default is 50. LT.0: NBKT is a *DEFINE_CURVE ID defining a table: time vs NBKT as defined above. EQ.0: NBKT = 50 (default). If the mesh is moving, NBKT is adapted for the buckets to follow the mesh more closely. GT.0: NBKT remains constant.
IMASCL	A flag for turning ON/OFF mass scaling for ALE parts. The global mass scaling control parameter DT2MS for the *CONTROL-TIMESTEP keyword must be nonzero. If the ALE time step becomes smaller than the mass scaling time step (DT2MS), then IMASCL has the following effects: EQ.0: No mass scaling for ALE parts (default). The DTMS time step will be used for the calculation. Note that because the DTMS time step is larger than the ALE time step, the CFL condition will not be satisfied in the ALE domain, potentially causing instability. Print out a maximum of 20 warnings. EQ.1: No mass scaling for ALE parts. Stop the run. EQ.2: Do mass scaling for ALE parts (the result may not be correct due to this scaling). EQ.3: No mass scaling for ALE parts. Use the ALE time step. This time step may be small enough to substantially increase calculation run time.

VARIABLE	DESCRIPTION
CHECKR	A parameter for reducing or eliminating an ALE pressure locking pattern. It may range from 0.01 to 0.1 (See Remark 9).
BEAMIN	Flag to align the dynamics of plain strain and axisymmetric beams in 2D FSI ALE models to their shell counterparts in 3D FSI ALE models: EQ.0.0: Off (default) EQ.1.0: On
MMGPREF	A flag to select the method for assigning a reference pressure to multiple ALE multi-material groups (see Remark 8). EQ.0: Off (default). PREF applies to every AMMG in the model. LT.0: MMGPREF is an ID of either a curve defined using *DEFINE_CURVE or a table defined using *DEFINE_TABLE . If it is a curve, then its abscissa contains the multi-material group IDs (AMMGID), and the ordinates are the constant reference pressure values associated with each of those AMMGIDs. This is the PREF = constant case for each AMMG. If it is a table, then a load curve for PREF as a function of time must be defined for each AMMGID (the value in the table) in the model.
PDIFMX	Maximum of pressure difference between neighboring ALE elements under which the nodal forces are zeroed out: EQ.0: Off (default) GT.0: On
DTMUFAC	Scale a time step called DTMU that depends on the dynamic viscosity μ , the initial density ρ , and an element characteristic length ℓ : $DTMU = \frac{\rho\ell^2}{2\mu}$ DTMU is emitted by the element to the solver as an element time step, thereby making DTMU an upper bound on the global time step. EQ.0: Off (default) GT.0: On

*CONTROL

*CONTROL_ALE

VARIABLE	DESCRIPTION
OPTIMPP	Optimize the MPP communications in the penalty coupling (*CONSTRAINED_LAGRANGE_IN_SOLID , CTYPE = 4) and group ALE parts together for the element processing. EQ.0: Off (default) EQ.1: On
IALEDR	Include ALE computations in the dynamic relaxation analysis (*CONTROL_DYNAMIC_RELAXATION). EQ.0: Off (default) EQ.1: On
BNDFLX	Multi-Material ALE group set ID selecting only the materials in elements at mesh boundaries with influxes that can flow in. By default, when the flow is inwards at the boundary faces of ALE elements, every material in these elements flows in. This option can select only a few of these ALE groups. EQ.0: Off (default) GT.0: *SET_MULTI-MATERIAL_GROUP_LIST ID EQ.-1: No influx
MINMAS	Factor of the minimum mass allowed in an element (see Remark 11): $\text{MINMAS} \times \text{initial density} \times \text{material volume}$

Remarks:

1. **Recommended starting point settings.** Although this keyword has many fields, only a few are required input. As a starting point, you can try setting DCT = -1, NADV = 1, METH = 1, AFAC = -1, and the rest as "0". When needed, PREF should also be defined. These settings are adequate for most cases. Depending on the physics, you may also need to change METH to 2 or 3.
2. **The DCT field.** For the general ALE solver in both 2D and 3D, the DCT field specifies the advection scheme. By default, the solver uses the original advection scheme. Setting DCT = -1 invokes the improved advection scheme. The S-ALE solver in both 2D and 3D always uses the improved advection scheme (DCT is ignored).

We recommend DCT = -1 over the default scheme, especially for simulating explosives. This scheme includes the following major changes:

- a) Relaxes an artificial limit on the expansion ratio limit. The default limit improves stability in some situations but can overestimate the explosive impulse.
 - b) Corrects redundant out-flux of material at corner elements. The redundancy can lead to negative volume.
 - c) Removes several artificial constraints in the advection which were originally implemented to assist in stability but are no longer needed.
3. **METH = -2.** The METH = -2 advection type is the same as METH = 2 with only one exception. It employs a looser constraint on monotonicity requirement during ALE advection. When METH = 2, for each advection process along three directions (front/back, top/bottom, left/right), the maximum/minimum values for advected history variables in the three elements along that direction are capped. METH = -2 relaxed the monotonicity condition so that the advected value is capped at the maximum/minimum value in the element itself and its neighboring 26 elements. This option, in certain conditions, can better preserve the material interface for materials defined with *MAT_HIGH_EXPLOSIVE_BURN.
4. **METH = 3 for conserving total energy.** Generally, it is not possible to conserve both momentum and kinetic energy (KE) at the same time. Typically, internal energy (IE) is conserved and KE may not be. This may result in some KE loss (hence, total energy loss). For many analyses this is tolerable, but for airbag application, this may lead to the reduction of the inflating potential of the inflator gas. METH = 3 tries to eliminate this loss in KE over the advection step by storing any loss KE under IE, thus conserving total energy of the system.
5. **Smoothing factors.** All the smoothing factors (AFAC, BFAC, CFAC, DFAC, EFAC) are generally most applicable to ELFORM = 5 (single material ALE formulation). The ALE smoothing feature is *not* supported by MPP versions.
6. **The PRIT field.** Most of the fast transient applications do not need this feature. It could be used in specific slow dynamic problems for which material constitutive laws with very different compressibility are linear, and the stresses in multi-material elements require to be balanced.
7. **The EBC field.** This option is used for EULER formulations. It automatically defines velocity boundary condition constraints for the user. The constraints, once defined, are applied to all nodes on free surfaces of an Eulerian domain. For problems where the normal velocity of the material at the boundary is zero, such as injection molding problems, the automatic boundary condition

parameter is set to 2. This will play the same role as the nodal single point constraint. For EBC = 1, the material velocity of all free surface nodes of an Eulerian domain is set to zero.

8. **Environmental pressure.** By default, the pressure outside the ALE mesh is assumed to be zero. Any material with a pressure higher than zero will have the tendency to flow out of the ALE domain. Therefore, when there is any ALE material with initialized pressure greater than zero, defined using the *EOS keyword, PREF should be defined to prevent that ALE material from leaving the ALE domain.

To provide the effect of environmental pressure loading on all the free surfaces of the ALE mesh, the code performs an equivalent calculation. It subtracts PREF from the diagonal components of the stress tensor of each material before computing the internal forces. Thus, defining PREF is equivalent to applying *LOAD_SEGMENT cards to balance the internal pressure along the free ALE mesh boundaries. By default, PREF is applied to all the materials in the ALE domain. When PREF is a constant value, then a fixed value of ambient pressure is applied. When PREF is a load curve, then the ambient pressure is a function of time.

MMGPREF cannot be used to set the initial pressure for a material. The initial pressure must be set using the *EOS or *MAT keyword for the material. The shift of the stresses by PREF cannot be seen in the LS-PrePost fringe of the pressures.

When MMGPREF < 0, there are 2 possible cases, assigning (a) PREF = constant or (b) PREF = PREF(*t*) (reference pressure as a function of time) to each AMMG.

[Example 1]

Consider for example, if a model has 3 ALE groups:

- AMMG1 = air (with reference pressure for AMMG1 = PREF1 = 1.0 bar)
- AMMG2 = explosive (with reference pressure for AMMG2 = PREF2 = 0.0 bar)
- AMMG3 = water (with reference pressure for AMMG3 = PREF3 = 0.0 bar)

Here is how it may be defined in the input file.

PREF = 0.0 bar

MMGPREF = -LCID

where LCID is the ID of the following curve:

```
$----+---1----+---2----+---3----+---4----+---5----+---6----+---7----+  
*DEFINE_CURVE  
    LCID  
$      AMMGID      PREF  
        1          1.0  
        2          0.0  
        3          0.0  
$----+---1----+---2----+---3----+---4----+---5----+---6----+---7----+
```

[Example 2]

Consider a model also having 3 ALE groups where each group may have a different reference pressure as a function of time. One scenario may be that of a pre-pressurized container. Another may be the simulation of some reservoir conditions. Assume for this example that we have a pressurized

container. All 3 AMMGs are initialized to 4 bars at $t=0.0$. Then, we lower the environmental pressure of the outside air from 4.0 bars to 1.0 over a short time. Over this duration, the FSI will have time to build up the pre-stressed state in the container before another dynamic process is introduced, such as an impact.

```
- AMMG1 = air outside                                (with PREF1=PREF1(t))
- AMMG2 = pressurized gas    inside container (with PREF2=PREF2(t))
- AMMG3 = pressurized liquid inside container (with PREF3=PREF3(t))

$-----1-----2-----3-----4-----5-----6-----7-----
*DEFINE_TABLE
$      TBID
      101
$      AMMGID
      1
      2
      3
$-----1-----2-----3-----4-----5-----6-----7-----
$ The 1st curve immediately following the table corresponds to AMMG1, the 2nd
$ curve to AMMG2, and the 3rd curve to AMMG3, respectively.
$-----1-----2-----3-----4-----5-----6-----7-----
*DEFINE_CURVE
$      LCID
      11
$      time          PREF1(t)
      0.000          4.0
      0.001          1.0
      1.000          1.0
*DEFINE_CURVE
$      LCID
      12
$      time          PREF2(t)
      0.000          1.0
      0.001          1.0
      1.000          1.0
*DEFINE_CURVE
$      LCID
      13
$      time          PREF3(t)
      0.000          1.0
      0.001          1.0
      1.000          1.0
$-----1-----2-----3-----4-----5-----6-----7-----
```

9. **CHECKR field for one-point integration.** Due to one-point integration, ALE elements may experience spatial instability in the pressure field referred to as checkerboarding. CHECKR is a scale for diffusive flux calculation to alleviate this problem.
10. **Pressure checkerboarding.** Because the internal forces are located at the nodes, while the pressure is stored at the element center, sometimes a "checkerboard pattern" arises in the pressure distribution. It is a kind of locking effect that normally occurs only in problems having very small volumetric strains, i.e., at small pressures. CHECKR is designed to alleviate this problem.
11. **MINMAS for materials with large expansions.** Materials with large expansions could have element masses smaller than MINMAS. In such cases, MINMAS needs to be decreased to avoid artificial mass increases.

*CONTROL

*CONTROL_BULK_VISCOSITY

*CONTROL_BULK_VISCOSITY

Purpose: Reset the default values of the bulk viscosity coefficients globally. This may be advisable for shock wave propagation and some materials. Bulk viscosity is used to treat shock waves. 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.

Card 1	1	2	3	4	5	6	7	8
Variable	Q1	Q2	TYPE	BTYPEn	TSTYPEn			
Type	F	F	I	I	I			
Default	1.5	.06	1	0	0			

VARIABLE	DESCRIPTION
Q1	Default quadratic viscosity coefficient.
Q2	Default linear viscosity coefficient.
TYPE	Default bulk viscosity type, IBQ (default = 1): EQ.-2: Same as -1 but the internal energy dissipated by the viscosity in the shell elements is computed and included in the overall energy balance. EQ.-1: Same as 1 but also includes viscosity in shell formulations 2, 4, 10, 16, and 17. The internal energy is not computed in the shell elements. EQ.1: Standard bulk viscosity. Solid elements only and internal energy is always computed and included in the overall energy balance. EQ.2: Richards-Wilkins bulk viscosity. Two-dimensional plane strain and axisymmetric solid elements only. Internal energy is always computed and included in the overall energy balance.
BTYPEn	Beam bulk viscosity type (default = 0): EQ.0: The bulk viscosity is turned off for beams.

VARIABLE	DESCRIPTION
	<p>EQ.1: The bulk viscosity is turned on for beam types 1 and 11. The energy contribution is not included in the overall energy balance.</p> <p>EQ.2: The bulk viscosity is turned on for beam type 1 and 11. The energy contribution is included in the overall energy balance.</p>
TSTYPE	<p>Bulk viscosity for thick shells (default = 0):</p> <p>EQ.0: The bulk viscosity is turned off for thick shells.</p> <p>EQ.1: The bulk viscosity is turned on for thick shells 5, 6, and 7.</p>

Remarks:

The bulk viscosity creates an additional additive pressure term given by:

$$q = \begin{cases} \rho l(Q_1 l \dot{\varepsilon}_{kk}^2 - Q_2 a \dot{\varepsilon}_{kk}) & \dot{\varepsilon}_{kk} < 0 \\ 0 & \dot{\varepsilon}_{kk} \geq 0 \end{cases}$$

where Q_1 and Q_2 are dimensionless input constants which default to 1.5 and 0.06, respectively, l is a characteristic length given as the square root of the area in two dimensions and as the cube root of the volume in three, and a is the local sound speed. See Chapter 21 in the LS-DYNA Theory Manual for more details.

The Richards-Wilkins, see [Richards 1965, Wilkins 1976], bulk viscosity considers the directional properties of the shock wave which has the effect of turning off the bulk viscosity in converging geometries minimizing the effects of “ q -heating.” The standard bulk viscosity is active whenever the volumetric strain rate is undergoing compression even though no shock waves are present.

*CONTROL

*CONTROL_CHECK_SHELL

*CONTROL_CHECK_SHELL

Purpose: Check for various problems in the mesh after trimming for metal forming simulations. It only fixes boundary elements. This keyword should be included in a trimming input deck.

Part cards. Include one card for each part or part set to be checked. The next keyword ("**") card terminates this input.

Card 1	1	2	3	4	5	6	7	8
Variable	PSID	IFAUTO	CONVEX	ADPT	ARATIO	ANGLE	SMIN	
Type	I	I	I	I	F	F	F	
Default	0	0	1	1	0.25	150.0	0.0	

VARIABLE	DESCRIPTION
PSID	Part or part set ID to be checked: EQ.0: Do not check GT.0: Part ID LT.0: Part set ID
IFAUTO	Flag to automatically correct bad elements: EQ.0: Write warning message only EQ.1: Fix bad element, write message
CONVEX	Check element convexity (internal angles less than 180 degrees). See Remark 2 . EQ.0: Do not check EQ.1: Check
ADPT	Check adaptive constraints: EQ.0: Do not check EQ.1: Check
ARATIO	Minimum allowable aspect ratio. Elements which do not meet minimum aspect ratio test will be treated according to IFAUTO above.

VARIABLE	DESCRIPTION
ANGLE	Maximum allowable internal angle. Elements which fail this test will be treated according to IFAUTO above.
SMIN	Minimum element size. Elements which fail this test will be treated according to IFAUTO above. See Remark 4 .

Remarks:

1. **Metal Forming Applications.** Shell element integrity checks which have been identified as important in metal forming applications are performed. These checks can improve springback convergence and accuracy. This keyword will repair bad elements created, for example, during trimming operations.
2. **Convexity.** If the convexity test is activated, all failed elements will be fixed regardless of IFAUTO.
3. **Mesh Connectivities.** In addition to illegal constraint definitions, checks are performed for mesh connectivities which have been found to cause convergence trouble in implicit springback applications.
4. **SMIN.** Variable SMIN should be set between 1/4 and 1/3 of the smallest pre-trim element length.

*CONTROL

*CONTROL_COARSEN

*CONTROL_COARSEN

Purpose: Adaptively de-refine (coarsen) a shell mesh by selectively merging four adjacent elements into one. Adaptive constraints are added and removed as necessary.

Card 1	1	2	3	4	5	6	7	8
Variable	ICOARSE	ANGLE	NSEED	PSID	SMAX			
Type	I	F	I	I	F			
Default	0	none	0	0	0			

Card 2	1	2	3	4	5	6	7	8
Variable	N1	N2	N3	N4	N5	N6	N7	N8
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

VARIABLE	DESCRIPTION
ICOARSE	Coarsening flag: EQ.0: do not coarsen (default) EQ.1: coarsen mesh at beginning of simulation for forming model EQ.2: coarsen mesh at beginning of simulation for crash model
ANGLE	Allowable angle change between neighboring elements. Adjacent elements which are flat to within ANGLE degrees are merged. Suggested starting value is 8.0 degrees.
NSEED	Number of seed nodes (optional). EQ.0: use only automatic searching. GT.0: the number of seed nodes with which to supplement the search algorithm. See Remark 2 . NSEED must be an integer less than or equal to 8.

VARIABLE	DESCRIPTION
PSID	Part set ID. All the parts defined in this set will be prevented from being coarsened.
SMAX	Maximum element size. For ICOARSE = 2, no elements larger than this size will be created.
N1, ..., N8	Optional list of seed node IDs for extra searching. If no seed nodes are specified, leave Card 2 blank.

Remarks:

1. **Coarsened Mesh Input Deck.** Coarsening is performed at the start of a simulation. The first plot state represents the coarsened mesh. By setting the termination time to zero and including the keyword *INTERFACE_SPRINGBACK_LSDYNA, a keyword input deck can be generated containing the coarsened mesh.
2. **Seed Nodes.** By default, an automatic search is performed to identify elements for coarsening. In some meshes, isolated regions of refinement may be overlooked. Seed nodes can be identified in these regions to assist the automatic search. Seed nodes identify the central node of a four-element group which is coarsened into a single element if the angle criterion is satisfied.
3. **Non-Coarsened Regions.** The keyword *DEFINE_BOX_COARSEN can be used to indicate regions of the mesh which are protected from coarsening.

*CONTROL

*CONTROL_CONSTRAINED

*CONTROL_CONSTRAINED

Purpose: Define global control parameters for constraint-related properties.

Card 1	1	2	3	4	5	6	7	8
Variable	SPRCHK	SPRSMD	SPRSRCH					
Type	I	I	I					
Default	0	0	0					

VARIABLE	DESCRIPTION
SPRCHK	SPR2/SPR3 initialization check: EQ.0: Automatically increase search radius to find enough nodes (default) EQ.1: Same as 0 but also write a warning EQ.2: Error termination if not enough nodes found immediately
SPRSMD	Shear moment distribution behavior for SPR3: EQ.0: Distributed as force pairs EQ.1: Distributed as nodal moments (old behavior)
SPRSRCH	Search method for SPR2 and SPR3: EQ.0: Include nodes inside the search radius. EQ.1: Include not only nodes inside the search radius but also all nodes from elements inside the search radius. This search method generally leads to more nodes involved in the connection. In addition, this option invokes a different search algorithm from SPRSRCH = 0 when solid element parts are involved: nodes on the surface facing the SPR connector are detected, leading to automatically finding enough nodes. Because the default algorithm depends on a user-specified search cylinder, it sometimes does not detect enough nodes for coarse meshes. Also, sometimes, the default algorithm finds nodes beneath the surface of the solid element part. Thus, this alternative algorithm for elements leads to more robust search results in general.

CONTROL_CONTACT**CONTROL*****CONTROL_CONTACT**

Purpose: Change defaults for computation with contact surfaces.

Card Summary:

Unless noted otherwise in the description of the field, the optional cards (Cards 3 through 7) apply only to the following contact types:

SINGLE_SURFACE

AUTOMATIC_GENERAL

AUTOMATIC_SINGLE_SURFACE

AUTOMATIC_NODES_TO....

AUTOMATIC_SURFACE_...

AUTOMATIC_ONE_WAY_...

ERODING_SINGLE_SURFACE

The friction coefficients SFRIC, DFRIC, EDC, and VFC are active only when *PART_CONTACT is invoked with FS = -1 in *CONTACT, and the corresponding frictional coefficients in *PART_CONTACT are set to zero. This keyword's TH, TH_SF, and PEN_SF override the corresponding parameters in *CONTACT but will not override corresponding nonzero parameters in *PART_CONTACT.

Card 1. This card is required.

SLSFAC	RWPNAL	ISLCHK	SHLTHK	PENOPT	THKCHG	ORIEN	ENMASS
--------	--------	--------	--------	--------	--------	-------	--------

Card 2. This card is required.

USRSTR	USRFRC	NSBCS	INTERM	XPENE	SSTHK	ECDT	TIEDPRJ
--------	--------	-------	--------	-------	-------	------	---------

Card 3. This card is optional.

SFRIC	DFRIC	EDC	VFC	TH	TH_SF	PEN_SF	PTSCL
-------	-------	-----	-----	----	-------	--------	-------

Card 4. This card is optional.

IGNORE	FRCENG	SKIPRWG	OUTSEG	SPOTSTP	SPOTDEL	SPOTHIN	DIR_TIE
--------	--------	---------	--------	---------	---------	---------	---------

*CONTROL

*CONTROL_CONTACT

Card 5. This card is optional.

ISYM	NSEROD	RWGAPS	RWGDTH	RWKSF	ICOV	SWRADF	ITHOFF
------	--------	--------	--------	-------	------	--------	--------

Card 6. This card is optional.

SHLEDG	PSTIFF	ITHCNT	TDCNOF	FTALL		SHLTRW	IGACTC
--------	--------	--------	--------	-------	--	--------	--------

Card 7. This card is optional.

IREVSPT		COHTIEM	TIEOPT	STROBJ			
---------	--	---------	--------	--------	--	--	--

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	SLSFAC	RWPNAL	ISLCHK	SHLTHK	PENOPT	THKCHG	ORIEN	ENMASS
Type	F	F	I	I	I	I	I	I
Default	.1	none	1	0	1	0	1	0

VARIABLE	DESCRIPTION
SLSFAC	Scale factor for sliding interface penalties, SLSFAC: EQ.0.0: Default = 0.1.
RWPNAL	Scale factor for rigid wall penalties (see *RIGIDWALL) that treats nodal points interacting with rigid walls. The penalties are set so that an absolute value of unity should be optimal; however, this penalty value may be very problem dependent. If rigid/deformable materials switching is used, this option should be used if the switched materials are interacting with rigid walls. If you have IGA parts in your model, see Remark 10 . LT.0.0: All nodes are treated by the penalty method. This is set to -1.0 for implicit calculations. Since seven (7) variables are stored for each possible tracked node (see NSID on *RIGIDWALL_PLANAR/GEOMETRIC), only the nodes that may interact with the wall should be included in the node list. EQ.0.0: The constraint method is used and nodal points which

VARIABLE	DESCRIPTION
	belong to rigid bodies are not considered.
	GT.0.0: Rigid body nodes are treated by the penalty method, and all other nodes are treated by the constraint method.
ISLCHK	Initial penetration check in contact surfaces with indication of initial penetration in output files (see Remark 3): EQ.1: No checking (default) EQ.2: Full check of initial penetration is performed.
SHLTHK	Flag for consideration of shell thickness offsets in non-automatic surface-to-surface and non-automatic nodes-to-surface type contacts. Shell thickness offsets are always included in single surface, constraint-based, automatic surface-to-surface, and automatic nodes-to-surface contact types (see Remarks 1 and 2): EQ.0: Thickness is not considered. EQ.1: Thickness is considered, but rigid bodies are excluded. EQ.2: Thickness is considered, including rigid bodies.
PENOPT	Penalty stiffness value option (applies to the Standard Penalty Formulation and the Soft Constraint Penalty Formulation, that is, SOFT = 0 and 1 on *CONTACT_OPTION). For default calculation of the penalty value, please refer to the LS-DYNA Theory Manual. EQ.1: Minimum of reference segment and tracked node (default for most contact types) EQ.2: Use reference segment stiffness (old way). EQ.3: Use tracked node value. EQ.4: Use tracked node value, area or mass weighted. EQ.5: Same as 4 but inversely proportional to the shell thickness. This may require special scaling and is not generally recommended. PENOPT = 4 and 5 can be used for metal forming calculations. In general, PENOPT = 2 - 5 should be avoided if both the tracked nodes and the reference segments belong to deformable parts.
THKCHG	Shell thickness changes considered in single surface contact (see Remark 1):

*CONTROL

*CONTROL_CONTACT

<u>VARIABLE</u>	<u>DESCRIPTION</u>
	<p>EQ.0: No consideration (default)</p> <p>EQ.1: Shell thickness changes are included.</p> <p>EQ.2: Applies to MPP only. Shell thickness changes are included, but a different algorithm is used than for THK-CHG = 1. This method is more consistent with the way the initial contact thickness is computed.</p>
ORIEN	<p>Optional automatic reorientation of contact interface segments during initialization. See Remark 4.</p> <p>EQ.1: Active for automated (part) input only (default). Contact surfaces are given by *PART definitions.</p> <p>EQ.2: Active for manual (segment) and automated (part) input</p> <p>EQ.3: Inactive for non-forming contact</p> <p>EQ.4: Inactive for *CONTACT_FORMING types and *CONTACT_DRAWBEAD</p>
ENMASS	<p>Flag for treatment of eroded nodes in contact. An eroded node is defined as a node that is no longer attached to any element owing to element deletion. ENMASS is not supported by all contact types; it is suggested that the user toggle on “Show Deleted Nodes” in LS-PrePost when postprocessing to display eroded nodes as particles, and in so doing, determine if ENMASS affects the contact behavior. ENMASS is not supported when SOFT = 2 on Optional Card A of *CONTACT.</p> <p>EQ.0: Eroded nodes are not considered in the contact algorithm.</p> <p>EQ.1: Eroded nodes of solid elements remain active in the contact algorithm.</p> <p>EQ.2: Eroded nodes of solid and shell elements remain active in the contact algorithm.</p>

CONTROL_CONTACT**CONTROL**

Card 2	1	2	3	4	5	6	7	8
Variable	USRSTR	USRFRC	NSBCS	INTERM	XPENE	SSTHK	ECDT	TIEDPRJ
Type	I	I	I	I	F	I	I	I
Default	0	0	10-100	0	4.0	0	0	0

VARIABLE	DESCRIPTION
USRSTR	Storage per contact interface for user supplied interface control subroutine; see Appendix F. If zero, no input data is read, and no interface storage is permitted in the user subroutine. This storage should be large enough to accommodate input parameters and any history data. This input data is available in the user supplied subroutine.
USRFRC	Storage per contact interface for user supplied interface friction subroutine; see Appendix G. If zero, no input data is read, and no interface storage is permitted in the user subroutine. This storage should be large enough to accommodate input parameters and any history data. This input data is available in the user supplied subroutine.
NSBCS	Number of cycles between contact searching using three-dimensional bucket searches. Using the default value for this field is strongly recommended. For mortar contact (option MORTAR on the CONTACT card), the default is 100. For MPP, this field is ignored when SOFT = 0 and 1, and only BCKT on MPP 1 of *CONTACT_OPTION_... applies in those cases.
INTERM	Flag for intermittent searching in old surface-to-surface contact using the interval specified as NSBCS above: EQ.0: Off EQ.1: On
XPENE	Contact surface maximum penetration check multiplier. If the small penetration checking option, PENCHK, on the contact surface control card is active, then nodes whose penetration then exceeds the product of XPENE and the element thickness are set free, see *CONTACT_OPTION_...

CONTROL**CONTROL_CONTACT**

VARIABLE	DESCRIPTION
	EQ.0.0: Set to default which is 4.0.
SSTHK	Flag for determining default contact thickness for shells in single surface contact types. This variable does not apply when SOFT = 2. See Remark 2 . EQ.0: Default contact thickness may be controlled by shell thickness or by shell edge length (default). EQ.1: Default contact thickness is equal to the shell thickness.
ECDT	Time step size override for eroding contact: EQ.0: Contact time size may control DT. EQ.1: Contact is not considered in DT determination.
TIEDPRJ	Bypass projection of SURFA nodes to SURFB surface in types: *CONTACT_TIED_NODES_TO_SURFACE *CONTACT_TIED_SHELL_EDGE_TO_SURFACE *CONTACT_TIED_SURFACE_TO_SURFACE Tied interface options: EQ.0: Eliminate gaps by projection nodes EQ.1: Bypass projection. Gaps create rotational constraints which can substantially affect results.

This card is optional.

Card 3	1	2	3	4	5	6	7	8
Variable	SFRIC	DFRIC	EDC	VFC	TH	TH_SF	PEN_SF	PTSCL
Type	F	F	F	F	F	F	F	F
Default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

VARIABLE	DESCRIPTION
SFRIC	Default static coefficient of friction (see *PART_CONTACT)

CONTROL_CONTACT**CONTROL**

VARIABLE	DESCRIPTION
DFRIC	Default dynamic coefficient of friction (see *PART_CONTACT)
EDC	Default exponential decay coefficient (see *PART_CONTACT)
VFC	Default viscous friction coefficient (see *PART_CONTACT)
TH	Default contact thickness (see *PART_CONTACT)
TH_SF	Default thickness scale factor (see *PART_CONTACT)
PEN_SF	Default local penalty scale factor (see *PART_CONTACT)
PTSCL	Scale factor on the contact stress exerted onto shell formulations 25, 26, and 27. When DOF = 3 the scale factor also applies to shell formulations 2, 4, and 16.

This card is optional.

Card 4	1	2	3	4	5	6	7	8
Variable	IGNORE	FRCENG	SKIPRWG	OUTSEG	SPOTSTP	SPOTDEL	SPOTHIN	DIR_TIE
Type	I	I	I	I	I	I	F	I
Default	0	0	0	0	0	0	inactive	0

VARIABLE	DESCRIPTION
IGNORE	<p>Ignore initial penetrations for the *CONTACT_AUTOMATIC options. In the SMP contact this flag is not implemented for the AUTOMATIC_GENERAL option. “Initial” in this context refers to the first time step that a penetration is encountered. This option can also be specified for each interface on the third optional card under the keyword, *CONTACT. The value defined here will be the default.</p> <p>EQ.0: Move nodes to eliminate initial penetrations in the model definition.</p> <p>EQ.1: Allow initial penetrations to exist by tracking the initial penetrations.</p> <p>EQ.2: Allow initial penetrations to exist by tracking the initial penetrations. However, penetration warning messages</p>

***CONTROL**

***CONTROL_CONTACT**

VARIABLE	DESCRIPTION
	are printed with the original coordinates and the recommended coordinates of each penetrating node given.
FRCENG	<p>Flag to activate the calculation of frictional sliding energy:</p> <p>EQ.0: Do not calculate.</p> <p>EQ.1: Calculate frictional energy in contact and store as "Surface Energy Density" in the binary INTFOR file. Convert mechanical frictional energy to heat when doing a coupled thermal-mechanical problem. When PKP_SEN = 1 on the keyword card *DATABASE_EXTENT_BINARY, it is possible to identify the energies generated on the upper and lower shell surfaces, which is important in metal forming applications. This data is mapped after each <i>h</i>-adaptive remeshing.</p> <p>EQ.2: Same behavior as above (set to 1) except that frictional energy is not converted to heat.</p>
SKIPRWG	<p>Flag not to display stationary rigid wall by default:</p> <p>EQ.0: Generate 4 extra nodes and 1 shell element to visualize a stationary planar rigid wall.</p> <p>EQ.1: Do not generate stationary rigid walls.</p>
OUTSEG	<p>Flag to determine whether to output for contact type *CONTACT_SPOTWELD each beam spot weld SURFA node and its SURFB segment into the d3hsp file:</p> <p>EQ.0: No, do not write out this information.</p> <p>EQ.1: Yes, write out this information.</p>
SPOTSTP	<p>Flag to determine whether an error termination should occur if a spot weld node or face, which is related to a *MAT_SPOTWELD beam or solid element, respectively, cannot be found on the SURFB surface:</p> <p>EQ.0: No error termination. Silently delete the weld and continue.</p> <p>EQ.1: Print error message and terminate.</p> <p>EQ.2: No error termination. Delete the weld, print a message, and continue.</p> <p>EQ.3: No error termination; keep the weld. This is not</p>

VARIABLE	DESCRIPTION
	recommended as it can lead to instabilities.
SPOTDEL	<p>This option controls the behavior of spot welds when the parent element erodes. When SPOTDEL is set to 1, the beam or solid spot weld is deleted, and the tied constraint is removed when the parent element erodes. The parent element is the element to which the SURFA node is attached using the TIED interface. This option also works for SPRs, namely, they automatically fail if at least one of the parent elements fails. To avoid instabilities, this option is recommended to be set to 1 for any situation in which the parent element is expected to erode.</p> <p>EQ.0: Do not delete the spot weld beam or solid element or SPR.</p> <p>EQ.1: Delete the spot weld elements or SPRs when the attached shells on one side of the element fail.</p> <p>GT.1: Delete the SPR when SPOTDEL nodes are attached to failed elements in the search radius.</p> <p>On vector processors, this option can significantly slow down the calculation if many weld elements fail since the vector lengths are reduced. On non-vector processors the cost-penalty is minimal.</p>
SPOTHIN	<p>Optional thickness scale factor. If active, define a factor greater than zero, but less than one. Premature failure of spot welds can occur due to contact of the spot welded parts in the vicinity of the spot weld. This contact creates tensile forces in the spot weld.</p> <p>Although this may seem physical, the compressive forces generated in the contact are large enough to fail the weld in tension before failure is observed in an experimental test. With this option, the thickness of the parts in the vicinity of the weld is automatically scaled, the contact forces do not develop, and the problem is avoided. We recommend setting the IGNORE option to 1 or 2 if SPOTHIN is active. In MPP, this option applies to all non-Mortar contacts that have SINGLE_SURFACE, AUTOMATIC_GENERAL, SURFACE_TO_SURFACE, or NODES_TO_SURFACE in the name. In SMP it only applies to the AUTOMATIC_SINGLE_SURFACE option. See Remark 5.</p>
DIR_TIE	Directional tie for <i>MPP non-groupable</i> tied contacts. If this flag is set to 1, then each node in the SURFA side of a tied contact is associated with outward normal vectors. A node belonging to the surface of a shell, solid, or thick shell surface is associated with one or more vectors depending on the angle between adjacent segments.

*CONTROL

*CONTROL_CONTACT

VARIABLE	DESCRIPTION							
	Typically the node of a flat surface will have <i>one</i> normal vector, the node of an edge will have <i>two</i> normal vectors, and the node of a corner will have <i>three</i> normal vectors. Nodes on a cylinder may have <i>one or two</i> normal vectors, depending on the mesh resolution of the faceted surface. When deciding which SURFB segment to tie to, the algorithm gives preference to those segments for which the direction from the SURFA node to the SURFB segment has a positive dot product with at least one of the aforementioned normal vectors. Therefore, a SURFA node will not necessarily tie to the closest SURFB segment, but to the “correct” segment. This feature avoids nonphysical tie situations or even zero solid element volumes as a result.							

This card is optional.

Card 5	1	2	3	4	5	6	7	8
Variable	ISYM	NSEROD	RWGAPS	RWGDTH	RWKSF	ICOV	SWRADF	ITHOFF
Type	I	I	I	F	F	I	F	I
Default	0	0	1	0.	1.0	0	0.	0

VARIABLE	DESCRIPTION
ISYM	Symmetry plane option default for automatic segment generation when contact is defined by part IDs: LT.0: ISYM is a node set on the symmetry boundary, supported and recommended for Mortar contact. This will allow for a correct treatment of segments close to the symmetry face/edge. See Remark 8 . EQ.0: Off EQ.1: Do not include faces with normal boundary constraints (for example, segments of brick elements on a symmetry plane). This option is important to retain the correct boundary conditions in the model with symmetry.

VARIABLE	DESCRIPTION
NSEROD	Flag to invoke old method for ERODING_NODES_TO_SURFACE (SMP only): EQ.0: Use two-way algorithm (default) EQ.1: Use one-way algorithm (old method)
RWGAPS	Flag to add rigid wall gap stiffness (see parameter RWGDTH below): EQ.1: Add gap stiffness (default). EQ.2: Do not add gap stiffness.
RWGDTH	Death time for gap stiffness. After this time the gap stiffness is no longer added.
RWKSF	Rigid wall penalty scale factor for contact with deformable parts during implicit calculations. This value is independent of SLSFAC and RWPNAL. If RWKSF is also specified in *RIGIDWALL_PLANNER, the stiffness is scaled by the product of the two values.
ICOV	Invokes the covariant formulation of Konyukhov and Schweizerhof in the FORMING contact option. EQ.0: Standard formulation (default) EQ.1: Covariant contact formulation
SWRADF	Spot weld radius scale factor for neighbor segment thinning: EQ.0: Neighbor segments are not thinned (default). GT.0: The radius of a beam spot weld is scaled by SWRADF when searching for close neighbor segments to thin.
ITHOFF	Flag for offsetting thermal contact surfaces for thick thermal shells: EQ.0: No offset: If thickness is not included in the contact, the heat will be transferred between the mid-surfaces of the corresponding contact segments (shells). EQ.1: Offsets are applied so that contact heat transfer is always between the outer surfaces of the contact segments (shells).

*CONTROL

*CONTROL_CONTACT

This card is optional.

Card 6	1	2	3	4	5	6	7	8
Variable	SHLEDG	PSTIFF	ITHCNT	TDCNOF	FTALL		SHLTRW	IGACTC
Type	I	I	I	I	I		F	I
Default	0	0	0	0	0		0.	0

VARIABLE	DESCRIPTION
SHLEDG	Flag for assuming edge shape for shells when measuring penetration. This field is available for segment-to-segment contact (see SOFT = 2 on *CONTACT) EQ.0: Shell edges are assumed round (default). EQ.1: Shell edges are assumed square and are flush with the nodes.
PSTIFF	Flag to choose the method for calculating the penalty stiffness. This field is available for segment-to-segment contact (see SOFT = 2 on *CONTACT). See Remark 6 . EQ.0: Based on material density and segment dimensions (default). EQ.1: Based on nodal masses or on material density and segment dimensions. The segment mass is taken as the larger of the two values calculated.
ITHCNT	Thermal contact heat transfer methodology: LT.0: Conduction evenly distributed (pre R4) EQ.0: Set to default which is 1 EQ.1: Conduction weighted by shape functions, reduced integration EQ.2: Conduction weighted by shape functions, full integration
TDCNOF	Tied constraint offset contact update option: EQ.0: Update velocities and displacements from accelerations. EQ.1: Update velocities and accelerations from displacements. This option is recommended only when there are large

CONTROL_CONTACT**CONTROL**

VARIABLE	DESCRIPTION
	angle changes where the default does not maintain a constant offset to a small tolerance. This latter option is not as stable as the default and may require additional damping for stability. See *CONTROL_BULK_VISCOSITY and *DAMPING_PART_STIFFNESS.
FTALL	<p>Option to output contact forces to RCFORC for all two surface force transducers when the force transducer surfaces overlap. See Remark 7.</p> <p>EQ.0: Output to the first force transducer that matches (default).</p> <p>EQ.1: Output to all force transducers that match.</p>
SHLTRW	Optional shell thickness scale factor for contact with rigid walls. Shell thickness is not considered when SHLTRW = 0.0 (default). SHLTRW = 0.5 will result in an offset of half a shell thickness in contact with rigid walls.
IGACTC	<p>Options to use isogeometric shells for contact detection for contact involving isogeometric shells:</p> <p>EQ.0: Contact between interpolated nodes and interpolated shells</p> <p>EQ.1: Contact between interpolated nodes and isogeometric shells</p>

This card is optional.

Card 7	1	2	3	4	5	6	7	8
Variable	IREVSP		COHTIEM	TIEOPT	STROBJ			
Type	I		I	I	I			
Default	0		0	0	0			

VARIABLE	DESCRIPTION
IREVSP	<p>Flag to revert the spot weld thinning behavior where beam and brick spot welds share nodes with shell parts instead of being tied to the shells:</p> <p>EQ.0: Thinning at shared nodes will be done as it has been in all</p>

*CONTROL

*CONTROL_CONTACT

VARIABLE	DESCRIPTION
	versions after R9.3.1. EQ.1: Behavior reverts to that of R9.3.1. In this version and previous versions, spot weld thinning was not done.
COHTIEM	Flag to treat how the mass from SURFB of a tied contact affects the time step estimation of cohesive elements: EQ.0: No treatment EQ.1: Assuming the cohesive element's nodes are on SURFA of a tied contact, LS-DYNA includes the mass from SURFB when estimating the cohesive element's time step. Note that groupable tied contacts are not currently supported with this option.
TIEOPT	Option for constrained tied contact formulations to circumvent some of the shortcomings present in the standard implementations: EQ.0: Not active EQ.1: Active. See Remark 11 . EQ.2: Same as 1 but avoids constraining rotations when possible. See Remark 11 .
STROBJ	Flag for strong objectivity (frame invariance) in non-groupable single surface contacts with SOFT = 0 or 1. Strong objectivity incurs an extra cost but may improve results. EQ.0: Do not turn on strong objectivity. EQ.1: Turn on strong objectivity.

Remarks:

1. **Shell thickness change.** The shell thickness change option (ISTUPD) must first be activated in *CONTROL_SHELL for shell thickness to change. Secondly, for a single surface contact to recognize the change in shell thickness, THKCHG must be set to 1. For surface-to-surface contacts with shell thickness offsets, that is, *CONTACT_SURFACE_TO_SURFACE with SHLTHK set to 1 or 2, or *CONTACT_AUTOMATIC_SURFACE_TO_SURFACE, the shell thickness change is recognized by default, and THKCHG is not used.
2. **Default contact thickness for single surface contacts.** For single-surface contact types,

SINGLE_SURFACE
AUTOMATIC_SINGLE_SURFACE
AUTOMATIC_GENERAL
AUTOMATIC_GENERAL_INTERIOR
ERODING_SINGLE_SURFACE,

the default contact thickness is taken as the smaller of two values, namely:

- shell thickness, and
- 40% of the minimum edge length.

The minimum edge length is calculated as the minimum of the distances from N4 to N1, N1 to N2, and N2 to N3. N3 to N4 is neglected because the shell is possibly triangular. For shells that have a large thickness, the “40% of the minimum edge length” criterion may control the contact thickness. This edge length criterion is an ad hoc criterion included to help prevent solution instability or add a slow down in the contact searching in the case where the shell thickness-to-edge-length ratio is large.

If SSTHK = 1 or if SOFT = 2 (Optional Card A in *CONTACT), the edge length criterion is not considered, that is, the default contact thickness is simply the shell thickness.

3. **Initial penetration check.** As of version 950 the initial penetration check option is always performed regardless of the value of ISLCHK. If you do not want to remove initial penetrations, then set the contact birth time (see *CONTACT_...) so that the contact is not active at time 0.0.
4. **Automatic reorientation.** Automatic reorientation requires offsets between the segments on each side of the contact interface. The reorientation is based on segment connectivity, and once all segments are oriented consistently based on connectivity, a check is made to see if the surfaces on each side of the contact interface face each other based on the right-hand rule. If not, all segments in a given surface are reoriented. This procedure works well for non-disjoint surfaces. If the surfaces are disjoint, the AUTOMATIC contact options, which do not require orientation, are recommended. For FORMING contact types, automatic reorientation works for disjoint surfaces.
5. **Neighbor segment thinning option.** Setting SPOTHIN and SWRADF greater than zero activates a neighbor segment thinning option. The radius of a spot weld is scaled by SWRADF, and then a search is made for shell segments that are neighbors of the tied shell segments that are touched by the weld but not tied by it. The SWRADF option is available for beam element welds, solid element

welds, and solid weld assemblies. For solids and solid assemblies, the weld radius is calculated assuming a circular shape with an equivalent area.

6. **Segment masses for penalty stiffness.** Segment-to-segment contact (see SOFT = 2 on *CONTACT) calculates a penalty stiffness based on the solution time step and the masses of the segments in contact. By default, segment masses are calculated using the material density of the element associated with the segment and the volume of the segment. This method does not take into account added mass introduced by lumped masses or mass scaling and can lead to stiffness that is too low. Therefore, a second method (PSTIFF = 1) was added which estimates the segment mass using the nodal masses. Note that for PSTIFF = 1, the algorithm compares the mass calculated by the two methods and uses the larger value. Setting a PSTIFF value here sets the default values of PSTIFF for all interfaces. The PSTIFF option can also be specified for individual contact interfaces by defining PSTIFF on Optional Card F of *CONTACT.
7. **Force transducer search option.** Force transducers between two surfaces measure the contact force from any contact interfaces that generate force between the SURFA and SURFB surfaces of the force transducer. When contact is detected, a search is made to see if the contact force should be added to any two surface force transducers. By default, when a force transducer match is found, the force is added, and the search terminates. When FTALL = 1, the search continues to check for other two surface force transducer matches. This option is useful when the SURFA and SURFB force transducer surfaces overlap. If there is no overlap, the default is recommended.
8. **Symmetry option for Mortar contact.** By specifying a node set through ISYM < 0, a segment will be ignored in the Mortar contact if *all* nodes in the segment are contained in the specified node set. Edge treatment will also be avoided on the interface between the symmetry plane and the true contact surface. *Important note:* When using this option, it is important that segments *close* to the symmetry plane but *not in it* have at least one node on the true contact surface. For instance, if a quarter symmetry model is used, avoid using a wedge element at the center as this is likely to render a triangular contact segment with all of its nodes in the symmetry node set.
9. **MORTAR contact fields.** The following table lists parameters that in some sense are related to MORTAR contact. Any parameter that is not mentioned in this list is ignored. See also a similar table on *CONTACT.

Data Card	Comment
Card 1	Only SLSFAC applies, and it applies as for any other contact. Other parameters are ignored. Shell thickness offsets are always included, except for SURFB surface thickness with the FORMING option. For the FORMING option the normal of the SURFB surface must point towards the SURFA side of the contact. For any contact, shell thickness changes are considered if ISTUPD on *CONTROL_SHELL is used. Offset of the shell reference surface is considered if NLOC on *SECTION_SHELL is used, regardless of the value of CNTCO on *CONTROL_SHELL.
Card 2	USRFRC and NSBCS apply as for any other contact; all other parameters are ignored.
Card 3	All parameters apply as for any other contact.
Card 4	IGNORE = 0/1/2 applies as described on *CONTACT, as does FRCENG. Other parameters are ignored.
Card 5	ISYM < 0 applies, which indicates a node set on symmetry planes. Other parameters are ignored.
Card 6	No parameters apply. Edges in automatic contact are always flat.

10. **RWPNAL and IGA.** For IGA parts specified with either *IGA_SHELL or *IGA_SOLID, the rigid wall is enforced through the interpolation nodes generated for the IGA parts. The interpolation nodes are treated like nodal points that belong to rigid bodies and are treated with the penalty method. The actual control points for the IGA parts are removed from the list of possible tracked nodes.

If RWPNAL = 0.0, rigid nodes are not included in the contact, but the interpolation nodes are still enforced by the penalty method using a rigid wall penalty scale factor.

11. **TIEOPT.** When invoking this option, the following happens for any of the constrained tied contacts in the model:

- a) SMP and MPP tied contacts become identical, and each finds the same node-segment pairs.
- b) The segment sides of different tied contact interfaces are allowed to share nodes, which otherwise would require the GROUPABLE option.
- c) The SHELL_EDGE_TO_SOLID_CONSTRAINED_OFFSET option becomes available, which otherwise is silently converted to a SHELL_EDGE_TO_SOLID without offset.

- d) The tied contacts automatically detect the nature of involved nodes and convert any invalid constraints involving rotational degrees of freedom to suitable alternatives, so “any” tied option is valid for “any” situation.
- e) This option is mandatory for selective mass scaling, meaning automatically invoked.
- f) For TIEOPT = 2, any pair using the SHELL_EDGE_TO_SURFACE option is converted to a SHELL_EDGE_TO_SOLID option to avoid the coupling between rotational degrees of freedom on the segment side to translational degrees of freedom on the node side. This conversion potentially avoids instabilities related to this option.

CONTROL_COUPLING**CONTROL*****CONTROL_COUPLING**

Purpose: Change defaults for MADYMO3D/CAL3D coupling; see Appendix I.

Card 1	1	2	3	4	5	6	7	8
Variable	UNLENG	UNTIME	UNFORC	TIMIDL	FLIPX	FLIPY	FLIPZ	SUBCYL
Type	F	F	F	F	I	I	I	I
Default	1.	1.	1.	0.	0	0	0	1

VARIABLE**DESCRIPTION**

UNLENG Unit conversion factor for length. MADYMO3D/GM-CAL3D lengths are multiplied by UNLENG to obtain LS-DYNA lengths.

UNTIME Unit conversion factor for time, UNTIME. MADYMO3D/GM-CAL3D time is multiplied by UTIME to obtain LS-DYNA time.

UNFORC Unit conversion factor for force, UNFORC. MADYMO3D/GM-CAL3D force is multiplied by UNFORC to obtain LS-DYNA force.

TIMIDL Idle time during which CAL3D or MADYMO is computing and LS-DYNA remains inactive. Important for saving computer time.

FLIPX Flag for flipping X-coordinate of CAL3D/MADYMO3D relative to the LS-DYNA model:

EQ.0: off,

EQ.1: on.

FLIPY Flag for flipping Y-coordinate of CAL3D/MADYMO3D relative to the LS-DYNA model:

EQ.0: off,

EQ.1: on.

FLIPZ Flag for flipping Z-coordinate of CAL3D/MADYMO3D relative to the LS-DYNA model:

EQ.0: off,

EQ.1: on.

***CONTROL**

***CONTROL_COUPLING**

VARIABLE	DESCRIPTION
SUBCYL	CAL3D/MADYMO3D subcycling interval (# of cycles): EQ.0: Set to 1, GT.0: SUBCYL must be an integer equal to the number of LS-DYNA time steps between each CAL3D/MADYMO3D step. Then the position of the contacting rigid bodies is assumed to be constant for n LS-DYNA time steps. This may result in some increase in the spikes in contact, thus this option should be used carefully. As the CAL3D/MADYMO3D programs usually work with a very small number of degrees of freedom, not much gain in efficiency can be achieved.

CONTROL_CPG**CONTROL*****CONTROL_CPG**

Purpose: Global control parameters for CPG (Continuum-based Gas Particle method).

Card 1	1	2	3	4	5	6	7	8
Variable		NCPC		VERB		NSLIP		ICORR
Type		I		I		I		I
Default		5		0		0		1

VARIABLE	DESCRIPTION
NCPC	Number of cycles between point cloud checks. A negative value points to a load curve ID giving the number of cycles as a function of time. See Remark 1 .
VERB	CPG verbosity control: EQ.0: No CPG screen/stdout output EQ.1: CPG output displayed on screen/stdout EQ.2: Same as 1, except the information is also output in the mes0000 file
NSLIP	Slip condition flag: EQ.0: Free slip condition applied at walls EQ.1: Non-slip condition applied at wall
ICORR	Correction of density and total energy: EQ.1: Thermodynamic balance correction on (default). See Remark 2 . EQ.2: Correction turned off.

Remarks:

1. **Point cloud check.** CPG relies on a point cloud sampling of the entire volume. As the airbag deforms, the point cloud needs to be updated with the creation and deletion of CPG particles and the redistribution of particles across processors. If no deformation occurs throughout the analysis, this resampling is not

needed and a very high value of NCPC can be set to save some calculation time. For typical airbag applications, the default value is recommended.

2. **ICORR.** At any given time, the solver calculates the total volume of the domain, the total mass of gas inside the volume, and the total energy. From these quantities, an expected average pressure can be calculated and used to correct the particle pressure field for a better thermodynamic balance.

CONTROL_CPM**CONTROL*****CONTROL_CPM**

Purpose: Global control parameters for CPM (Corpuscular Particle Method).

Card 1	1	2	3	4	5	6	7	8
Variable	CPMOUT	NP2P	NCPMTS	CPMERR	SFFDC	BLKV	CPMMF	P2PMIX
Type	I	I	I	I	F	I	I	I
Default	11	5	0	0	1.0	0	0	0

This card is optional.

Card 2	1	2	3	4	5	6	7	8
Variable	PMIS							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
CPMOUT	Control CPM output database to the d3plot files (see Remark 1): EQ.11: Full CPM database in version 3 format (default) EQ.21: Full CPM database in version 4 format EQ.22: CPM coordinates only in version 4 format EQ.23: CPM summary only in version 4 format
NP2P	Number of cycles for repartition particle among processors. This option is only used in LS-DYNA/MPP. (Default = 5)
NCPMTS	Time step size estimation: EQ.0: Do not consider CPM (default) EQ.1: Use 1 microsecond as CPM time step size. This provides a better time step size if the model is made up of rigid bodies.

VARIABLE	DESCRIPTION
CPMERR	<p>EQ.0: Disable checking and only output warning messages (Default)</p> <p>EQ.1: Enable error checking. If LS-DYNA detects any problem, it will either error terminate the job or try to fix the problem. Activated checks include:</p> <ol style="list-style-type: none">1. <i>Airbag</i> integrity (see Remark 2)2. <i>Chamber</i> integrity: this step applies the airbag integrity check to the chamber.3. Inconsistent orientation between the shell reference geometry and FEM shell connectivity.
SFFDC	Scale factor for the force decay constant. The default value is 1.0 and allowable range is [0.01,100.0]. See Remark 3 .
BLKV	Allocate additional memory for contact nodal forces, excluding force transducers and airbag single surface contacts using SOFT = 2. These nodal forces will be used to estimate blockage of external vents. BLKV does not affect the porosity and blockage of internal vents.
CPMMF	Flag to consider airbag system velocity based on the coordinates system defined by fields NID1, NID2, and NID3 on *AIRBAG_PARTICLE: <p>EQ.0: No (default)</p> <p>EQ.1: Yes. The flow energy from the rigid body motion is fed back to the CPM particles.</p>
P2PMIX	Control the energy transfer during particle-to-particle collision and change the thermalization of the particles (see Remark 4): <p>EQ.0: Thermalization considered in all particle-to-particle collisions (default).</p> <p>EQ.1: Thermalization only considered within same gas species</p> <p>EQ.2: Same as 0 but treat temperature dependent $\xi(T)$</p> <p>EQ.3: Same as 1 but treat temperature dependent $\xi(T)$</p>
PMIS	Flag for choosing logic to use when a particle leaks out due to undetected contact (see Remark 5):

VARIABLE	DESCRIPTION
	EQ.0: Return particle to around inflator orifice node (default)
	EQ.1: Return particle to the nearest airbag fabric

Remarks:

1. **D3PLOT Version.** “Version 3” is an older format than “Version 4”. Version 4 stores data more efficiently than version 3 and has options for what data is stored but may not be readable by old LS-PrePost executables.
2. **Airbag Integrity Checking.** The bag’s volume is used to evaluate all bag state variables. If the volume is ill-defined or inaccurate, then the calculation will fail. Therefore, it is vital that the volume be closed, and that all shell normal vectors point in the same direction.

When CPMERR = 1, the calculation will error terminate if either the bag’s volume is not closed or if one of its parts is not internally oriented (meaning that it contains elements that are not consistently oriented). Once it is verified that each part has a well-defined orientation, an additional check is performed to verify that all of bag’s constituent parts are consistently oriented with respect to each other. If they are not, then the part orientations are flipped until the bag is consistently oriented with an *inward* pointing normal vector.

3. **Force Decay Constant.** Particle impact force is gradually applied to airbag segment by a special smoothing function with the following form.

$$F_{\text{apply}} = \left[1 - \exp \left(\frac{-dt}{SFFDC \times \tau} \right) \right] (F_{\text{current}} + F_{\text{stored}}) ,$$

where τ is the force decay constant stored in LS-DYNA.

4. **P2PMIX.** P2PMIX sets the energy transfer scheme during particle-to-particle collision.

Based on the molecular kinetic theory, a fraction, ξ , of the total internal energy, e , is translational kinetic energy, w_k , for the CPM particle performing PV work:

$$p = \frac{2}{3} w_k = \frac{2}{3} \xi(T) e$$

This pressure should be the same as the pressure given by the Ideal Gas Law:

$$p = (\gamma - 1)e$$

In the above, ξ is a weak function of temperature like γ .

Each CPM particle represents a cloud of molecules, and CPM assumes those molecules are under thermal equilibrium with a proper Maxwell-Boltzmann distribution. When two particles collide, they will follow a perfectly elastic collision and the amount of energy transferred between them will keep both particles under the new Maxwell-Boltzmann distribution. By default, CPM assumes constant ξ during the process for better speed.

Since each gas species has its own Maxwell-Boltzmann distribution, the above calculation should only be applied to the same gas species. However, by default (P2PMIX = 0), thermalization is considered for all interactions. When P2PMIX equals 1, CPM particles collide perfectly elastically, and simple energy conservation is enforced when particles of different species interact. In this case the above calculation is only applied to particles of the same gas species.

5. **PMIS.** A particle may leak from a particle-to-surface contact due to numerical error. Once this particle is detected, it will be relocated to the gas inflator orifice node by default. If the bag is modeled with a multiple chamber definition, the default scheme may put the particle into a different chamber and, thus, unintentionally change the chamber pressure. If PMIS equals 1, this particle will be returned to the closest fabric segment.

***CONTROL_CPU**

Purpose: Control CPU time.

Card 1	1	2	3	4	5	6	7	8
Variable Type	CPUTIM	IGLST						

VARIABLE	DESCRIPTION
CPUTIM	Seconds of CPU time: EQ.0.0: No CPU time limit set GT.0.0: Time limit for cumulative CPU of the entire simulation, including all restarts. LT.0.0: Absolute value is the CPU time limit in seconds for the first run and for each subsequent restart.
IGLST	Flag for outputting CPU and elapsed times in the glstat file: EQ.0: No EQ.1: Yes

Remarks:

The CPU limit is not checked until after the initialization stage of the calculation. Upon reaching the CPU limit, the code will output a restart dump file and terminate. The CPU limit can also be specified on the LS-DYNA execution line via "c=". The value specified on the execution line will override CPUTIM specified in this keyword.

***CONTROL_DEBUG**

Purpose: Write supplemental information to the messag file(s). One effect of this command is that the sequence of subroutines called during initialization and memory allocation is printed. Aside from that, the extra information printed pertains only to a select few features, including:

1. Spot weld connections which use *MAT_100_DA and *DEFINE_CONNECTION_PROPERTIES.
2. The GISSMO damage model invoked using *MAT_ADD_EROSION. (Supplemental information about failed elements is written.)

CONTROL_DISCRETE_ELEMENT**CONTROL*****CONTROL_DISCRETE_ELEMENT**

Purpose: Define global control parameters for discrete element spheres.

Card Summary:

Card 1. This card is required.

NDAMP	TDAMP	FRICS	FRICR	NORMK	SHEARK	CAP	VTK
-------	-------	-------	-------	-------	--------	-----	-----

Card 2. Additional card for $CAP \neq 0$. If $CAP = 0$, include this card as a blank line if optional Card 3 is included.

GAMMA	VOL	ANG	GAP		IGNORE	NBUF	PARALLEL
-------	-----	-----	-----	--	--------	------	----------

Card 3. Card 3 and beyond are optional. If any of the following cards are included, then the preceding optional cards must be included, but they can be blank lines.

LNORM	LSHEAR		FRICD	DC	NCRB	BT	DT
-------	--------	--	-------	----	------	----	----

Card 4. This card is optional.

CP	TC	TFAC					
----	----	------	--	--	--	--	--

Card 5. This card is optional.

IDESOF	SOFSC		ISKIP	MAXNEI			
--------	-------	--	-------	--------	--	--	--

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	NDAMP	TDAMP	FRICS	FRICR	NORMK	SHEARK	CAP	VTK
Type	F	F	F	F	F	F	I	I
Default	0.	0.	0.	0.	0.01	2/7	0	0

VARIABLE	DESCRIPTION
NDAMP	Normal damping coefficient
TDAMP	Tangential damping coefficient

*CONTROL

*CONTROL_DISCRETE_ELEMENT

VARIABLE	DESCRIPTION
FRICS	Static coefficient of friction (see Remark 4): EQ.0: 3 DOF NE.0: 6 DOF (consider rotational DOF)
FRICR	Rolling friction coefficient
NORMK	Optional: scale factor of normal spring constant. Norm contact stiffness is calculated as $K_n = \begin{cases} \frac{k_1 r_1 k_2 r_2}{k_1 r_1 + k_2 r_2} NORMK & \text{if } NORMK > 0 \\ NORMK & \text{if } NORMK < 0 \end{cases}$. NORMK is ignored if LNORM ≠ 0 or IDESOFT = 1.
SHEARK	Optional: ratio between SHEARK/NORMK. Tangential stiffness is calculated as $K_t = SHEARK \times K_n$. SHEARK is ignored if LS-HEAR ≠ 0.
CAP	Capillary force flag: EQ.0: Dry particles NE.0: Wet particles. Capillary force is considered, and an additional input card is needed. See Remark 1 .
VTK	Output DES in VTK format for ParaView: EQ.0: No EQ.1: Yes

Capillary Card. Additional card for CAP ≠ 0. If CAP = 0, include as a blank line if optional Card 3 is included.

Card 2	1	2	3	4	5	6	7	8
Variable	GAMMA	VOL	ANG	GAP		IGNORE	NBUF	PARALLEL
Type	F	F	F	F		I	I	I
Default	0.	0.	0.	0.		0	6	0

CONTROL_DISCRETE_ELEMENT**CONTROL**

VARIABLE	DESCRIPTION
GAMMA	Liquid surface tension, γ
VOL	Volume fraction
ANG	Contact angle, θ
GAP	Optional parameter affecting the spatial limit of the liquid bridge. A liquid bridge exists when δ , as illustrated in Figure 12-15 , is less than or equal to $\min(\text{GAP}, d_{\text{rup}})$ where d_{rup} is the rupture distance of the bridge automatically calculated by LS-DYNA.
IGNORE	Ignore initial penetration option: EQ.0: Calculate the contact force for DES with initial penetration. GT.0: Ignore the contact force calculation for DES with initial penetration.
NBUF	GE.0: Factor of memory use for asynchronous message buffer (Default = 6). LT.0: Disable asynchronous scheme and use minimum memory for data transfer.
PARALLEL	Flag for calculating contact force between bonded DES: EQ.0: Skip contact force calculation for bonded DES (Default) EQ.1: Consider contact force calculation for bonded DES

Card 3 is optional. If included, then Card 2 must be defined, even as a blank card.

Card 3	1	2	3	4	5	6	7	8
Variable	LNORM	LSHEAR		FRICD	DC	NCRB	BT	DT
Type	I	I		F	F	I	F	F
Default	0	0		FRICS	0	0	0.	10^{20}

*CONTROL

*CONTROL_DISCRETE_ELEMENT

VARIABLE	DESCRIPTION
LNORM	Load curve ID of a curve that defines a function for normal stiffness with respect to norm penetration ratio. See Remark 2 .
LSHEAR	Load curve ID of a curve that defines a function for shear stiffness with respect to norm penetration ratio. See Remark 3 .
FRICD	Dynamic coefficient of friction. See Remark 4 .
DC	Exponential decay coefficient. See Remark 4 .
NCRB	Rebalancing frequency, that is, the number of cycles between each rebalancing. This parameter only applies to MPP. EQ.0: No rebalancing is performed (default).
BT	Birth time
DT	Death time

Thermal Properties Card. Card 4 is optional.

Card 4	1	2	3	4	5	6	7	8
Variable	CP	TC	TFAC					
Type	F	F	F					
Default	0.0	0.0	0.0					

VARIABLE	DESCRIPTION
CP, TC, TFAC	DES thermal properties (<i>Under development</i>)

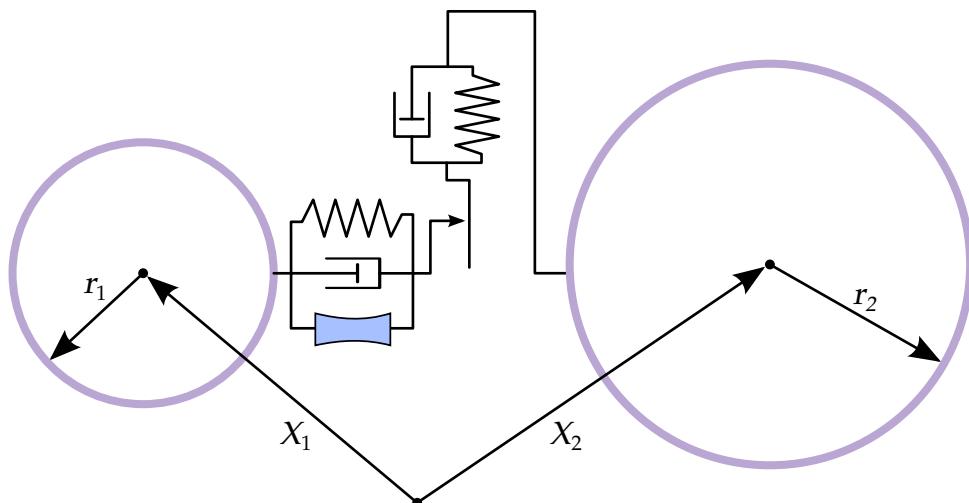


Figure 12-14. Schematic representation of sphere-sphere interaction

Card 5 is optional.

Card 5	1	2	3	4	5	6	7	8
Variable	IDESOFT	SOFSC		ISKIP	MAXNEI			
Type	I	F		I	I			
Default	0	0.1		0	20			

VARIABLE	DESCRIPTION
IDESOFT	Flag for soft constraint formulation: EQ.1: Soft constraint formulation. The contact stiffness is based on the nodal mass and the global time step size. This input provides a different way for calculating NORMK. NORMK is ignored if IDESOFT = 1. IDESOFT is ignored if LNORM ≠ 0.
SOFSC	Scale factor applied to the contact stiffness in the soft constraint formulation
ISKIP	Flag for skipping the calculation of contact force between DES: EQ.0: Consider the particle-particle contact calculation (default). EQ.1: Skip the particle-particle contact calculation.

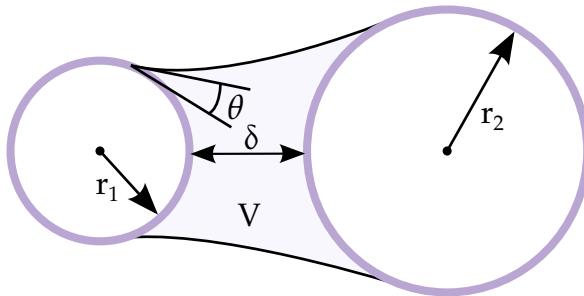


Figure 12-15. Schematic representation of capillary force model.

VARIABLE	DESCRIPTION
MAXNEI	Number of neighbors to be tracked for DES contact and capillary force calculation (default = 20). If particle sizes are very different, MAXNEI needs to be increased to capture more neighbors.

Background:

This method models all parts as being comprised of rigid spheres. These spheres interact with both conventional solids and other spheres. Sphere-sphere interactions are modeled in contact points using springs and dampers as illustrated in [Figure 12-14](#). [Cundall & Strack 1979]

Remarks:

1. **Capillary Forces to Model Cohesion.** This extension is enabled using the CAP field. Capillary force between wet particles is based on the following reference: "Capillary Forces between Two Spheres with a Fixed Volume Liquid Bridges: Theory and Experiment", Yakov I. Rabinovich et al. Langmuir 2005, 21, 10992-10997. See [Figure 12-15](#).

The capillary force is given by

$$F = -\frac{2\pi R\gamma \cos \theta}{1 + \frac{\delta}{2d}} ,$$

where

$$d = \frac{\delta}{2} \left(-1 + \sqrt{1 + \frac{2V}{\pi R \delta^2}} \right) ,$$

and

$$R = \frac{2r_1 r_2}{r_1 + r_2} .$$

2. **User Defined Norm Stiffness.** Let $y = f(x)$ be a load curve specified with LNORM for the norm stiffness between two interacting discrete element spheres. x is the relative penetration, meaning $x = \delta/\min(r_1, r_2)$. δ is penetration. The normal spring force is calculated as

$$F_n = k_{\text{eff}} \times y \times (\min(r_1, r_2))^2,$$

where k_{eff} is the effective bulk modulus of the two interacting DEM particles, that is, $k_{\text{eff}} = k_1 k_2 / (k_1 + k_2)$. If the curve is defined as $y = cx$, the behavior is the same as NORMK = c .

3. **User Defined Shear Stiffness.** Let $y = f(x)$ be a load curve specified with LS-HEAR for the shear stiffness between two interacting discrete element spheres. x is the relative penetration, meaning $x = \delta/\min(r_1, r_2)$. δ is penetration. The tangential stiffness is calculated as $K_t = y \times K_n$, where K_n is the norm stiffness defined by NORMK or a user defined curve. If a curve is defined as $y = c$, the behavior is the same as SHEARK = c .
4. **Coefficient of Friction.** The frictional coefficient is assumed to be dependent on the relative velocity v_{rel} of the two DEM in contact, that is,

$$\mu_c = \text{FRICD} + (\text{FRICS} - \text{FRICD}) e^{-\text{DC} \times |v_{\text{rel}}|}.$$

*CONTROL

*CONTROL_DYNAMIC_RELAXATION

*CONTROL_DYNAMIC_RELAXATION

Purpose: Initialize stresses and deformation in a model to simulate a preload. Examples of preload include load due to gravity, load due to a constant angular velocity, and load due to torquing of a bolt. After the preloaded state is achieved by one of the methods described below (see field IDRFLG), the time resets to zero and the normal phase of the solution automatically begins from the preloaded state.

Card Summary:

Card 1. This card is required.

NRCYCK	DRTOL	DRFCTR	DRTERM	TSSFDR	IRELAL	EDTTL	IDRFLG
--------	-------	--------	--------	--------	--------	-------	--------

Card 2a. This card is included if IDRFLG = -3, 3, or 6.

DRPSET							
--------	--	--	--	--	--	--	--

Card 2b. This card is included if IDRFLG = 2.

NC	NP						
----	----	--	--	--	--	--	--

Card 2b.1. Include NP of this card.

PSID	VECID						
------	-------	--	--	--	--	--	--

Data Cards:

Card 1	1	2	3	4	5	6	7	8
Variable	NRCYCK	DRTOL	DRFCTR	DRTERM	TSSFDR	IRELAL	EDTTL	IDRFLG
Type	I	F	F	F	F	I	F	I
Default	250	0.001	0.995	infinity	TSSFAC	0	0.04	0

VARIABLE	DESCRIPTION
NRCYCK	Number of time steps between convergence checks for explicit dynamic relaxation.
DRTOL	Convergence tolerance for explicit dynamic relaxation (default = 0.001). See Remark 1 .

VARIABLE	DESCRIPTION
DRFCTR	Dynamic relaxation factor for transient dynamic relaxation (default = .995). See Remark 1 .
DRTERM	Optional termination time for dynamic relaxation. Termination occurs at this time or when convergence is attained (default = infinity). See Remarks 1 and 6 .
TSSFDR	Scale factor for computed time step during explicit dynamic relaxation. If zero, the value is set to TSSFAC defined on *CONTROL_TIMESTEP. After converging, the scale factor is reset to TSSFAC.
IRELAL	Automatic control for dynamic relaxation option based on algorithm of Papadrakakis [1981]: EQ.0: Not active EQ.1: Active
EDTTL	Convergence tolerance on automatic control of dynamic relaxation.
IDRFLG	Dynamic relaxation flag which controls how the preloaded state is computed: EQ.-999: Dynamic relaxation not activated even if specified on a load curve, see *DEFINE_CURVE. See Remark 3 . 2 . EQ.-3: Dynamic relaxation is activated as with IDRFLG = 1, but the convergence check is made based only on the part set specified by DRPSET. All parts are active during the dynamic relaxation phase. EQ.-1: Dynamic relaxation is activated, and time history output is produced during dynamic relaxation; see Remarks 1 and 2 . EQ.0: Not active. See Remark 3 . EQ.1: Dynamic relaxation is activated. See Remark 1 . EQ.2: Initialization to a prescribed geometry, see Remark 4 , EQ.3: Dynamic relaxation is activated as with IDRFLG = 1, but only for the part set specified by DRPSET; see Remark 5 . EQ.5: Initialize implicitly; see Remark 6 . EQ.6: Initialize implicitly but only for the part set specified by DRPSET; see Remark 6 .

CONTROL**CONTROL_DYNAMIC_RELAXATION**

Part Set Card. Additional card for IDRFLG = -3, 3 or 6.

Card 2a	1	2	3	4	5	6	7	8
Variable	DRPSET							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
DRPSET	Part set ID for IDRFLG = -3, 3 or 6.

Optional Card for IDRFLG = 2. This card is included if and only if IDRFLG = 2.

Card 2b	1	2	3	4	5	6	7	8
Variable	NC	NP						
Type	I	I						
Default	100	0						

VARIABLE	DESCRIPTION
NC	Number of time steps for initializing geometry when IDRFLG = 2. This variable applies to IDRFLG = 2 whether polar initialization is invoked or not.
NP	Number of cards defining parts to be initialized using the polar initialization approach and the associated polar coordinate system(s).

Optional Polar Initialization Cards. When IDRFLG = 2 and NP > 0, include NP additional cards which define the parts and polar coordinate systems used in polar initialization. This polar initialization interpolates from the initial to final state using a polar coordinate system. The objective is to avoid creating spurious stresses in parts subjected to large rigid body rotations and this option should not be used if that is not the case.

Card 2b.1	1	2	3	4	5	6	7	8
Variable	PSID	VECID						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
PSID	Part set ID
VECID	Vector ID which defines the origin and axis of rotation of the polar coordinate system used in initializing parts of part set PSID. See Remark 7 .

Remarks:

1. **Transient Dynamic Relaxation.** If IDRFLG is 1 or -1, a transient “dynamic relaxation” analysis is invoked in which an explicit analysis, damped by means of scaling nodal velocities by the factor DRFCTR each time step, is performed. When the ratio of current distortional kinetic energy to peak distortional kinetic energy (the convergence factor) falls below the convergence tolerance (DRTOL) or when the time reaches DRTERM, the dynamic relaxation analysis stops, and the current state becomes the initial state of the subsequent normal analysis.

Distortional kinetic energy is defined as total kinetic energy less the kinetic energy due to rigid body motion. A history of the distortional kinetic energy computed during the dynamic relaxation phase is automatically written to a file called `relax`. This file can be read as an ASCII file by LS-PrePost and its data plotted. The `relax` file also includes a history of the convergence factor.

2. **Dynamic Relaxation with Time History Data.** If IDRFLG is set to -1, the dynamic relaxation proceeds as normal, but time history data is written to the `d3thdt` file in addition to the normal data being written to the `d3drif` file. At the end of dynamic relaxation, the problem time is reset to zero. However,

information is written to the d3thdt file with an increment to the time value. The time increment used is reported at the end of dynamic relaxation.

3. **Dynamic Relaxation from Load Curves.** Dynamic relaxation will be invoked if SIDR is set to 1 or 2 in any of the *DEFINE_CURVE commands, even if IDRFLG = 0 in *CONTROL_DYNAMIC_RELAXATION. Curves so tagged are applicable to the dynamic relaxation analysis phase. Curves with SIDR set to 0 or 2 are applicable to the normal phase of the solution. Dynamic relaxation will always be skipped if IDRFLAG is set to -999.
4. **Restarts and Prescribed Geometry File.** Following the dynamic relaxation phase and before the start of the normal solution phase, a binary dump file (d3dump01) and a “prescribed geometry” file (drdisp.sif) are written by LS-DYNA. Either of these files can be used in a subsequent analysis to quickly initialize to the preloaded state without having to repeat the dynamic relaxation run. The binary dump file is used with a restart analysis (see RESTART_INPUT_DATA). The drdisp.sif file is used by setting IDRFLG = 2. The IDRFLG = 2 approach does not initialize the location of rigid bodies in the model.

When IDRFLG = 2, an ASCII file specified by "m=" on the LS-DYNA execution line is read which describes the initialized state of deformable bodies. The ASCII file contains each node ID with prescribed values of nodal displacement (x, y, z), nodal rotation (x, y, z), and nodal temperature in (I8,7E15.0) format. The pre-loaded state is quickly reached by linearly ramping nodal displacements, rotations, and temperatures of deformable bodies to prescribed values over a number of time steps as indicated by the variable NC.

5. **Part Set for Convergence Checking.** When IDRFLG = 3, a dynamic relaxation is performed only for the part set specified by DRPSET. For example, if only the tires are being inflated on a vehicle, it may be sufficient in some cases to apply the dynamic relaxation on the part IDs in the tire and possibly the suspension system.
6. **Implicit Preload.** If IDRFLG is set to 5 or 6, LS-DYNA performs an implicit analysis to obtain the preloaded state. Parameters for controlling the implicit preload solution are defined using appropriate *CONTROL_IMPLICIT keywords to specify solver type, implicit time step, etc. For example, *CONTROL_IMPLICIT_GENERAL specifies the implicit step size. The implicit analysis is, by default, static but can be made transient via the *CONTROL_IMPLICIT_DYNAMICS command.

DRTERM indicates the termination “time” of the implicit preload analysis. When DRTERM is reached, the implicit preload phase terminates, and LS-DYNA begins the next phase of the analysis according to IMFLAG in *CONTROL_IMPLICIT_GENERAL. For example, if it is desired to run an implicit preload

phase and switch to the explicit solver for the subsequent transient phase, IDRFLG should be set to 5 and IMFLAG should be set to 0.

In contrast to IDRFLG = 5, IDRFLG = 6 performs an implicit analysis only for the part subset specified with DRPSET.

7. **Vector for Prescribed Geometry.** When the displacements for IDRFLG = 2 are associated with large rotations, the linear interpolation of the displacement field introduces spurious compression and tension into the part. If a part set is specified with a vector, the displacement is interpolated by using polar coordinates with the tail of the vector specifying the origin of the coordinate system and the direction specifying the normal to the polar coordinate plane. The run terminates with an error if VECID is not specified. If no large rotations are present, the optional cards may be omitted.
8. **Output Database.** To create a binary output database having the same format as a d3plot database that pertains to the dynamic relaxation analysis, use *DATABASE_BINARY_D3DRLF. The output interval is given by this command as an integer representing the number of convergence checks between output states. The frequency of the convergence checks is controlled by the parameter NRCYCK.

*CONTROL

*CONTROL_EFG

*CONTROL_EFG

Purpose: Define controls for the mesh-free computation.

Card 1	1	2	3	4	5	6	7	8
Variable	ISPLINE	IDILA	ININT					
Type	I	I	I					
Default	0	0	12					
Remarks			1					

Card 2	1	2	3	4	5	6	7	8
Variable	IMLM	ETOL	IDEB	HSORT	SSORT			
Type	I	F	I	I	I			
Default	0	10^{-4}	0	not used	0			

VARIABLE

DESCRIPTION

- ISPLINE Optional choice for the mesh-free kernel functions:
EQ.0: Cubic spline function (default)
EQ.1: Quadratic spline function
EQ.2: Cubic spline function with circular disk (see [Remark 2](#))
- IDILA Optional choice for the normalized dilation parameter:
EQ.0: Maximum distance based on the background element
EQ.1: Maximum distance based on surrounding nodes
- ININT Factor needed for the estimation of maximum workspace (MWS-PAC) that can be used during the initialization phase.

VARIABLE	DESCRIPTION
IMLM	Optional choice for the matrix operation, linear solving and memory usage (see Remark 3): EQ.1: Original BCSLIB-EXT solvers EQ.2: EFGPACK (recommended). When this option is used, IN-INT in Card 1 becomes redundant.
ETOL	Error tolerance for the IMLM option.
IDEB	Output internal debug message
HSORT	Not used
SSORT	Automatic sorting of background triangular shell elements to FEM #2 when EFG shell type 41 is used: EQ.0: no sorting EQ.1: full sorting

Remarks:

1. **Maximum Workspace.** The mesh-free computation requires calls to use BCSLIB-EXT solvers during the initialization phase. The maximum workspace (MWSPAC) that can be used during the call is calculated as

$$\text{MWSPAC} = \text{ININT}^3 \times \text{NUMNEFG},$$

where NUMNEFG is the total number of mesh-free nodes. ININT, which is the number of nodes that a node influences along each cardinal direction, defaults to 12. When the normalized dilation parameters (DX,DY,DZ) in *SECTION_SOLID_EFG are increased, ININT must likewise increase.

2. **Cubic Spline Function with Circular Disk.** When ISPLINE = 2 is used, the input of the normalized dilation parameters (DX,DY,DZ) for the kernel function in *SECTION_SOILD_EFG and SECTION_SHELL_EFG only requires the DX value.
3. **Solvers.** EFGPACK was added to automatically compute the required maximum workspace in the initialization phase and to improve efficiency in the matrix operations, linear solving, and memory usage. The original BCSLIB-EXT solver requires an explicit workspace (ININT) for the initialization.

*CONTROL

*CONTROL_ENERGY

*CONTROL_ENERGY

Purpose: Provide controls for energy dissipation options.

Card 1	1	2	3	4	5	6	7	8
Variable	HGEN	RWEN	SLNTEN	RYLEN	IRGEN	MATEN	DRLEN	DISEN
Type	I	I	I	I	I	I	I	I
Default	1	2	1	1	2	1	1	1

VARIABLE	DESCRIPTION
HGEN	Hourglass energy calculation option. This option requires significant additional storage and increases cost by ten percent: EQ.1: Hourglass energy is not computed (default). EQ.2: Hourglass energy is computed and included in the energy balance. The hourglass energies are reported in the ASCII files glstat and matsum, see *DATABASE_OPTION. For implicit, or if the DRCPSID is active on *CONTROL-SHELL, the drilling energy is included here.
RWEN	Rigidwall energy (a.k.a. stonewall energy) dissipation option: EQ.1: Energy dissipation is not computed. EQ.2: Energy dissipation is computed and included in the energy balance (default). The rigidwall energy dissipation is reported in the ASCII file glstat; see *DATABASE_OPTION.
SLNTEN	Sliding interface energy dissipation option (This parameter is always set to 2 if contact is active. The option SLNTEN = 1 is not available.): EQ.1: Energy dissipation is not computed. EQ.2: Energy dissipation is computed and included in the energy balance. The sliding interface energy is reported in ASCII files glstat and sleout; see *DATABASE_OPTION.
RYLEN	Rayleigh energy dissipation option (damping energy dissipation):

VARIABLE	DESCRIPTION
	EQ.1: Energy dissipation is not computed (default). EQ.2: Energy dissipation is computed and included in the energy balance. The damping energy is reported in ASCII file glstat and matsum; see *DATABASE_OPTION.
IRGEN	Initial reference geometry energy option (included in internal energy, resulting from *INITIAL_FOAM_REFERENCE_GEOMETRY): EQ.1: Initial reference geometry energy is not computed. EQ.2: Initial reference geometry energy is computed and included in the energy balance as part of the internal energy (default).
MATEN	Detailed material energies option. For a choice of material models (currently supported are 3, 4, 15, 19, 24, 63, 81, 82, 98, 104, 105, 106, 107, 123, 124, 188, 224, 225, 240, and 251 for shell and solid elements), internal energy is additionally split into elastic, plastic, and damage portions: EQ.1: Detailed material energies are not computed (default). EQ.2: Detailed material energies are computed and reported as <i>mat_energy_elastic</i> , <i>mat_energy_plastic</i> , and <i>mat_energy_damage</i> in the ASCII files glstat and matsum.
DRLEN	Drilling energy calculation option, for implicit and with use of DR-CPSID/DRCPRM on *CONTROL_SHELL: EQ.1: Drilling energy is not computed (default). EQ.2: Drilling energy is computed and included in the energy balance. The drilling energies are reported in the ASCII file glstat, see *DATABASE_OPTION.
DISEN	Dissipation energy calculation option, for implicit: EQ.1: Dissipated energy is not computed (default). EQ.2: Dissipated kinetic and internal energy is computed and included in the energy balance. The dissipation energies are reported in the ASCII file glstat, see *DATABASE_OPTION.

*CONTROL

*CONTROL_EOS_USER_LIBRARY

*CONTROL_EOS_USER_LIBRARY_{OPTION}

Available options include:

<BLANK>

ASCII

Purpose: Provide a path to a user-defined library called **seslib** and scaling factors to convert the units of the input deck into the units of the **seslib** data (g, cm, μ s, GPa).

The ASCII keyword option enables specifying the name of an ASCII file that contains data tables. The library **seslib** is generated during initialization. These tables should have the format of the Sesame ASCII File (LA-UR-19-24891).

Card 1	1	2	3	4	5	6	7	8
Variable				PATH				
Type				A80				
Default			Current working directory					

Optional Unit Conversion Card. Optional card for converting the units of the input deck into the units for **seslib** data.

Card 2	1	2	3	4	5	6	7	8
Variable	CONM	CONL	CONT	CONP				
Type	F	F	F	F				
Default	1.0	1.0	1.0	100.0				

Optional Plot Card. Optional card for outputting 5 keyword files (*.k files with *NODE and *ELEMENT_SHELL) during initialization to facilitate plotting 3D surface graphs based on seslib data in LS-PrePost. When data differ by several orders of magnitude, the scale factors on this card improve displaying the graphs in LS-PrePost, such as by preventing these surfaces from being stretched.

Card 3	1	2	3	4	5	6	7	8
Variable	SCLR	SCLT	SCLE	SCLP				
Type	F	F	F	F				
Default	1.0	1.0	1.0	1.0				

VARIABLE	DESCRIPTION
PATH	Path to the library seslib. The default path is the current working directory. For the ASCII keyword option, the path is replaced by the name of an ASCII file containing data tables. The filename should include the path if the file is not in the current directory.
CONM	Scaling factor for mass conversion from the input deck units to the library units
CONL	Scaling factor for length conversion from the input deck units to the library units
CONT	Scaling factor for time conversion from the input deck units to the library units
COMP	Scaling factor for pressure conversion from the input deck units to the library units
SCLR	Plot scaling factor for densities
SCLT	Plot scaling factor for temperatures
SCLE	Plot scaling factor for energies
SCLP	Plot scaling factor for pressures

***CONTROL_EXPLICIT_THERMAL**

The *CONTROL_EXPLICIT_THERMAL_SOLVER keyword activates an explicit finite volume code to solve heat transfer by conduction. Enthalpies and temperatures are element centered. The elements supported by the thermal solver are beams, shells, solids, and multi-material 3D ALE elements. The *CONTROL_EXPLICIT_THERMAL_PROPERTIES keyword defines the heat capacities and conductivities by parts. These 2 keywords are mandatory to properly run the solver. Other keywords can be used to set the initial and boundary conditions and control the outputs. They are all listed below in alphabetical order:

- *CONTROL_EXPLICIT_THERMAL_ALE_COUPLING
- *CONTROL_EXPLICIT_THERMAL_BOUNDARY
- *CONTROL_EXPLICIT_THERMAL_CONTACT
- *CONTROL_EXPLICIT_THERMAL_INITIAL
- *CONTROL_EXPLICIT_THERMAL_OUTPUT
- *CONTROL_EXPLICIT_THERMAL_PROPERTIES
- *CONTROL_EXPLICIT_THERMAL_SOLVER

***CONTROL_EXPLICIT_THERMAL_ALE_COUPLING**

Purpose: Define the shell and solid parts involved in an explicit finite volume thermal coupling with multi-material ALE groups. This keyword requires *CONSTRAINED_LAGRANGE_IN_SOLID with CTYPE = 4.

Card 1	1	2	3	4	5	6	7	8
Variable	PARTSET	MMGSET						
Type	I	I						
Default	none	none						

VARIABLE	DESCRIPTION
PARTSET	Part set ID (See *SET_PART)
MMGSET	Multi-material set ID (see *SET_MULTI-MATERIAL_GROUP_LIST)

*CONTROL

*CONTROL_EXPLICIT_THERMAL_BOUNDARY

*CONTROL_EXPLICIT_THERMAL_BOUNDARY

Purpose: Set temperature boundaries with segment sets for an explicit finite volume thermal analysis.

Card 1	1	2	3	4	5	6	7	8
Variable	SEGSET	LCID						
Type	I	F						
Default	none	none						

VARIABLE	DESCRIPTION
SEGSET	Segment set ID (See *SET_SEGMENT)
LCID	*DEFINE_CURVE ID defining the temperature as a function of time

Remarks:

1. **Boundary Elements.** The boundary temperatures are set at segment centers. If shells or beams have all their nodes in the segment set, these elements will be considered boundary elements; the temperatures at their centers will be controlled by the curve LCID.

***CONTROL_EXPLICIT_THERMAL_CONTACT**

Purpose: Define the beam, shell and solid parts involved in an explicit finite volume thermal contact.

Card 1	1	2	3	4	5	6	7	8
Variable	PARTSET	NCYCLE						
Type	I	F						
Default	none	1						

VARIABLE	DESCRIPTION
PARTSET	Part set ID (See *SET_PART)
NCYCLE	Number of cycle between checks of new contact

*CONTROL

*CONTROL_EXPLICIT_THERMAL_INITIAL

*CONTROL_EXPLICIT_THERMAL_INITIAL

Purpose: Initialize the temperature centered in beams, shells or solids involved in an explicit finite volume thermal analysis.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	IDTYP	TEMPINI					
Type	I	F	F					
Default	none	none	0.0					

VARIABLE	DESCRIPTION
ID	Flag to determine if ID is an element or set ID: GT.0: ID is a set LT.0: ID is an element
IDTYP	Type of ID: EQ.1: solid EQ.2: shell EQ.3: beam EQ.4: thick shell
TEMPINI	Initial temperature (see Remark 1)

Remarks:

1. **Material with *EOS.** The volumetric enthalpy is the sum of the pressure and volumetric internal energy (as defined in *EOS). If the material has an equation of state, the enthalpy should not be initialized by the temperature but by the initial volumetric internal energy and pressure set in *EOS.

CONTROL_EXPLICIT_THERMAL_OUTPUT**CONTROL*****CONTROL_EXPLICIT_THERMAL_OUTPUT**

Purpose: Output temperatures and enthalpies for an explicit finite volume thermal analysis. See [Remark 1](#).

Card 1	1	2	3	4	5	6	7	8
Variable	DTOUT	DTOUTYP	SETID	SETYP				
Type	F	I	I	I				
Default	none	0	0	0				

VARIABLE	DESCRIPTION
DTOUT	Time interval between outputs
DTOUTYP	Type of DTOUT: EQ.0: DTOUT is a constant EQ.1: DTOUT is the ID of *DEFINE_CURVE defining a table of time as function of DTOUT
SETID	Set ID. If SETID = 0, then temperatures and enthalpies are output for the whole model. See Remark 1 .
SETYP	Type of set: EQ.1: solid set (see *SET_SOLID) EQ.2: shell set (see *SET_SHELL) EQ.3: beam set (see *SET_BEAM) EQ.4: thick shell set (see *SET_TSHELL)

Remarks:

- Output Files.** The temperatures and enthalpies are calculated at the element center. If SETID is the default 0, then the temperature and enthalpy data is output in d3plot format files with the following basename: `xplcth_output`. If SETID is not 0, then the temperatures and enthalpies are calculated for a subset of elements defined as specified in the SETYP field. For this case the temperature and enthalpy histories are output by element in a .xy format. The file names are `temperature_{beam,shell,solid}ID.xy` and `enthalpy_{beam,shell,solid}ID.xy`.

2. **Viewing Results with LS-Prepost.** When SETID = 0, the output can be viewed with LS-Prepost using a fringe plot. To view the output for solid and shell elements select *FriComp* → *X-stress and Y-stress* (select *Mid* for the integration points in shells); for beams select *FriComp* → *beam* → *axial force resultant* and *s-force resultant*.

CONTROL_EXPLICIT_THERMAL_PROPERTIES**CONTROL*****CONTROL_EXPLICIT_THERMAL_PROPERTIES**

Purpose: Define the thermal properties of beam, shell and solid parts involved in an explicit finite volume thermal analysis.

Card 1	1	2	3	4	5	6	7	8
Variable	PARTSET	CP	CPTYP	VECID1	VECID2	LOCAL		
Type	I	F	I	I	I	I		
Default	none	none	0	0	0	0		

Card 2	1	2	3	4	5	6	7	8
Variable	Kxx	Kxy	Kxz	KxxTYP	KxyTYP	KxzTYP		
Type	F	F	F	I	I	I		
Default	0.0	0.0	0.0	0	0	0		

Card 3	1	2	3	4	5	6	7	8
Variable	Kyx	Kyy	Kyz	KyxTYP	KyyTYP	KyzTYP		
Type	F	F	F	I	I	I		
Default	0.0	0.0	0.0	0	0	0		

CONTROL**CONTROL_EXPLICIT_THERMAL_PROPERTIES**

Card 4	1	2	3	4	5	6	7	8
Variable	Kzx	Kzy	Kzz	KzxTYP	KzyTYP	KzzTYP		
Type	F	F	F	I	I	I		
Default	0.0	0.0	0.0	0	0	0		

VARIABLE	DESCRIPTION
PARTSET	Part set ID (See *SET_PART)
CP	Heat capacity
CPTYP	Type of CP: EQ.0: CP is a constant EQ.1: CP is the ID of a *DEFINE_CURVE defining a table of temperature as a function of heat capacity
VECID1, VECID2	*DEFINE_VECTOR IDs to define a specific coordinate system. VECID1 and VECID2 give the <i>x</i> - and <i>y</i> -direction, respectively. The <i>z</i> -vector is a cross product of VECID1 and VECID2. If VECID2 is not orthogonal to VECID1, its direction will be corrected with a cross-product of the <i>z</i> - and <i>x</i> -vectors. The conductivity matrix Kij is applied in this coordinate system.
LOCAL	Flag to activate an element coordinate system: EQ.0: The vectors VECIDj are considered in a global coordinate system. EQ.1: The vectors VECIDj are considered in a local system attached to the element. For shells and solids, the system is the same as DIREC = 1 and CTYPE = 12 in *CONSTRAINED_LAGRANGE_IN_SOLID. For shells, the edge centers replace the face centers. For beams, the <i>x</i> -direction is aligned with the first 2 nodes in *ELEMENT_BEAM and there should be a 3 rd node for the <i>y</i> -direction.
Kij	Heat conductivity matrix

VARIABLE	DESCRIPTION
KijTYP	Type of Kij: EQ.0: Kij is a constant EQ.1: Kij is the ID of a *DEFINE_CURVE defining a table of temperature as a function of heat conductivity

*CONTROL

*CONTROL_EXPLICIT_THERMAL_SOLVER

*CONTROL_EXPLICIT_THERMAL_SOLVER

Purpose: Define the beam, shell, and solid parts involved in a finite volume thermal analysis. The enthalpies and temperatures are explicitly updated in time.

Card 1	1	2	3	4	5	6	7	8
Variable	PARTSET	DTFAC						
Type	I	F						
Default	none	1.0						

VARIABLE	DESCRIPTION
PARTSET	Part set ID (See *SET_PART)
DTFAC	Time step factor (see Remark 1)

Remarks:

1. **Time Step.** The time step is the minimum of the mechanical and thermal time steps. The thermal time step is the minimum of the element thermal time steps, which are half the enthalpies divided by the right hand side of the heat equation (conductivity \times temperature Laplacian). The thermal time step is scaled by DTFAC.

***CONTROL_EXPLOSIVE_SHADOW_{OPTION}**

The available options are:

<BLANK>

SET

Purpose: Compute detonation times for explosive elements for which there is no direct line of sight. If this command is not included in the input, the lighting time for an explosive element is computed using the distance from the center of the element to the nearest detonation point, L_d ; the detonation velocity, D ; and the lighting time for the detonator, t_d :

$$t_L = t_d + \frac{L_d}{D}$$

The detonation velocity for this option is taken from the element whose lighting time is computed and does not account for the possibilities that the detonation wave may travel through other explosives with different detonation velocities or that the line of sight may pass outside of the explosive material.

If this command is present, the lighting time of each explosive element is based on the shortest path through the explosive material from the associated detonation point(s) to the explosive element. 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.

The SET option requires input of a set ID of two-dimensional shell elements or three-dimensional solid elements for which explosive shadowing is active. If the SET option is not used, Card 1 should be omitted and shadowing is active for all explosive elements.

See also *INITIAL_DETONATION and *MAT_HIGH_EXPLOSIVE.

Card 1. Card for SET keyword option.

Card 1	1	2	3	4	5	6	7	8
Variable	SETID							
Type	I							
Default	none							

***CONTROL**

***CONTROL_EXPLOSIVE_SHADOW**

VARIABLE	DESCRIPTION
SETID	Set ID of a *SET_SHELL or *SET_SOLID. If the SET option is active, the lighting times are computed for a set of shells (*SET_SHELL in two dimensions) or solids (*SET_SOLID in three dimensions).

***CONTROL_FORMING**

Purpose: Set parameters for metal forming related features.

- *CONTROL_FORMING_AUTO_NET
- *CONTROL_FORMING_AUTOCHECK
- *CONTROL_FORMING_AUTOPOSITION_PARAMETER
- *CONTROL_FORMING_BESTFIT
- *CONTROL_FORMING_HOME_GAP
- *CONTROL_FORMING_INITIAL_THICKNESS
- *CONTROL_FORMING_MAXID
- *CONTROL_FORMING_ONESTEP
- *CONTROL_FORMING_OUTPUT
- *CONTROL_FORMING_PARAMETER_READ
- *CONTROL_FORMING_POSITION
- *CONTROL_FORMING_PRE_BENDING
- *CONTROL_FORMING_PROJECTION
- *CONTROL_FORMING_REMOVE_ADAPTIVE_CONSTRAINTS
- *CONTROL_FORMING_SCRAP_FALL
- *CONTROL_FORMING_SHELL_TO_TSHELL
- *CONTROL_FORMING_STONING
- *CONTROL_FORMING_STRAIN_RATIO_SMOOTH
- *CONTROL_FORMING_TEMPLATE
- *CONTROL_FORMING_TIPPING
- *CONTROL_FORMING_TRAVEL
- *CONTROL_FORMING_TRIM_MERGE

***CONTROL**

***CONTROL_FORMING**

*CONTROL_FORMING_TRIM_SOLID_REFINEMENT

*CONTROL_FORMING_TRIMMING

*CONTROL_FORMING_UNFLANGING

*CONTROL_FORMING_USER

***CONTROL_FORMING_AUTO_NET**

Purpose: Automatically generate rectangular nets with specified dimensions and positions for use with springback. This keyword is used for simulating springback when the stamping panel is resting on the nets of a checking fixture.

Card Sets. Add to the deck as many pairs of Cards 1 and 2 as needed. This section is terminated by the next keyword ("**") card. In general, for N nets add 2N cards.

Card 1	1	2	3	4	5	6	7	8
Variable	IDNET	ITYPE	IDV	IDP	X	Y	Z	
Type	I		I	I	F	F	F	
Default	none		0	0	0.0	0.0	0.0	

Card 2	1	2	3	4	5	6	7	8
Variable	SX	SY	OFFSET					
Type	F	F	F					
Default	0.0	0.0	0.0					

VARIABLE	DESCRIPTION
IDNET	ID of the net; must be unique. See Remark 1 .
ITYPE	Not used at this time.
IDV	Vector ID for surface normal of the net. See *DEFINE_VECTOR . If not defined, the normal vector will default to the global z-axis.
IDP	Part ID of the panel undergoing springback simulation
X	The <i>x</i> -coordinate of a reference point for the net to be generated
Y	The <i>y</i> -coordinate of a reference point for the net to be generated
Z	The <i>z</i> -coordinate of a reference point for the net to be generated

*CONTROL

*CONTROL_FORMING_AUTO_NET

VARIABLE	DESCRIPTION
SX	Length of the net along the first tangential direction. (The <i>x</i> -axis when the normal is aligned along the global <i>z</i> -axis).
SY	Length of the net along the second tangential direction. (The <i>y</i> -axis when the normal is aligned along the global <i>z</i> -axis).
OFFSET	The net center will be offset a distance of OFFSET in the direction of its surface normal. For positive values, the offset is parallel to the normal; for negative values, antiparallel.

Remarks:

- Net ID.** The IDNET field of card 1 sets the “net ID,” which is distinct from the part ID of the net; the net ID serves distinguishes *this* net from *other* nets.
- Properties of the Net.** The part ID assigned to the net is generated by incrementing the largest part ID value in the model. Other properties such as section, material, and contact interfaces between the panel and nets are likewise automatically generated.
- Contact Type.** The auto nets use contact type *CONTACT_FORMING_ONE WAY_SURFACE_TO_SURFACE.

Examples:

The partial input file (see below) specifies four auto nets having IDs 1 through 4. The vector with ID = 89 is normal to the net. The nets are offset 4 mm *below* their reference points; the direction is *below* because the normal vector (ID = 89) is parallel to the *z*-axis and the offset is negative. This example input can be readily adapted to a typical gravity-loaded springback simulation obviating the need for SPC constraints (see *CONSTRAINED_COORDINATE).

```
*CONTROL_FORMING_AUTO_NET
$-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
$ IDNET      ITYPE      IDV      IDP          X          Y          Z
$      1           89         5   2209.82  -33.6332  1782.48
$ SX        SY        OFFSET
$    15.0     15.0      -4.0
$ IDNET      ITYPE      IDV      IDP          X          Y          Z
$      2           89         5   3060.23  -33.6335  1782.48
$ SX        SY        OFFSET
$    15.0     15.0      -4.0
$ IDNET      ITYPE      IDV      IDP          X          Y          Z
$      3           89         5   3061.21   31.4167  1784.87
$ SX        SY        OFFSET
$    15.0     15.0      -4.0
$ IDNET      ITYPE      IDV      IDP          X          Y          Z
$      4           89         5   2208.84   31.4114  1784.87
```

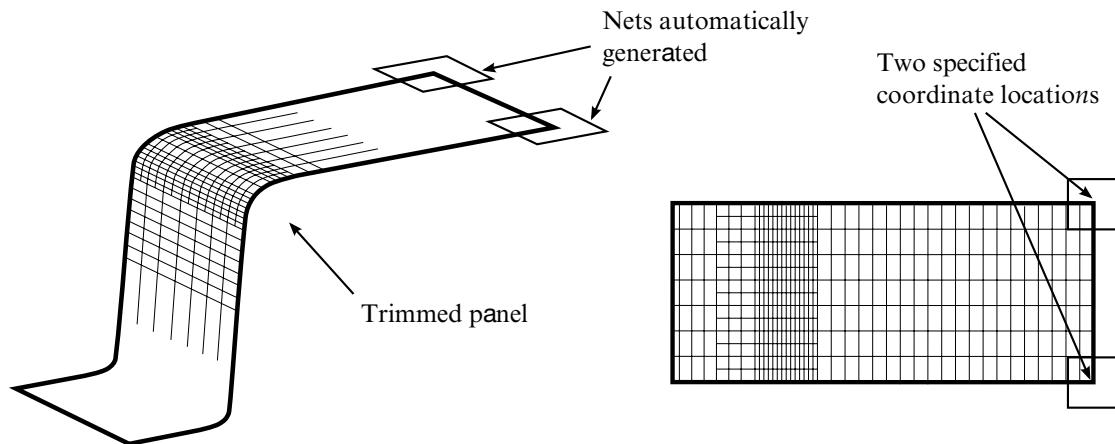


Figure 12-16. An example problem.

```
$      SX      SY      OFFSET
     15.0    15.0     -4.0
*DEFINE_VECTOR
$ VID, Tail X, Y, Z, Head X, Y, Z
89,0.0,0.0,0.0,0.0,0.0,100.0
```

Discussion of Figures:

Figure 12-16 shows a formed and trimmed panel of a hat-shaped channel with an auto net at two corners. The nets are offset 4 mm away from the panel. When gravity loading is downward the nets must be below the panel (Figure 12-17 left) so that the panel comes into contact with the nets after springback as expected (Figure 12-17 right). As shown in Figure 12-18 the situation must be reversed when gravity loading points upward.

Revision Information:

This feature is now available starting in implicit static in double precision LS-DYNA Revision 62781.

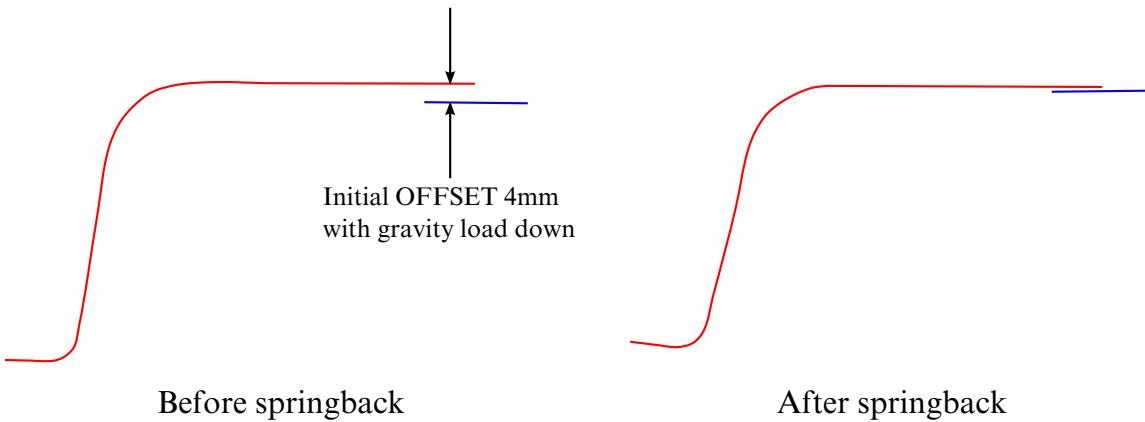


Figure 12-17. Springback and contact with nets - gravity down.

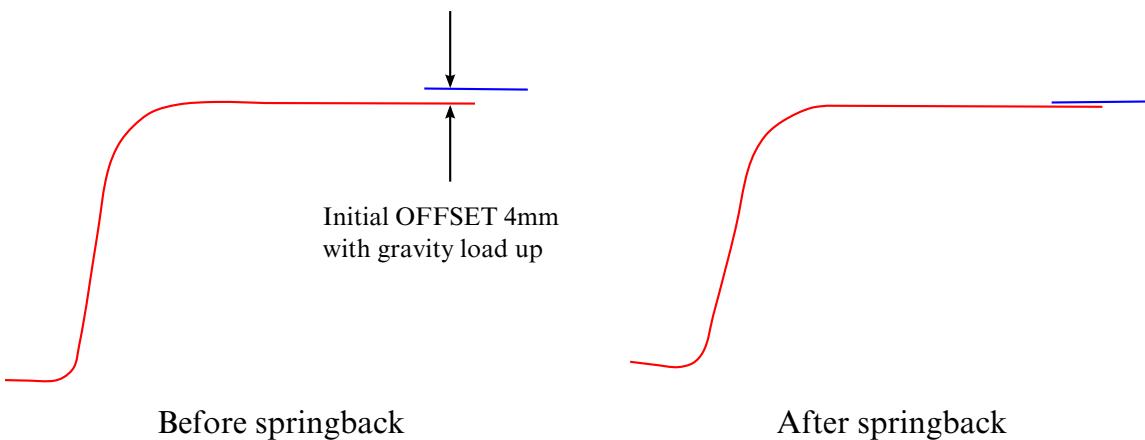


Figure 12-18. Springback and contact with nets – gravity up.

***CONTROL_FORMING_AUTOCHECK**

Purpose: This keyword detects and corrects flaws in the mesh for the *rigid body* that models the tooling. Among its diagnostics are checks for duplicated elements, overlapping elements, skinny/long elements, degenerated elements, disconnected elements, and inconsistent element normal vectors.

This feature also automatically orients each tool's element normal vectors so that they face the blank. Additionally, an offset can be specified to create another tool (tool physical offset) based on the corrected tool meshes. Note that this keyword is distinct from [*CONTROL_CHECK_SHELL](#), which checks and corrects mesh quality problems after trimming, to prepare the trimmed mesh for the next stamping process. This keyword only applies to shell elements.

The tool offset feature is now available in LS-PrePost 4.2 under Application → Metal-Forming → Easy Setup. The offset from Die button under Binder can be used to create offset tools.

Card 1	1	2	3	4	5	6	7	8
Variable	ICHECK	IGD	IOFFSET	IOUTPUT	IFSHARP			
Type	I	I	I	I	I			
Default	0	blank	0	blank	0			

VARIABLE	DESCRIPTION
ICHECK	<p>Tool mesh checking/correcting flag:</p> <p>EQ.0: Do not activate mesh checking/correcting feature.</p> <p>EQ.1: Activate comprehensive mesh check and correct problematic tool meshes (see Figure 12-20). This option reduces the likelihood of unreasonable forming results and/or error termination. This is only for regular forming simulations. The calculation will continue after the tool mesh checking/correcting phase is completed. See Example 1.</p> <p>The corrected tool meshes can be viewed and recovered from the resulting d3plot files. If the termination time is set to “0.0” or the keyword *CONTROL_TERMINATION is absent all together, the simulation will terminate as soon</p>

VARIABLE	DESCRIPTION
	as checking/correcting is completed, and corrected tool meshes can be extracted from the d3plot files.
IGD	Not used.
IOFFSET	Tool mesh offset flag. This variable works only when IOUTPUT is defined, and ICHECK is set to "1": EQ.0: Do not offset rigid tool mesh. The sheet blank does not need to be present. In this case the output files rigid_offset.inc and rigid_offset_before.inc will be identical. See Example 2 . EQ.1: Perform rigid tool mesh offset using the variable SBST (see Figure 12-19) as specified on a *CONTACT_FORMING_... card. The blank must be defined and positioned completely above or below the rigid tool to be offset. Both part ID and part SID (SURFBTYP) can be used in defining SURFB. IOUTPUT must also be defined.
IOUTPUT	Output option flag: EQ.1: Output offset rigid tool meshes into a keyword file rigid_offset.inc , and terminate the simulation. EQ.2: Output offset rigid tool meshes as well as nodes used to define draw beads into a keyword file rigid_offset.inc , and terminate the simulation. See Example 4 . EQ.3: Output checked/corrected tool as well as offset rigid tool meshes into two separate keyword files, rigid_offset_before.inc , and rigid_offset.inc , respectively, and terminate the simulation. See Example 3 . EQ.4: Output checked/corrected tool meshes, offset rigid tool meshes as well as the nodes used to define draw beads into two separate keyword files, rigid_offset_before.inc , and rigid_offset.inc , respectively, and terminate the simulation. If not defined and ICHECK = 1, corrected and reoriented tool meshes can be viewed and recovered from d3plot files. See Example 1 .
IFSHARP	Sharp edge checking option for those tool area without fillet radii: EQ.0: Check any sharp edges and delete the elements.

VARIABLE	DESCRIPTION
	EQ.1: Skip checking sharp edges.

Remarks:

In sheet metal forming, tools are typically modelled as rigid bodies, and their meshes are prepared from CAD (IGES or STEP) files according to the following procedure:

1. The user imports the CAD data into a preprocessor, such as LS-PrePost.
2. The preprocessor automatically generates a mesh. LS-PrePost features a streamlined GUI for this application.
3. Export the generated mesh to LS-DYNA input files. The LS-PrePost eZ-Setup user interface provides quick access to generate the necessary input files for metal forming applications.

Ideally, this process should produce a good mesh requiring no manual intervention. Often, though, such meshes that have been automatically generated from CAD data have flaws severe enough to prevent an accurate or complete calculation. This feature, *CONTROL_FORMING_AUTOCHECK, is intended to make LS-DYNA more robust with respect to tooling mesh quality.

This keyword *requires* that the tooling meshes represent rigid bodies. Also, when this keyword is used, a part ID or a part set ID, corresponding to SURFBTYP = 2 or 3 on the *CONTACT_FORMING_... card, may be used to define the SURFB side. Segment set ID input, SURFBTYP = 0, is not supported.

Some cases of incoming bad tooling meshes which can be corrected by this keyword are shown in [Figure 12-20](#). This keyword can be inserted anywhere in the input deck. To include the corrected tooling mesh into the d3plot the ICHECK field must be defined. The corrected mesh is written to *rigid_offset_before.inc* file if IOFFSET and IOUTPUT are defined.

When IOFFSET = 1 and IOUTPUT is defined, the tool meshes will first be checked, corrected, and reoriented correctly towards the blank. Then the tool is offset by an amount of $0.5|SBST|$ either on the same or opposite side of the blank, depending on the signs of the SBST field on the *CONTACT_FORMING_... card ([Figure 12-19](#)). A new keyword file, “*rigid_offset.inc*” file, will be output as containing the corrected, reoriented, and offset tooling mesh.

Example 1 - Mesh checking/correction in a regular forming simulation:

The keyword can be inserted anywhere in a regular forming simulation input deck. A partial input example of checking, correcting the tool meshes and reorienting all tools' normals is provided below. Note that although SBST is defined between blank and die contact interface, die meshes will not be offset, since IOFFSET is not defined. Simulation will continue if "&endtime" is not zero but will terminate as soon as the checking and correcting are done if "&endtime" is set to "0.0", or *CONTROL_TERMINATION is absent all together. Corrected and reoriented tool meshes can be viewed and recovered from d3plot files.

```
*KEYWORD
*INCLUDE
Tool_blank.k
*CONTROL_FORMING_AUTOCHECK
$ ICHECK      IGD    IOFFSET   IOUTPUT
      1
*CONTROL_TERMINATION
&endtime
:
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ID
  1 blank to punch
    1          2          2          2
  0.110E+00 0.000E+00 0.000E+00 0.000E+00 0.200E+02           1          1
  0.000      0.000          0.0
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ID
  2 blank to die
$ SURFA      SURFB    SURFATYP  SURFBTYP  SABOXID  SBBOXID  SAPR     SBPR
  1          3          2          2
$ FS         FD        DC        VC        VDC       PENCHK    BT        DT
  0.110E+00 0.000E+00 0.000E+00 0.000E+00 0.200E+02           0.000E+00 0.100E+21
$ SFSA      SFSB      SAST      SBST      SFSAT    SFSBT     FSF      VSF
  0.000      0.000          -1.600
$ SOFT      SOFSCL   LCIDAB    MAXPAR    PENTOL   DEPTH     BSORT   FRCFRQ
  0
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ID
  3 blank to binder
    1          4          2          2
  0.110E+00 0.000E+00 0.000E+00 0.000E+00 0.200E+02           1          1
  0.000      0.000          0.0
:
*END
```

Example 2 - Mesh checking/correction only for rigid tool mesh (sheet blank not required):

A much shorter but complete input example of checking, correcting the tool meshes and reorienting all tools' normal vectors is shown below. Note the sheet blank does not need to be present, and both rigid_offset.inc and rigid_offset_before.inc will be the same, representing the checked, corrected, and reoriented tool mesh file, since IOFFSET is undefined (no tool offset will be done).

```
*KEYWORD
*INCLUDE
toolmesh.k
*CONTROL_FORMING_AUTOCHECK
```

```

$ ICHECK      IGD    IOFFSET   IOUTPUT
      1                      1
*PARAMETER_EXPRESSION
I toolpid      3
*PART

$ PID        SECID   MID     EOSID    HGID    GRAV    ADPOPT   TMID
&toolpid          2       2
*MAT_RIGID
$ MID        RO      E       PR       N       COUPLE   M       ALIAS
  2  7.83E-09  2.07E+05  0.28
$ CMO        CON1    CON2
  1       4       7
$ LCO or A1  A2      A3     V1      V2      V3

*SECTION_SHELL
$ SECID    EIFORM   SHRF    NIP      PROPT   QR/IRID   ICOMP   SETYP
  2       2       1.0    3.0      0.0
$ T1        T2      T3       T4      NLOC
  1.0      1.0      1.0    1.0

*END

```

Example 3 - Mesh checking/correction and tool offset (sheet blank required):

In addition to checking, correcting, and reorienting all tools' normals, the following partial input will offset the die meshes in toolmesh.k by 0.88 mm (using the SBST value defined for the die) on the opposite side of the blank, and output the offset tool meshes in the file rigid_offset.inc. The checked/corrected original die meshes will be written to rigid_offset_before.inc. The simulation will terminate as soon as the files are written, regardless of what the "&endtime" value is. In fact, the keyword *CONTROL_TERMINATION can be omitted all together.

```

*KEYWORD
*INCLUDE
Tool_blank.k
*PARAMETER_EXPRESSION
R blankt      0.8
R offset      -1.1
R sbst        blankt*offset*2.0
*CONTROL_FORMING_AUTOCHECK
$ ICHECK      IGD    IOFFSET   IOUTOUT
      1                      1       3
*CONTROL_TERMINATION
&endtime
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ID
  1 blank to punch
    1           2           2           2
    0.110E+00  0.000E+00  0.000E+00  0.000E+00  0.200E+02
    0.000      0.000          0.0
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ID
  2 blank to die
$ SURFA      SURFB    SURFATYP  SURFBTYP  SABOXID  SBBOXID  SAPR    SBPR
    1           3           2           2
$ FS         FD       DC        VC        VDC      PENCHK   BT      DT
    0.110E+00  0.000E+00  0.000E+00  0.000E+00  0.200E+02
    0.000      0.000          &sbst
$ SOFT       SOFSCL   LCIDAB   MAXPAR   PENTOL   DEPTH    BSORT   FRCFRQ
    0
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ID

```

*CONTROL

*CONTROL_FORMING_AUTOCHECK

```
3 blank to binder
 1      4      2      2
0.110E+00 0.000E+00 0.000E+00 0.000E+00 0.200E+02           1      1
 0.000      0.000          0.0
*END
```

Example 4 - Mesh checking/correction and tool offset, bead nodes output (sheet blank required):

In addition to checking, correcting, and reorienting all tools' normals, the following partial input will create an offset tool in the file rigid_offset.inc on the same side of the blank; the file will also contain the nodes used to define the contact draw beads #1 and #2.

```
*KEYWORD
*INCLUDE
Tool_blank.k
R blankt      0.8
R offset      1.1
R sbst        blankt*offset*2.0
*CONTROL_FORMING_AUTOCHECK
$ ICHECK      IGD    IOFFSET   IOUTPUT
  1           1       2
*CONTROL_TERMINATION
&endtime
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ID
  1 blank to punch
  1      2      2      2
0.110E+00 0.000E+00 0.000E+00 0.000E+00 0.200E+02           1      1
  0.000      0.000          0.0
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ID
  2 blank to die
$ SURFA      SURFB    SURFATYP   SURFBTYP   SABOXID   SBBOXID     SAPR      SBPR
  1          3          2          2
$ FS         FD        DC          VC          VDC        PENCHK      1          1
0.110E+00 0.000E+00 0.000E+00 0.000E+00 0.200E+02           0.000E+00 0.100E+21
$ SFSA      SFSB      SAST        SBST        SFSAT      SFSBT      FSF        VSF
  0.000      0.000          &sbst
$ SOFT      SOFSCL   LCIDAB     MAXPAR     PENTOL     DEPTH      BSORT     FRCFRQ
  0
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ID
  3 blank to binder
  1      4      2      2
0.110E+00 0.000E+00 0.000E+00 0.000E+00 0.200E+02           0.000E+00 0.100E+21
  0.000      0.000          0.0
:
*CONTACT_DRAWBEAD_ID
  10001 Draw bead #1
  1      1      4      2
0.110E+00 0.000E+00 0.000E+00 0.000E+00 0.200E+00           0      0      0
  0.200      0.200
$ LCIDRF    LCIDNF    DBDTH      DFSCL      NUMINT
  10      9 0.100E+02 0.700E+00
*CONSTRAINED_EXTRA_NODES_SET
  40      1
*SET_NODE_LIST
  1
  915110    915111    915112    915113    915114    915115    915116    915117
  915118    915119    915120    915121    915122
*CONTACT_DRAWBEAD_ID
  10002 Draw bead #2
  2      1      4      2
0.110E+00 0.000E+00 0.000E+00 0.000E+00 0.200E+00           0      0      0
  0.0615    0.100E+21
```

CONTROL_FORMING_AUTOCHECK**CONTROL**

```
    0.200      0.200
$  LCIDRF      LCIDNF      DBDTH      DFSCL      NUMINT
    11          9 0.100E+02 0.400E+00
*CONSTRAINED_EXTRA_NODES_SET
$      PID      NSID
    40          2
*SET_NODE_LIST
    2
  915123    915124    915125    915126    915127    915128    915129    915130
  915131    915132    915133    915134    915135    915136    915137
:
*END
```

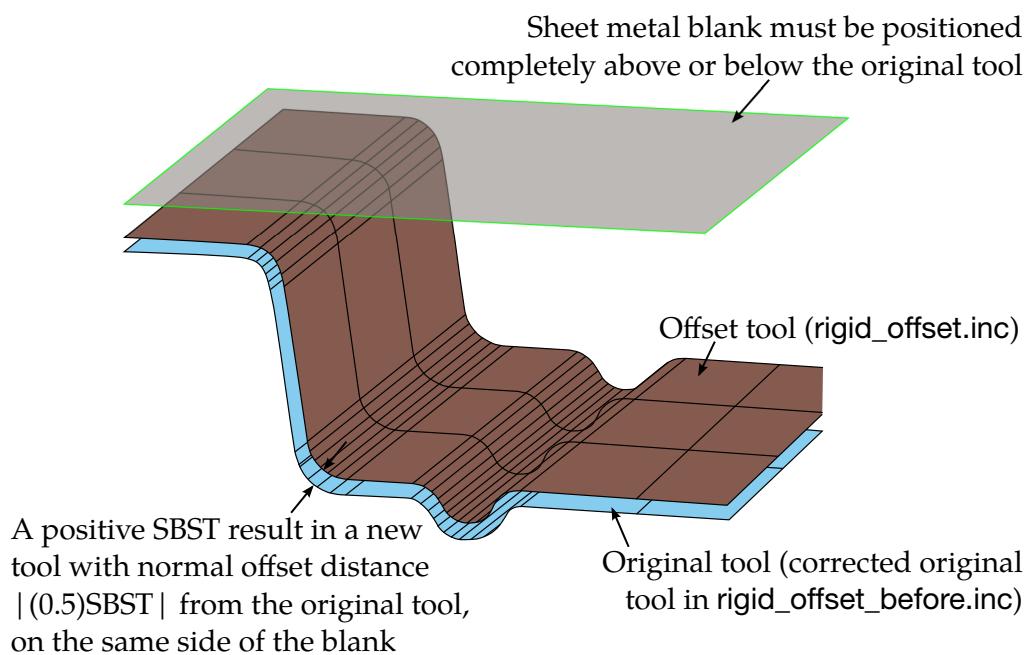
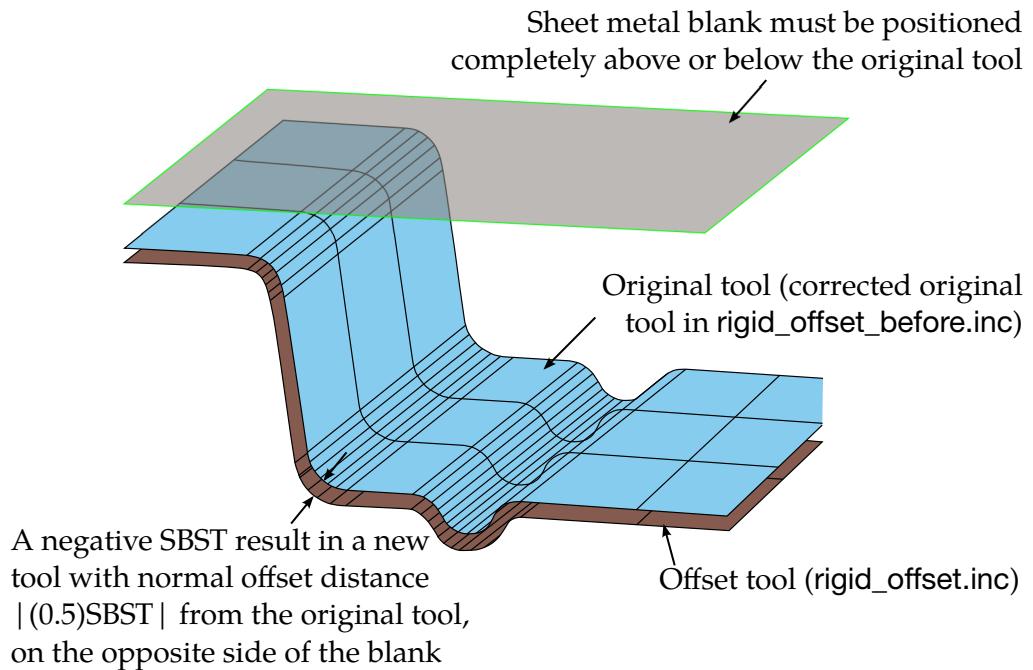


Figure 12-19. Offset using the SBST value defined in *CONTACT_FORMING_...

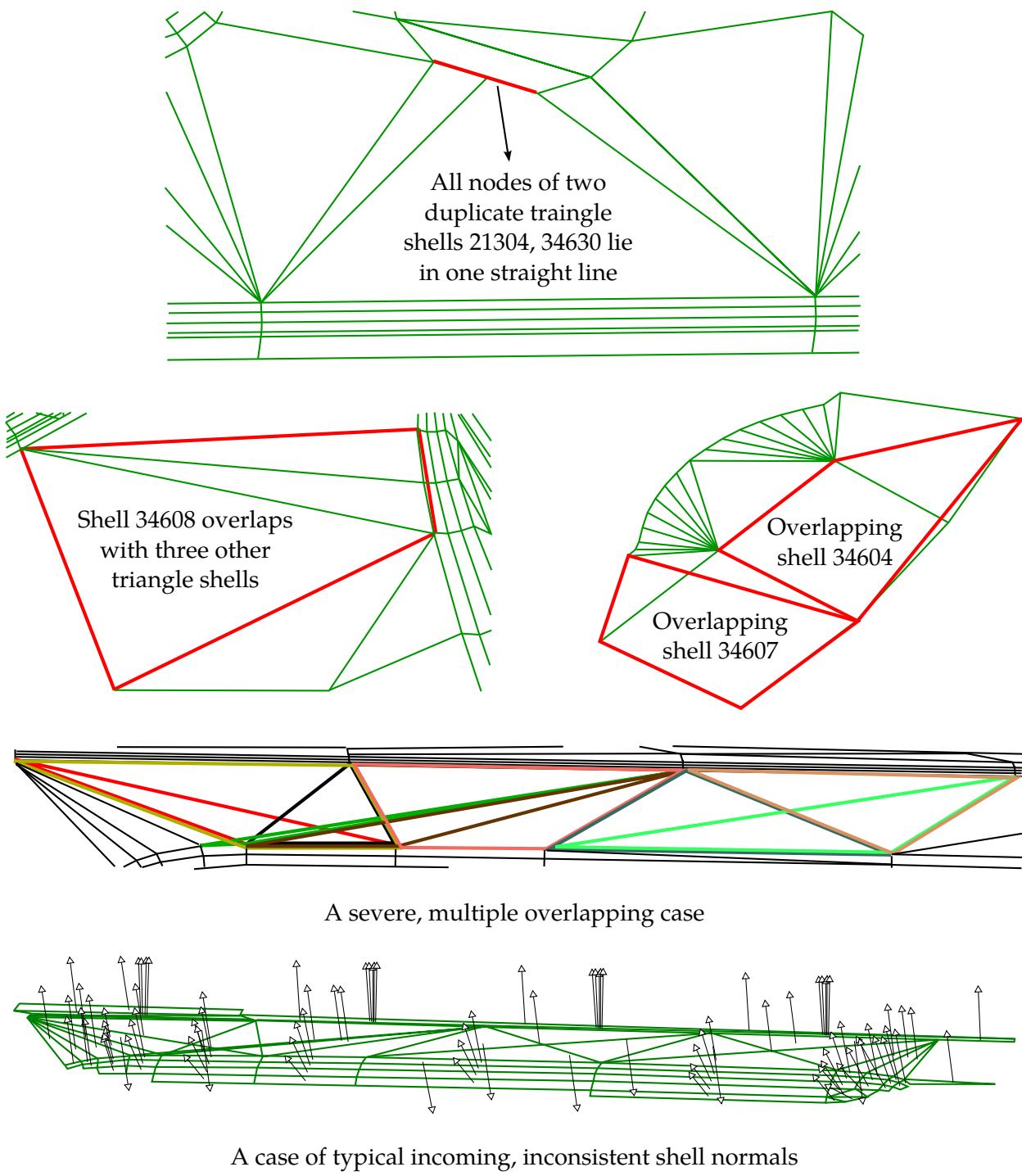


Figure 12-20. A few cases of the tooling mesh problems handled by this keyword.

*CONTROL

*CONTROL_FORMING_AUTOPOSITION_PARAMETER

***CONTROL_FORMING_AUTOPOSITION_PARAMETER_{OPTION1}_{OPTION2}**

Available options for *OPTION1* include:

<BLANK>

SET

With SET, a set of parts will be repositioned together.

Available options for *OPTION2* include:

<BLANK>

POSITIVE

When *OPTION2* is set to POSITIVE, the calculated distance to reposition will always be a positive value.

Purpose: The purpose of this keyword is to *calculate* the minimum required separation distances among forming tools for initial tool and blank positioning in metal forming simulation. It is applicable to sheet blanks with shell and solid elements. It does not, actually, move the part; for that, see [*PART_MOVE](#).

NOTE: This keyword requires that the model begins in its home position. While processing this card, LS-DYNA moves the parts to match the auto-position results so that auto-position operations correctly compose. Upon completion of the auto-positioning phase, the parts are returned to their home positions.

Auto-Position Part Cards. Add one card for each part to be auto-positioned. The next keyword ("*") card terminates this input.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	CID	DIR	MPID	POSITION	PREMOVE	THICK	PORDER
Type	I	I	I	I	I	F	F	I/A
Default	none	global	none	none	0	0.0	0.0	none

VARIABLE	DESCRIPTION
PID	<p>Part ID. This part will be moved based on the following controlling parameters.</p> <p>When the option SET is activated, PID becomes part set ID, defined by *SET_PART_LIST. This is useful in defining tailor-welded blanks, where two pieces of the blank must be moved simultaneously.</p>
CID	<p>Coordinate ID set with *DEFINE_COORDINATE_SYSTEM or *DEFINE_COORDINATE_VECTOR. The default is the global coordinate system.</p> <p>LT.0: CID is vector ID giving the direction the part will be moved.</p>
DIR	<p>Direction in which the part will be moved:</p> <ul style="list-style-type: none">EQ.1: x-direction,EQ.2: y-direction,EQ.3: z-direction.
MPID	<p>Master part ID, whose position is to be referenced by PID for positioning. When the option SET is activated, MPID becomes part set ID, defined by *SET_PART_LIST.</p>
POSITION	<p>Definition of relative position between PID and MPID:</p> <ul style="list-style-type: none">EQ.1: PID is above MPID;EQ.-1: PID is below MPID. <p>Definition of "above" is determined by the defined coordinate system. If PID is above MPID, it means PID has a larger z-coordinate. This definition is helpful in line-die simulations where the local coordinate system may be used.</p>
PREMOVE	<p>Move PID through distance PREMOVE <i>prior</i> to processing the other *CONTROL_FORMING_AUTOPOSITION cards. See Remark 4.</p>
THICK	<p>One half of this distance value (THICK) will be used to separate the tools and the blank. The same value must be used in <i>all</i> defined move operations under this keyword. The calculation error of the separation distance is one half of THICK.</p>

VARIABLE	DESCRIPTION
PORDER	The name of the parameter without the ampersand “&,” as defined in *PARAMETER, or the position or order of the parameter defined in the *PARAMETER list.

Background:

In a line-die (multi-stage) simulation, initial positioning of the tools and blank is one of the major issues preventing several die processes from being run automatically from a single job submission. The most basic method for running a line-die simulation is to chain a series of calculations together using the previous calculation’s partially formed blank, written to a dynain file, as a part of the input for the next calculation.

Since the partial results are unknown until the preceding calculation completes, the tools need to be repositioned before the next calculation. Without this card the repositioning step must be done by hand using a preprocessor. With the combination of this card and the LS-DYNA case driver (see *CASE card) or a continuous run script (works for both Linux and Windows), the repositioning can be fully automated, enabling a complete line-die simulation to be performed with a single job submission.

Workflow:

This card requires that all parts start in their home (tool closed) position. It calculates how far the parts need to be moved to prevent initial penetration. The results are stored into the parameter listed in the PORDER field to be used for a part move operation.

1. For each defined move operation a *PARAMETER card *must* initialize the parameter referred to in the PORDER field.
2. All tools must start in home position including *desired final gaps*.
3. The required distance between each contact pair is calculated and stored in the initialized parameter named in the PORDER field. Starting with Revision 124103, if the separation distance cannot be found, such as when MPID is not found or is out of position, or DIR is not input correctly, the value of PREMOVE will be returned instead of a very large number.
4. The parts are repositioned through a distance based on the value written to the parameter PORDER using the *PART_MOVE card.
5. The *PARAMETER_EXPRESSION can be used to evaluate expressions depending on the move distances, such as times and tool move speeds.

6. The *CASE feature, can be used to chain together the sub-processes in a line die (process chain) simulation. Alternatively, a continuous run script (for both Linux and Windows environment) can be generated for an entire process chain simulation using the Metal Forming GUI starting in LS-PrePost 4.3.

Remarks:

1. **Order Dependence.** Input associated with this keyword is order sensitive. The following order should be observed:
 - a) All model information *including* all elements and node
 - b) Part definitions (see *PART)
 - c) Part set definitions (see *SET_PART_LIST)
 - d) *PARAMETER initialization
 - e) This keyword
 - f) *PARAMETER_EXPRESSION
 - g) *PART_MOVE
2. **New Keyword Input with Positioned Model.** This keyword can also be used to generate a new keyword input (dynain) containing the fully positioned model (without actually running the entire simulation). This procedure is identical to a full calculation except that the *PARAMETER_EXPRESSION keyword, the *CONTROL_TERMINATION keyword, and tool kinematic definitions are omitted.
3. **Local Coordinate Systems and Computed Parameters.** When working in local coordinate systems, the sign of the computed parameter may not necessarily correspond to its intended use. In this case, the absolute value function, ABS, for the *PARAMETER_EXPRESSION keyword is especially useful.
4. **PREMOVE.** Cards with the PREMOVE field set are processed before *all* other *CONTROL_FORMING_AUTOPOSITION cards, regardless of their location in the input deck. The PREMOVE field serves to modify the initial state on which the calculations of the other AUTOPOSITION cards are based.

For instance, when a binder is moved downward with the PREMOVE feature, it will be in its post-PREMOVE position for *all other* AUTOPOSITION calculations. But, as is the case with the other AUTOPOSITION cards, the model will be returned to its home position upon completion of the AUTOPOSITION phase. Note that the master part, MPID, and the POSITION fields are *ignored* when the

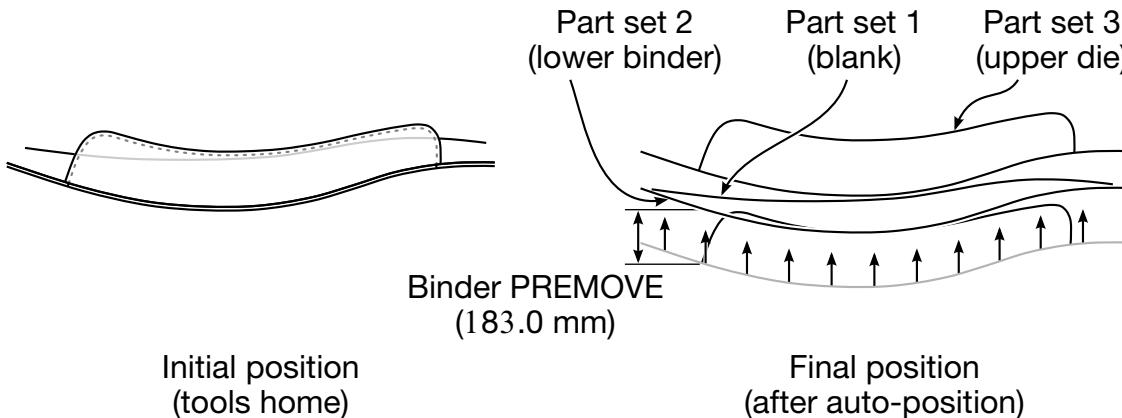


Figure 12-21. An example of using the variable PREMOVE

PREMOVE field is set, and that the PREMOVE value is copied into the PORDER parameter.

5. **CALCULATED MOVE DISTANCE.** Calculated move distance for each part will be displayed in the beginning of the messag file. All parameters (defined and calculated) can be found in the d3hsp file.
6. **LS-PrePost.** This feature is implemented starting in LS-PrePost4.0 eZSetup for metal forming in both explicit and implicit application.

Example 1:

An air draw process like the one shown in [Figure 12-21](#) provides a clear illustration of how this card, and, in particular, the PREMOVE field is used to specify the lower binder's travel distance.

1. The card with the PREMOVE field set, the *third* AUTOPOSITION card, is processed first. It moves lower binder 183 mm upward from its home position, and it will form the base configuration for other AUTOPOSITION cards. It will also store this move into &bindmv. Note that although the POSITION and MPID fields are set, they are ignored.
2. The *first* autoposition card, which will be the *second* one processed, calculates the minimum offset distance (&blankmv) necessary for the blank (part set 1) to clear part set 9999, which consists of the lower binder (PID = 2), which is in its post-PREMOVE location, and of the lower punch (PID = &lpunid).
3. The next card determines the minimum offset (&updiemv) necessary to bring the upper die (part set 3) as close to the blank as possible without penetrating. *This calculation proceeds under the assumption that the blank part set has been moved through &blankmv.*

```

*SET_PART_LIST
9999
&lpunpid,2
$-----1-----2-----3-----4-----5-----6-----7-----8
*CONTROL_FORMING_AUTOPOSITION_PARAMETER_SET
$---+---1---+---2---+---3---+---4---+---5---+---6---+---7---+---8
$ PID CID DIR MPID POSITION PREMOVE THICK PORDER
$ blank move
    1            3      9999      1          &bthick   blankmv
$ upper die move
    3            3      1           1          &bthick   updiemv
$ lower binder move
    2            3      1      -1     183.0  &bthick   bindmv
$-----1-----2-----3-----4-----5-----6-----7-----8
*PART_MOVE
$ SID XMOV YMOV ZMOV CID IFSET
$ blank move
    1      0.0  0.0  &blankmv  1
$ upper die move
    3      0.0  0.0  &updiemv  1
$ lower binder move
    2      0.0  0.0  &bindmv  1

```

Example 2:

The following examples demonstrates the *PARAMETER_EXPRESSION card, which is used to derive new parameters from the value calculated during auto-positioning. In this example, the auto-positioned distance for binder, which is stored in the parameter, &bindmv, is used to define an additional parameter,

$$\&bindmv1 = \&bindmv - 30 \text{ mm}$$

The *PART_MOVE step uses &bindmv1 rather than &bindmv, to move both the lower binder and the draw beads.

```

$-----1-----2-----3-----4-----5-----6-----7-----8
*CONTROL_FORMING_AUTOPOSITION_PARAMETER_SET
$---+---1---+---2---+---3---+---4---+---5---+---6---+---7---+---8
$ PID CID DIR MPID POSITION PREMOVE THICK PORDER
$ blank move
    &blksid        3      9999      1          &bthick   blankmv
$ upper die move
    &udiesid       3      &blksid     1          &bthick   updiemv
$ lower binder move
    &bindsid       3      &blksid     -1         &bthick   bindmv
$-----1-----2-----3-----4-----5-----6-----7-----8
*PARAMETER_EXPRESSION
bindmv1 bindmv-30.0
$-----1-----2-----3-----4-----5-----6-----7-----8
*PART_MOVE
$ SID XMOV YMOV ZMOV CID IFSET
$ blank move
    &blksid       0.0    0.0  &blankmv  1
$ upper die move
    &udiesid       0.0    0.0  &updiemv  1
$ lower binder move
    &bindsid       0.0    0.0  &bindmv1  1
$ draw beads move
    909          0.0    0.0  &bindmv1  1

```

*CONTROL

*CONTROL_FORMING_AUTOPOSITION_PARAMETER

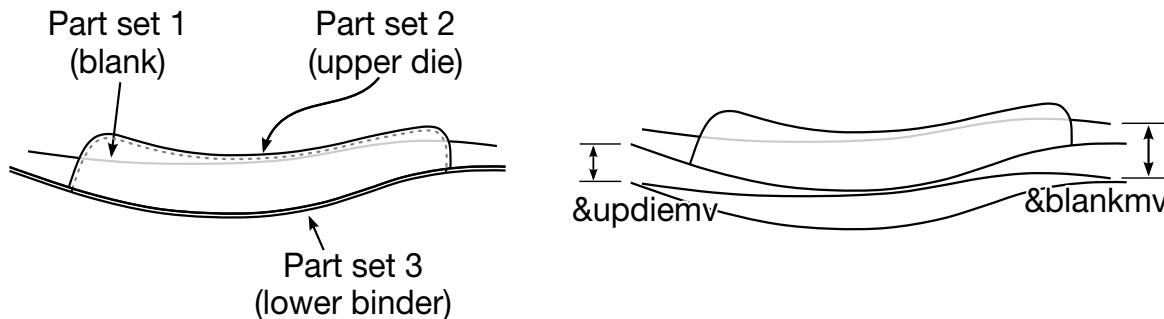


Figure 12-22. An example of binder closing in air draw

Example 3:

Figure 12-22 schematically shows the binder closing in the global Z-direction. A partial keyword details follow.

```
*INCLUDE
$blank from previous case
case5.dynain
*INCLUDE
closing_tool.k
*INCLUDE
beads_home.k
*SET_PART_LIST
$ blank
1
1
*SET_PART_LIST
$ upper die
2
2
*SET_PART_LIST
$ lower binder
3
3
-----1-----2-----3-----4-----5-----6-----7-----8
*parameter
$$$$$$$$$$$$$$$$$$$$$$$$$$$$ Tool move variables
R blankmv      0.0
R updiemv      0.0
R bindmv       0.0
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ Tool speed and ramp up definition
R tclsup        0.001
R vcls          1000.0
-----1-----2-----3-----4-----5-----6-----7-----8
*CONTROL_FORMING_AUTOPOSITION_PARAMETER_SET
$     PID      CID      DIR      MPID    POSITION   PREMOVE   THICK   PORDER
$ positioning blank on top of lower binder
      1           3           3           1           0.7      blankmv
$ positioning upper die on top of blank
      2           3           1           1           0.7      updiemv
-----1-----2-----3-----4-----5-----6-----7-----8
*PARAMETER_EXPRESSION
$     PRMR1      EXPRESSION
R clstime      (abs(updiemv)-vcls*tclsup)/vcls+2.0*tclsup
R endtime      &clstime
-----1-----2-----3-----4-----5-----6-----7-----8
*PART_MOVE
$     PID          XMOV          YMOV          ZMOV          CID
```

CONTROL_FORMING_AUTOPOSITION_PARAMETER**CONTROL**

1	0.0	0.0	&blankmv
2	0.0	0.0	&updiemv
\$-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8			
*CONTROL_TERMINATION			
&endtime			

*CONTROL

*CONTROL_FORMING_BESTFIT

*CONTROL_FORMING_BESTFIT

Available options include:

<BLANK>

VECTOR

Purpose: Rigidly move a part to the target using an iterative least-squares method so that they maximally coincide. This feature can be used to assess the accuracy of the spring-back prediction in sheet metal forming by translating and rotating a sprung part (source) to a scanned part (target). This keyword applies to shell elements only. The VECTOR option allows vector components of the normal distance from the target to the part node to be included in the output file, bestfit.out, under the keyword *NODE_TO_TARGET_VECTOR.

This feature is available as of LS-PrePost 4.5 in *Metal Forming Application/eZ Setup*.

Card 1	1	2	3	4	5	6	7	8
Variable	IFIT	NSKIP	GAPONLY	IFAST	IFSET	NSETS	NSETT	
Type	I	I	I	I	I	I	I	
Default	0	-3	0	1	0	none	none	

Card 2	1	2	3	4	5	6	7	8
Variable				FILENAME				
Type				A80				
Default				none				

VARIABLE

DESCRIPTION

IFIT

Best fit program activation flag:

EQ.0: do not perform best-fit.

EQ.1: activate the best-fit program.

VARIABLE	DESCRIPTION
NSKIP	Optional skipping scheme during bucket searching to aid the computational speed (zero is no skipping): GT.0: Number of nodes to skip in bucket searching. NSKIP set to 1 does not skip any nodes in searching, resulting in the slowest computing speed but the highest accuracy. Larger values of NSKIP speed up the calculation time with slightly deteriorating accuracies. Based on studies, a value of 5 is recommended with IFAST = 1, which balances the speed and accuracy. LT.0: Absolute value is the distance to skip in bucket searching. This scheme is faster compared to the previous method and therefore is recommended for computational efficiency and accuracy. A value of -5 is suggested.
GAPONLY	Separation distance calculation flag: EQ.0: after performing best-fit, calculate the separation between the two parts. EQ.1: no best-fit, just calculate separation between the two existing parts. EQ.2: the user is responsible for moving the parts closer together in both distance and orientation for the situation where the target and source are not similar in shape. Also see NSETS and NSETT (recommended method).
IFAST	Computing performance optimization flag: EQ.0: no computing speed optimization. EQ.1: activate computing speed optimization (default and recommended).
IFSET	Optional flag to define a node set to be included or excluded in the source mesh file for best fitting. The node set can be defined in a file together with the source mesh. EQ.0: All nodes in the source mesh file will be best fitted. GT.0: The input value is a node set ID; only the nodes in the set will be best fitted. LT.0: The absolute value is a node set ID; all nodes excluding those in the set will be best fitted.

VARIABLE	DESCRIPTION
NSETS	An optional node set ID of three nodes from the source mesh to help align the source to the target. See Remark 4 . The nodes should be selected based on distinctive geometry features, such as, the center of an arc, the center of a dart, or the end node of a take-up bead (see Figure 12-23). The three nodes must not be aligned in one straight line. Define NSETS if the orientations of the source mesh and the target have a large deviation ($> \sim 30$ degrees in any direction). This method is recommended.
NSETT	An optional node set ID of three nodes from the target mesh that consists of nodes corresponding to the same geometry features as the source mesh to help align the source to the target. The three nodes should be input in the same order as those from the source mesh. Approximate locations are acceptable. See Remark 4 . Define NSETT only if NSETS is defined. See Figure 12-23 for details. This method is recommended.
FILENAME	Target mesh file in keyword format, in which only *NODE and *ELEMENT_SHELL should be included. The target mesh is typically the scanned part.

Remarks:

1. **Source Mesh.** The source mesh can be included in the input file using *INCLUDE.
2. **Target Mesh Coarseness.** To reduce the computing time, the scan file (STL) mesh can be coarsened in a scan-processing software from a typically very dense mesh to a more reasonably sized mesh.
3. **Fit Evaluation.** The distances between corresponding portions of the two parts are calculated after they are fitted. The distance is given as a positive or negative value based on the target's normal directions. The distance vector points from the target to the corresponding portion on the source. If the projection of the distance vector onto the target's normal vector at this portion is in the direction of the target's normal direction, then the distance is "positive", otherwise it is "negative." For areas where no corresponding meshes can be found between the two parts, the distances are set to nearly zero. The fitting accuracy is within 0.02 mm.

The distances are stored as thickness values in `bestfit.out`. They can be plotted in LS-PrePost using *COMP* → *Thickness*. By importing both `bestfit.out` and the

target mesh to LS-PrePost, the deviation on cut-sections can be evaluated using the *SPLANE* feature.

4. **Initial Orientation.** If the orientation of the source and target exceeds 30 degrees, the fit process becomes computationally costly. NSETS and NSETT can be used to initially align the source mesh to the target mesh before a full fitting is performed.
5. **Output.** The following information will be output to the results file **bestfit.out**. The summary gives a percentage of the source nodes that are between 0.0 and 6.0 mm of the target, with a 0.5 mm interval range, along with the maximum deviation value of the source from the target. A transformation matrix as well as the transformation given as a set of euler angles (radians) and a translation vector (in mm) are also output starting in Revision 140609. The transformation matrix can be used directly in the keyword ***INCLUDE_STAMPED_PART_MATRIX** to perform a validation or for other needs. The euler angle and translation vector form allows you to transform (note it is order sensitive) the source to the target in a more direct method, namely rotate in RZ first, followed by RY and RX, followed by translations in DX, DY, and DZ.

```
$      Summary:  
$  between 0.00 to 0.50:    100.000   100.000  
$  between 0.50 to 1.00:     0.000   100.000  
$  between 1.00 to 1.50:     0.000   100.000  
$  between 1.50 to 2.00:     0.000   100.000  
$  between 2.00 to 2.50:     0.000   100.000  
$  between 2.50 to 3.00:     0.000   100.000  
$  between 3.00 to 3.50:     0.000   100.000  
$  between 3.50 to 4.00:     0.000   100.000  
$  between 4.00 to 4.50:     0.000   100.000  
$  between 4.50 to 5.00:     0.000   100.000  
$  between 5.00 to 5.50:     0.000   100.000  
$  between 5.50 to 6.00:     0.000   100.000  
$ The maximum deviation is:    0.004  
$ *TRANSFORMATION MATRIX  
$    0.649517E+00    0.433010E+00    0.625003E+00    -0.365657E+03  
$    0.125007E+00    0.749999E+00    -0.649519E+00    -0.216987E+02  
$   -0.750000E+00    0.500003E+00    0.433008E+00    0.252244E+02  
$  
$ --OR--  
$      RZ:    -0.588001E+00  
$      Ry:    0.675136E+00  
$      Rx:    0.982798E+00  
$  DX,DY,DZ:   -0.365657E+03    -0.216987E+02    0.252244E+02
```

Example 1 – fitting with NSETS and NSETT (recommended):

In the following partial keyword example (meshes and results shown in [Figure 12-23](#)) a source mesh **sourcemesh.k** is being best-fitted to a target mesh **targetmesh.k**.

Node sets 1 and 2 are defined from nodes in the source and target meshes, respectively, to help with alignment. Node ID 1001 and 1 are both located at the center of a dart on the top surface of the hat-shaped part. Node ID 1002 and 2 are selected at the center of

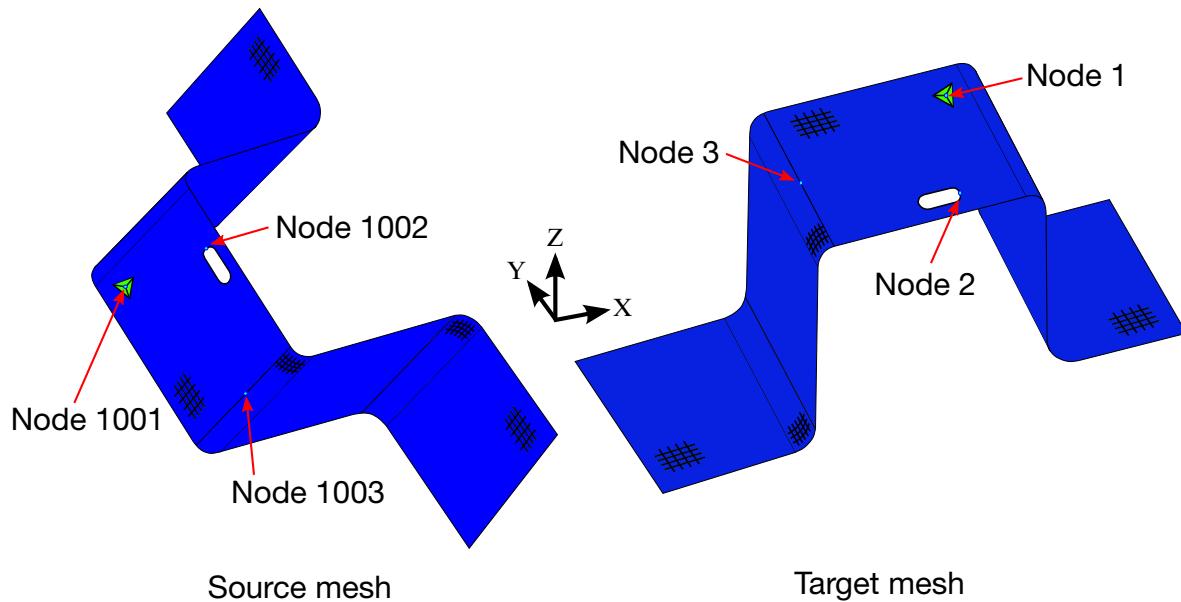
an arc of a cutout hole. Lastly, node ID 1003 and 3 are at the center of a tangent line of a radius.

In this example, since the source and target meshes are exactly the same, the normal distance, as displayed by *thickness* is nearly zero everywhere.

```
*CONTROL_FORMING_BESTFIT
-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
$#      IFIT      NSKIP     GAPONLY      IFAST      IFSET      NSETS      NSETT
      1          -5          0           1          0           1           2
$# FILENAME
targetmesh.k
*INCLUDE
sourcemesh.k
*SET_NODE_LIST
1
1,2,3
*SET_NODE_LIST
2
1001,1002,1003
```

Revision information:

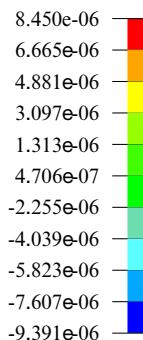
This feature is available starting from LS-DYNA Revision 96427 double precision SMP. The variable IFSET is available starting from Revision 96696. The variables NSETS, NSETT are available starting from Revision 99369. The VECTOR option is available starting from Revision 112655. A transformation matrix, and other transformation option in angles (in radians) and translation (in mm) are available starting in Revision 140609



Best fit results of part separation
Contours of shell thickness
min=-9.39123e-06 at elem# 102
max=8.45032e-06 at elem# 149

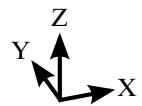
Node 1: geometry feature such as the center of a dart is a preferred choice to be one of the three nodes.

Part Separation (mm)



Node 3: the center node of a tangent line may also be used.

Node 2: the center of an arc of a hole can also be used to select one of the three nodes.



Best fit results - color contour of part separation plotted with "thickness" from the output file "Bestfit.out"

Figure 12-23. Best fit of two meshes with an orientation difference of greater than 30 degrees.

***CONTROL**

***CONTROL_FORMING_HOME_GAP**

***CONTROL_FORMING_HOME_GAP**

Purpose: Calculate the minimum gap between the upper and lower tools. Gaps smaller than the blank thickness can lead to problems with the simulation. The gap is initially measured from the home position. Then the tools are incrementally moved to the starting position where the gap is again measured.

Card 1	1	2	3	4	5	6	7	8
Variable	PSIDU	PSIDL	GAP	MVINC	ISTOP			
Type	I	I	F	F	I			
Default	none	none	none	none	0			

VARIABLE	DESCRIPTION
PSIDU	Part set ID of the tools above the blank (upper tools)
PSIDL	Part set ID of the tools below the blank (lower tools)
GAP	Minimum gap allowed between the upper and lower tools
MVINC	Incremental movement of tools from home position to starting position to check the gap
ISTOP	How to proceed if the minimum gap found is less than GAP: EQ.0: Output a warning message. Job continues EQ.1: Terminate the job.

***CONTROL_FORMING_INITIAL_THICKNESS**

Purpose: Specify a varying thickness field along a specific direction on a sheet blank (shell elements only) as a result of a metal forming process, such as a tailor-rolling, that will be used for an additional metal forming simulation. A related keyword is *ELEMENT_SHELL_THICKNESS.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	LCID	X0	Y0	Z0	VX	VY	VZ
Type	I	I	F	F	F/I	F	F	F
Default	none							

VARIABLE	DESCRIPTION
PID	Part ID of the sheet blank to be defined with varying thickness, as in *PART. Currently only 1 PID is allowed.
LCID	Load curve ID defining thickness (ordinate) as a function of distance (abscissa) starting from position coordinates (X0,Y0,Z0) and in the direction of the vector (VX,VY,VZ). See *DEFINE_CURVE.
X0, Y0, Z0	Starting position coordinates
VX, VY, VZ	Vector components defining the direction of the distance in the load curve

Background:

Tailor-rolling is a process used to vary the thickness of the blank. A judiciously designed and manufactured tailor-rolled blank will reduce the number of parts (reinforcements) involved in the stamping process, as well as the number tools needed to make them. By reducing the number of spot welds, tailor-rolled pieces also possess superior structural integrity.

Remarks:

1. **Thickness Curve.** Beyond the last data point LS-DYNA extrapolates the load curve specified in LCID as being constant.

2. **Overriding Set Thickness.** This card overrides thicknesses set with the *SECTION_SHELL keyword.

Application Example:

A reduced input deck containing a characteristic example of this keyword's application is given below. In this example the blank is part ID 1. The axis of the load curve starts at position (-295, -607, -43) and the direction along which the load curve sets the thickness is given by (524, 607, 0). For each of the load curve's abscissa values, t , the corresponding geometrical coordinate is given by:

$$\mathbf{r} = \begin{bmatrix} -295 \\ -607 \\ -43 \end{bmatrix} + \begin{bmatrix} 524 \\ 607 \\ 0 \end{bmatrix} t$$

For negative values along the load curve, $t < 0$, and values of $t > 101.0$, the thickness is extrapolated as a constant value of 0.8, and 0.9, respectively.

```
*CONTROL_FORMING_INITIAL_THICKNESS
$      PID       LCID      X0        Y0        Z0        VX        VY        VZ
      1       1012    -295.0    -607.0    -43.0     524.0     607.0     0.0
*DEFINE_CURVE
1012
0.0, 0.8
21.0, 0.9
43.0, 1.0
65.0, 1.1
82.0, 1.0
101.0, 0.9
```

In [Figure 12-24](#), a sheet blank is defined with a varying thickness across its surface in a vector direction pointed from the start to end point. The thickness variation as a function of the distance from starting point in section A-A is shown in [Figure 12-25](#).

Revision information:

This feature is available in LS-DYNA starting in Revision 82990.

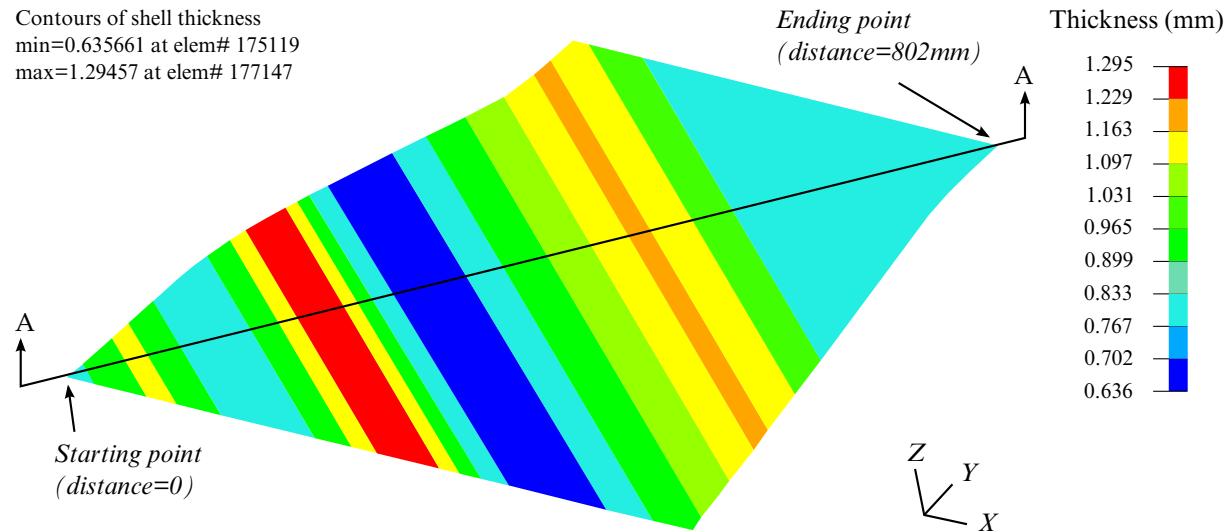


Figure 12-24. Define a varying thickness field across the sheet blank.

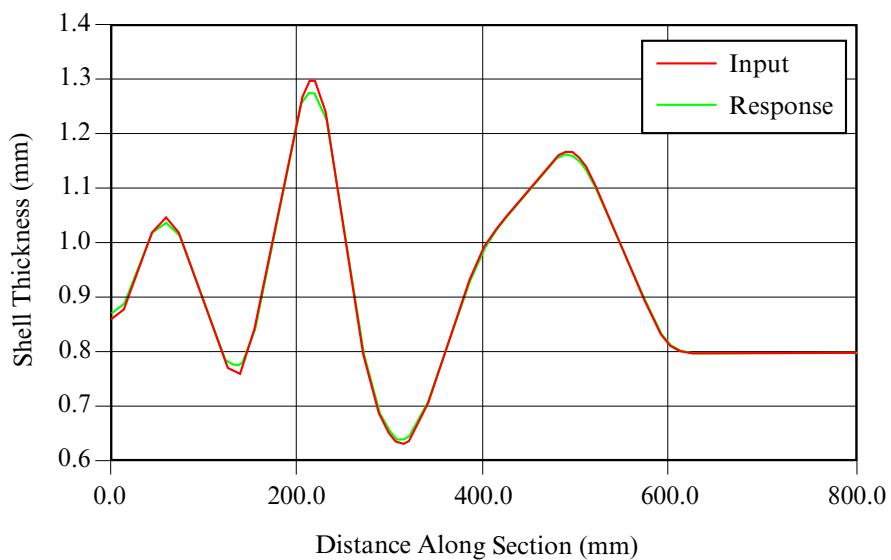


Figure 12-25. Thickness variation across section A-A

*CONTROL

*CONTROL_FORMING_MAXID

*CONTROL_FORMING_MAXID

Purpose: This card sets the node and element ID numbers for an adaptive sheet blank. The new node and element number of the adaptive mesh will start at the values specified on this card, typically greater than the last node and element number of all tools and blanks in the model. This keyword is often used in multi-stage sheet metal forming simulation. The *INCLUDE_AUTO_OFFSET keyword is related.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	MAXIDN	MAXIDE		I2DYNAIN			
Type	I	I	I		I			
Default	none	none	none		0			

VARIABLE

DESCRIPTION

PID	Part ID of the sheet blank, as in *PART.
MAXIDN	Node ID number from which adaptive node ID numbers will be created.
MAXIDE	Element ID number from which adaptive element ID numbers will be created.
I2DYNAIN	Setting I2DYNAIN to 1 will cause this keyword to be output to a dynain file with the updated maximum node and element IDs. This output simplifies post-processing for multi-step processes since it ensures that element and node IDs generated in subsequent steps are larger than those in previous steps. By default, this keyword is not output to a dynain file.

Remarks:

In a multi-stage automatic line die simulation the adaptivity feature may generate node and element IDs that collide with those of the tools used in the later stages of the process. Before the calculation begins, the set of IDs used by the tools is known. By setting MAXIDN to a value greater than the largest tool node ID and MAXIDE to a value greater than the largest tool element ID, it is guaranteed that refinement during the early stages will not lead to conflicts with tool IDs in the later stages.

The following example shows this feature applied in a 2D trimming simulation. Nodes and elements ID numbers generated from an adaptive trim simulation will be larger than the specified ID numbers of 5921980 and 8790292, respectively, for a sheet blank with part ID of 4.

```
*KEYWORD
*INCLUDE_TRIM
sim_trimming.dynain
:
*CONTROL_ADAPTIVE_CURVE
$ IDSET ITYPE N SMIN
  &blkSID 2 2 0.6
*CONTROL_CHECK_SHELL
$ PSID IFAUTO CONVEX ADPT ARATIO ANGLE SMIN
  &blkSID1 1 1 1 0.250000150.000000 0.000000
*INCLUDE
EZtrim.k
-----1-----2-----3-----4-----5-----6-----7-----8
*DEFINE_CURVE_TRIM_NEW
$# tcid tctype tflg tdir tctol tolN nseed1 nseed2
  90914 2 0 1 1.250000 1.000000 0 0
sim_trimming_trimline_01.igs
*DEFINE_VECTOR
$# vid xt yt zt xh yh zh cid
  1 0.000 0.000 0.000 0.000 0.000 1.000000 0
*CONTROL_FORMING_MAXID
$ pid maxidn maxide
  4 5921980 8790292
*END
```

***CONTROL_FORMING_ONESTEP_{OPTION}**

Purpose: This keyword activates a one-step solution using the *total strain theory* approximation to plasticity (also known as deformation theory) to implement an inverse method. Given the *final* geometry, the one-step method uses LS-DYNA's implicit statics solver to compute an approximate solution for (1) the stresses and strains in the formed part, (2) the thickness of the formed part, and (3) the size of the initial blank (unfolded flat blank). This method, which is implemented only for SMP double precision, is useful for estimating the initial blank size with attendant material costs, and for augmenting crashworthiness models to account for metal forming effects, such as plastic strains and blank thickness in crash simulation.

NOTE: The entire input deck must contain a single *fully connected* surface consisting entirely of shells (nothing more nothing less). This surface specifies the *final* geometry. Isogeometric shells *are* supported.

The input may contain more than one *PART provided the parts share common boundary nodes (for instance to model a tailor-welded blank).

Adaptive mesh is not supported. Use *CONTROL_FORMING_REMOVE_ADAPTIVE_CONSTRAINTS if the part is a formed blank with adaptive mesh.

The keyword *DEFINE_FORMING_ONESTEP_PRIMARY can be used to model "patch welded" blanks.

Keywords associated with *CONTROL_FORMING_ONESTEP are:

*CONTROL_FORMING_UNFLANGING

*INTERFACE_BLANKSIZE_DEVELOPMENT

*DEFINE_FORMING_ONESTEP_PRIMARY

Available options include:

<BLANK>

AUTO_CONSTRAINT

DRAWBEAD

FRICTION

TRIA

QUAD

QUAD2 (default)

ORTHO

Summary of keyword options:

1. The AUTO_CONSTRAINT option excludes rigid body motion from the implicit solution by automatically adding nodal constraints. A deck with a *CONTROL_FORMING_ONESTEP card should contain at most one *CONTROL_FORMING_ONESTEP_AUTO_CONSTRIANT card. In addition, starting from Revision 91229, three nodes can be specified on the final part to position the unfolded blank for easier blank nesting, and for blank alignment in forming simulation.
2. The DRAWBEAD option is used to apply draw bead forces in addition to those provided by AUTOBD field in Card 1a.1. A deck containing a *CONTROL_FORMING_ONESTEP card may contain as many *CONTROL_FORMING_ONESTEP_DRAWBEAD cards as there are draw beads to be defined. See [Remark 6](#).
3. The FRICTION option applies friction along the edge of the part based on the binder tonnage input by the user in the BDTON field of Card 1. A deck containing a *CONTROL_FORMING_ONESTEP card may contain as many *CONTROL_FORMING_ONESTEP_FRICTION cards as there are friction node sets to be defined.
4. When one-step forming method was first introduced back in 2011, all quadrilateral elements in the model were split into two triangular elements internally for calculation. As of Revision 112682, this original formulation (with no option) is set as option TRIA. A new option QUAD (Revision 112071) is now available supporting quadrilateral elements with improved algorithm in various areas, which leads to better results. In addition, this option greatly improves calculation speed under multiple CPUs in SMP mode. Another new option QUAD2 improves upon the option QUAD with enhanced element formulation, which further improves results in terms of thinning and plastic strain with slightly longer CPU times. Calculation speed comparisons among the three options can be found in the [Performance Among Options TRIA, QUAD, and QUAD2](#) section. The option QUAD2 is set as a default as of Revision 112682 and is the recommended option.

*CONTROL

*CONTROL_FORMING_ONESTEP

Card Summary:

Card 1a.1. This card is included if the keyword option is unset (<BLANK>), TRIA, QUAD, or QUAD2.

OPTION	TSCLMAX	AUTOBD	TSCLMIN	EPSMAX		LCSDG	DMGEXP
--------	---------	--------	---------	--------	--	-------	--------

Card 1a.2. This card is included if the keyword option is unset (<BLANK>), TRIA, QUAD, or QUAD2.

FLATNAME

Card 1b. This card is included if the AUTO_CONSTRAINT keyword option is used.

ICON	NODE1	NODE2	NODE3				
------	-------	-------	-------	--	--	--	--

Card 1c. This card is included if the DRAWBEAD keyword option is used.

NDSET	LCID	TH	PERCNT				
-------	------	----	--------	--	--	--	--

Card 1d. This card is included if the FRICTION keyword option is used.

NDSET	BDTON	FRICT					
-------	-------	-------	--	--	--	--	--

Card 1e. This card is included if and only if the ORTHO keyword option is used.

PID	NODE1	NODE2					
-----	-------	-------	--	--	--	--	--

Data Card Definitions:

Card 1 for no option (<BLANK>), TRIA, QUAD, and QUAD2.

Card 1a.1	1	2	3	4	5	6	7	8
Variable	OPTION	TSCLMAX	AUTOBD	TSCLMIN	EPSMAX		LCSDG	DMGEXP
Type	I	F	F	F	F		I	F
Default	6	0.0	0.0	0.0	1.0		none	none

Card 2 for no option (<BLANK>), TRIA, QUAD, and QUAD2.

Card 1a.2	1	2	3	4	5	6	7	8
Variable					FLATNAME			
Type					A			
Default					none			

VARIABLE	DESCRIPTION
OPTION	<p>Options to invoke the one-step solution methods which account for undercut conditions in the formed part:</p> <p>EQ.6: One-step solution with unfolded blank (flat) provided by LS-PrePost (see Remark 3). Input to Card 1a.2 is required.</p> <p>EQ.7: One-step solution with blank automatically unfolded in LS-DYNA. Card 1a.2 must be included as a blank line. This option is recommended.</p> <p>L.T.0: If a negative sign precedes any of the above OPTIONS, the stress and strain output in the file <code>onestepresult</code> will be in a large format (E20.0), which leads to more accurate stress results. Card 1a.2 must be included as a blank line.</p>
TSCLMAX	<p>If not zero, it defines a thickness scale factor limiting the maximum thickness in the part. See Effects of TSCLMAX, TSCLMIN and EPS-MAX.</p> <p>For example, if the maximum thickness allowed is 0.8 mm for a blank with initial thickness of 0.75 mm TSCLMAX can be set to 1.0667. All thicknesses that are computed as more than 0.8 mm in the sheet blank will be reset to 0.8 mm. The scale factor is useful in advance feasibility analysis where part design and stamping process have not been finalized and could potentially cause large splits or severe wrinkles during unfolding, rendering the forming results unusable for crash/safety simulation.</p>
AUTOBD	Apply a fraction of a fully locked bead force along the entire periphery of the blank. The fully locked bead force is automatically calculated based on a material hardening curve input. AUTOBD can be increased to easily introduce more thinning and effective plastic strain in the part. See Remark 6 .

*CONTROL

*CONTROL_FORMING_ONESTEP

VARIABLE	DESCRIPTION
	LT.0.0: Turns off the auto-bead feature. EQ.0.0: Automatically applies 30% of fully locked force. GT.0.0: Fraction input will be used to scale the fully locked force.
TSCLMIN	If not zero, defines a thickness scale factor limiting the maximum thickness reduction. See Effects of TSCLMAX, TSCLMIN and EPSMAX . For example, if the minimum thickness allowed is 0.6 mm for a blank with initial thickness of 0.75 mm TSCLMIN can be set to 0.8. All thicknesses that are computed as less than 0.6 mm in the sheet blank will be reset to 0.6 mm. The scale factor is useful in advance feasibility analysis where part design and stamping process have not been finalized and could potentially cause large splits or severe wrinkles during unfolding, rendering the forming results unusable for crash/safety simulation.
EPSMAX	If not zero, it defines the maximum effective plastic strain allowed. All computed effective plastic strains that are greater than this value in the blank will be set to this value. See Effects of TSCLMAX, TSCLMIN and EPSMAX .
LCSDG	Load curve ID defining equivalent plastic strain to failure as a function of stress triaxiality; see *MAT_ADD_EROSION.
DMGEXP	Exponent for nonlinear damage accumulation; see *MAT_ADD_EROSION. Damage accumulation is written as history variable #6 in the file onestepresult.
FLATNAME	File name of the initial unfolded blank by LS-PrePost (see Remark 3). This is needed only for the OPTION = 6. Leave a blank line for OPTION = 7.

CONTROL_FORMING_ONESTEP**CONTROL**

Card 1 for option AUTO_CONSTRAINT.

Card 1b	1	2	3	4	5	6	7	8
Variable	ICON	NODE1	NODE2	NODE3				
Type	I	I	I	I				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
ICON	Automatic nodal constraining option to eliminate the rigid body motion (see Remark 8): EQ.1: Apply
NODE[1,2,3]	Node IDs for which the position is fixed during the unfolding. The position of these nodes in the calculated unfolded piece will coincide with the corresponding nodes in the input. The transformed and unfolded blank will be written in a keyword file repositioned.k . <i>When these fields are undefined, the orientation of the unfolded blank is arbitrary.</i> Starting in Revision118294, the file repositioned.k will always be output regardless if these fields are defined.

Card 1 for option DRAWBEAD.

Card 1c	1	2	3	4	5	6	7	8
Variable	NDSET	LCID	TH	PERCNT				
Type	I	I	F	F				
Default	none	none	0.0	0.0				

VARIABLE	DESCRIPTION
NDSET	Node set ID along the periphery of the part; see *SET_NODE_LIST .
LCID	Load curve ID that defines the material hardening curve.
TH	Thickness of the unformed sheet blank.

*CONTROL

*CONTROL_FORMING_ONESTEP

VARIABLE	DESCRIPTION							
PERCNT	Draw bead lock force fraction of the fully locked bead force.							

Card 1 for option FRICTION.

Card 1d	1	2	3	4	5	6	7	8
Variable	NDSET	BDTON	FRICT					
Type	I	F	F					
Default	none	0.0	0.12					

VARIABLE	DESCRIPTION							
NDSET	Node set ID along the periphery of the part, as defined by keyword *SET_NODE_LIST.							
BDTON	Binder tonnage used to calculate friction force. See Remark 7 .							
FRICT	Coefficient of friction							

Card 1 for option ORTHO, for anisotropic material.

Card 1e	1	2	3	4	5	6	7	8
Variable	PID	NODE1	NODE2					
Type	I	I	I					
Default	none	none	none					

VARIABLE	DESCRIPTION							
PID	Part ID of the final formed blank mesh							
NODE1	First node in the part to define the metal rolling direction							
NODE2	Second node in the part to define the metal rolling direction							

About One-Step Forming Solution:

One-step solution employs the total strain (or deformation) theory of plasticity in place of the more realistic incremental strain (or flow) theory. The total deformation theory expresses stress as a function of total strain; whereas the incremental strain theory requires that LS-DYNA compute a stress update at each time step (strain increment) from the deformation that occurred *during that time step*. Deformation theory results, therefore, do not depend on strain path, forming history, or the details of the stamping process.

When this card is included, the input must contain the *final geometry* from which LS-DYNA calculates the initial flat state using the inverse method. The one-step solution results can get close to the incremental results only when the forming process involves a linear strain path for which the deformation is either monotonically increasing or decreasing. In most cases total strain theory does not match incremental forming.

Path independence leads to several key simplifications:

1. Binder and addendum geometry are not required. There is no need to measure or model these geometries.
2. The solution is independent of stamping die processes (including part tipping).
3. There is no need for contact treatment since there are no tools and dies involved.

The one-step solution is mostly used for advance formability studies in which the user needs to quickly compare a wide range of different design alternatives. With this method the user can evaluate blank size, estimate material cost, and generate a first guess for blank size development (see also *CONTROL_FORMING_UNFLANGING, and *INTERFACE_BLANKSIZE_DEVELOPMENT). This method is also widely used to initialize forming stresses and strains in crash and occupant safety analysis.

Remarks:

1. **Mesh.** In addition to the usual material and physical property definitions, this method requires that the final part be fully meshed using shell elements. *No adaptive mesh is supported.* This mesh should look quite different from a formed blank going through an incremental forming. For example, along the part bend radius, there is no need to build six elements along the arc length as one would do for the punch/die radius in an incremental forming; two elements may be enough. A mesh consisting of uniformly distributed quadrilateral shell elements is ideal. *All elements in the mesh must also have normal consistency.*

With LS-PrePost 4.0, this kind of mesh can be generated using *Mesh → AutoM → Size*. Since this method uses an implicit static solution scheme, the number of elements controls the computational cost; element size has no effect.

Furthermore, to obtain forming results that are similar to the incremental forming results, the part in the one-step input should be similar in size to the final formed blank shape in the incremental forming (before trimming).

Dynain files with *ELEMENT_SHELL_THICKNESS, *INITIAL_STRESS_SHELL and *INITIAL_STRAIN_SHELL are not supported.

2. **Holes.** Any trimmed-out holes can be filled (but not necessary). The filling can be done semi-automatically using LS-PrePost 4.0 by selecting *Mesh* → *EleGen* → *Shell* → *Shell by Fill_Holes* → *Auto Fill*. The filled area of the part can be saved in a different part, as multiple parts (PID) are allowed. The forming results will be more realistic with holes are filled, or left alone, based on the actual stamping process intent.
3. **Unfolding.** For OPTION = 6, the unfolded blank can be obtained from LS-PrePost using *EleTol* → *Morph* → *Type = Mesh_Unfolding* → *Unfold*. The unfolded mesh can be saved as a keyword file and used as input (see the FLATNAME field in Card 1a.2). With OPTION = 7, LS-DYNA unfolds the mesh itself.
4. **Element formulation.** Shell elements of type 2 and 16 are supported. Since this feature uses the implicit method, type 16 is more convergent and computationally efficient than type 2; therefore, it is strongly recommended. Results are output on all integration points, as seen in the ELMFORM and NIP variables in *SECTION_SHELL.
5. **Supported materials.** *MAT_024, *MAT_036, *MAT_037, *MAT_112, *MAT_114, *MAT_123, *MAT_133, and *MAT_238 are supported. The user *must* provide a material hardening curve. For *MAT_024 the hardening curve must be specified with the LCSS field. It can be input as a table. Strain rate, however, is ignored for this material, even if the variables C and P are provided. For MAT_036 the hardening curve must be a load curve (HR = 3), and for *MAT_037 it must be specified with HLCID.

For anisotropic material, the option ORTHO can be used to define the material rolling direction. Since there is no flat blank existing as an input, the only way to define the rolling direction is to use two existing nodes in the 3D formed part to meet the requirement.

6. **Boundary conditions.** The primary boundary/loading condition for the one-step solution is the draw bead forces, which are set with the AUTOBD field or with the DRAWBEAD keyword option.
 - a) With the DRAWBEAD option, draw bead forces are applied on a user defined node set (see NDSET). A fraction of the full lock force, determined by the tensile strength and sheet thickness, can be specified. The larger the

fraction, the less the metal will flow into the die resulting in more stretching and thinning.

- b) Boundary conditions may also be set using the auto-beads feature (see the AUTOBD field) with which draw bead forces are automatically applied to all nodes along the part boundary. The users must specify the fraction of the fully locked bead force to be applied. The default value of 30% is sufficient for crash/occupant safety applications.

The last important, but often overlooked, boundary condition is the part's shape. For example, an oil pan with a larger flange area will experience greater thinning in the part wall, whereas having a smaller flange area will have the reverse effect. To obtain results that are closer to the incremental strain theory, additional materials may need to be added to the final part geometry in cases where the sheet blank is not fully developed, meaning no trimming is required to finish the part.

7. **Friction.** Friction effects can be included with the FRICTION option. The frictional force is based on an expected binder tonnage, and is a percentage of the input force. Note that the binder tonnage value (see BDTON) is used *exclusively* in calculating friction forces. The binder tonnage is not actually applied on the binder as a boundary condition.
8. **Rigid body motion.** LS-DYNA will automatically add nodal constraints to prevent rigid body motion when the AUTO_CONSTRAINT option is used and ICON is set to 1.
9. **Implicit solver options.** All other implicit cards, such as *CONTROL_IMPLICIT_GENERAL, *CONTROL_IMPLICIT SOLUTION, *CONTROL_IMPLICIT_SOLVER, *CONTROL_IMPLICIT_AUTO, *CONTROL_IMPLICIT_TERMINATION, etc., are used to set the convergence tolerance, termination criterion, etc. The two most important variables controlling the solution convergence are DELTAU from *CONTROL_IMPLICIT_TERMINATION, and DCTOL from *CONTROL_IMPLICIT_SOULTION. Experience has shown that they should be set to 0.001 and 0.01, respectively, to obtain the most efficient solution with the best results. Typically, four implicit steps are sufficient, and DT0 in *CONTROL_IMPLICIT_GENERAL and ENDTIM in *CONTROL_TERMINATION should be set accordingly. For difficult parts, more steps maybe needed. For some parts, ILIMIT in *CONTROL_IMPLICIT_SOLUTION may need to be set to 1 for the full Newton iteration.
10. **Results.** Results are stored in an ASCII file named onestepresult using the dynain format. This file contains the forming thickness, the stress and the strain fields on the final part. It can be plotted with LS-PrePost. One quick and useful LS-PrePost plotting feature is the formability contour map, which colors the model to highlight various forming characteristics including cracks, severe

thinning, wrinkles, and good surfaces. The formability map feature is in *Post → FLD → Formability*.

The variable PSID in *INTERFACE_SPRINGBACK_LSDYNA can be used to control the output to onestepresult and dynain files, starting in Dev 138146.

11. **Final estimated blank size.** Additionally, the final estimated blank size in its initial, flat state is stored in the d3plot files. The d3plot files also contain intermediate shapes from each implicit step. The final blank mesh in its flat state can be written to a keyword file using LS-PrePost by the following steps:

- a) Go to *Post → Output → Keyword*,
- b) Check the box to include *Element* and *Nodal Coordinates*
- c) Move the animation bar to the last state, and,
- d) Click on *Curr* and *Write*.

In addition, blank outlines can be created by:

- e) Menu option *Curve → Spline → From Mesh (Method)*,
- f) Checking *Piecewise → byPart*,
- g) Select the blank,
- h) Click on *Apply*, and,
- i) Finally, save the curves in IGES format using the *File* menu at the upper left corner.

The unformed blank mesh will be output into the file **repositioned.k**.

Effects of TSCLMAX, TSCLMIN and EPSMAX:

During the early stage of product design, the initial product specifications may lead to large strains and excessive thinning or thickening on the formed panel. The ensuing one-step results may not be suitable to be used in a crashworthiness simulation. However, these kinds of forming issues are certain to be fixed as a natural part of the design and stamping engineering process. The variables TSCLMAX, TSCLMIN and EPSMAX, thus, impose artificial limits on the excessive thickening (indication of wrinkles), excessive thinning, and plastic strains, respectively. These fields provide a convenient way to run a crash simulation with approximate and reasonable forming effects before the design is finalized. In the keyword below (which is a part of the firewall model with original thickness of 0.75 mm), TSCLMIN and EPSMAX are set to 0.8 and 0.3, respectively.

```
*CONTROL_FORMING_ONESTEP
$   OPTION      TSCLMAX      AUTODB      TSCLMIN      EPSMAX
    7           0.5          0.8          0.3
```

The thickness and effective plastic strain plots for the firewall model are shown in [Figures 12-30](#) and [12-31](#), respectively. The minimum value in the thickness contour plot and maximum value in the plastic strain contour plot as shown in the upper left corner correspond to the values specified in TSCLMIN and EPSMAX, respectively.

Similarly, TSCLMAX can be set to 1.0667 to limit the max thickening in the part to 0.8 mm:

```
*CONTROL_FORMING_ONESTEP
$   OPTION      TSCLMAX      AUTODB      TSCLMIN      EPSMAX
    7       1.0667        0.5          0.8          0.3
```

Repositioning the Unfolded Flat Blank:

Often the input to a one-step simulation is the final product part in a coordinate system that is useful for subsequent simulations. However, after the one-step simulation, the unfolded flat blank will be in a different orientation and position, requiring manual repositioning of the blank to its desired orientation and position. The variables NODE1, NODE2, and NODE3 allow users to specify three nodes so that the blank is transformed onto the final part (the input), superimposing the exact same three nodes in both parts. In the example shown in [Figure 12-33](#), the three nodes (Nodes 197, 210 and 171) are defined near the edges of two holes. The transformed and unfolded flat blank (written to a keyword file **repositioned.k**) is seen superimposed onto the final part according to the three nodes specified ([Figure 12-33 bottom](#)). If these nodes are not defined, the simulation will result in the unfolded flat blank in a state shown in [Figure 12-33 \(top\)](#), which is undesirable to most users.

Damage Accumulation:

Damage accumulation D is calculated based on (see ***MAT_ADD_EROSION**):

$$D = \left(\frac{\varepsilon_p}{\varepsilon_f} \right)^{\text{DMGEXP}},$$

where ε_p is the equivalent plastic strain and ε_f is the equivalent plastic strain to failure.

In the example below, load curve 500 (LCSDG) gives plastic failure strain as a function stress triaxiality and DMGEXP is assumed to be 1.254. Since the damage accumulation is written into the file **onestepresult** as history variable #6, the variable NEIP in ***DATABASE_EXTENT_BINARY** should be set to at least 6.

```
*CONTROL_FORMING_ONESTEP
```

*CONTROL

*CONTROL_FORMING_ONESTEP

```
$      OPTION          AUTODB    TSCLMIN    EPSMAX          LCSDG    DMGEXP
        7                  0.8       0.3
*DEFINE_CURVE
500
-0.3,0.6
-0.2,0.3
0.0,0.2
0.2,0.25
0.4,0.46
0.65,0.28
0.9,0.18
*DATABSE_EXTENT_BINARY
$      NEIPH      NEIPS    MAXINT    STRFLG    SIGFLG    EPSFLG    RLTFLG    ENGFLG
        6          7       1
$      CMPFLG    IEVERP   BEAMIP   DCOMP     SHGE      STSSZ
        1          2
```

The damage accumulation contour map from the file onestepresult can be plotted in LS-PrePost.

Effects of Unfilled Holes on Forming Results:

In Figure 12-32, a thickness contour plot of a one-step calculation on the NCAC Taurus firewall model with its holes unfilled is shown. The unfilled case will undergo slightly less thinning, since the holes will expand as material flows outward away from the hole. However, the thicknesses with holes filled are likely closer to reality, since the holes are mostly filled during forming on the draw panel and then trimmed off afterwards in a trim process. On the other hand, it is important to realize that not all the holes are filled in a draw panel. Some holes are cut inside the part in the scrap area (but not all the way to the trim line) during the draw process to allow material to flow into areas that are difficult to form which avoids splitting.

Example:

The following example provides a partial input file with typical control cards. It will iterate for four steps, with auto beads of 0% lock force applied around the part boundary, and with automatic nodal constraints.

```
*CONTROL_TERMINATION
$      ENDTIM
        1.0
*CONTROL_IMPLICIT_GENERAL
$      IMFLAG      DT0
        1      0.25
*CONTROL_FORMING_ONESTEP
$      OPTION          AUTODB
        7
*CONTROL_FORMING_ONESTEP_AUTO_CONSTRAINT
$      ICON
        1
*CONTROL_IMPLICIT_TERMINATION
$      DELTAU
        0.001
*CONTROL_IMPLICIT_SOLUTION
```

```

$   NSLOLVR      ILIMIT      MAXREF      DCTOL      ECTOL
      2           11         1200       0.01       1.00
*CONTROL_IMPLICIT_SOLVER
$   LSOLVR
      5
*CONTROL_IMPLICIT_AUTO
$   IAUTO      ITEOPT      ITEWIN      DTMIN      DTMAX
      0           0           0         0.0        0.0

```

Additional cards below specify extra bead forces of 45% and 30% applied to node sets 22 and 23 along the part periphery, respectively. Also, the friction forces are applied to the same node sets with a friction coefficient of 0.1 and a binder tonnage of 10000.0 N.

```

*CONTROL_FORMING_ONESTEP_DRAWBEAD
$   NDSET      LCID      TH      PERCNT
      22        200      1.6      0.45
*CONTROL_FORMING_ONESTEP_DRAWBEAD
      23        200      1.6      0.30
*CONTROL_FORMING_ONESTEP_FRICTION
$   NDSET      BDTON      FRICT
      22     10000.0      0.1
*CONTROL_FORMING_ONESTEP_FRICTION
$   NDSET      BDTON      FRICT
      23     10000.0      0.1

```

The one-step forming results for the NCAC Taurus model's firewall are shown in [Figure 12-26](#). The average element size across the blank is 8 mm, and the trimmed part (with holes filled) consists of 15490 elements. *MAT_24 was used with BH210 material properties. On a 1 CPU Xeon E5520 Linux machine, it took 4 minutes to complete the run with a total of four steps. The thickness, the plastic strain, and the blank size prediction were reasonable, as shown in [Figures 12-27, 12-28](#) and [12-29](#).

Performance Among Options TRIA, QUAD and QUAD2:

The following partial keyword input is an example of using the option QUAD. Note the draw bead force parameter AUTOBD is set at 0.5. Calculation speed comparison among options QUAD, QUAD2 and TRIA can be found in [Table 12-1](#).

```

*KEYWORD
*include
model.k
*CONTROL_TERMINATION
1.0
*CONTROL_FORMING_ONESTEP_QUAD
$# option maxthick    autobd    thinmin    epsmax
      7          0.5

*CONTROL_FORMING_ONESTEP_AUTO_CONSTRAINT
      1
*CONTROL_IMPLICIT_GENERAL
$# imflag      dt0      imform      nsbs      igs      cnstn      form      zero_v
      1     0.2500      2           1           0           0           0           0
*CONTROL_IMPLICIT_TERMINATION
$# deltau      delta1      ketol      ietol      tetol      nstep
      0.001000      0.000      0.000      0.000      0.000           0
*CONTROL_IMPLICIT_NONLINEAR
$# nsolvr      ilimit      maxref      dctol      ectol      not used    lstol      rssf
      12           11         200     0.010000     0.100000      0.000      0.000      0.000

```

*CONTROL

*CONTROL_FORMING_ONESTEP

```
$# dnorm      diverg      istif      nlprint
      0          0          0          2
$# arcctl      armdir      arclen      arcmth      arcdmp
      0          0        0.000       1          2
*CONTROL_IMPLICIT_SOLVER
5
*PART
      5000000    5000000    5000000
*SECTION_SHELL
  5000000      16       1.        5.        1.
  0.72        0.72      0.72      0.72
```

	Number of elements	Calculation speed (D.P. SMP Rev.112720, 8 CPUs)		
		Option TRIA	Option QUAD	Option QUAD2
A hat shaped part	71000	21.0 min	14.1 min	16.6 min
An upper dash panel	61700	24.5 min	11.5 min	17.2 min

Table 12-1. Calculation speed for various options

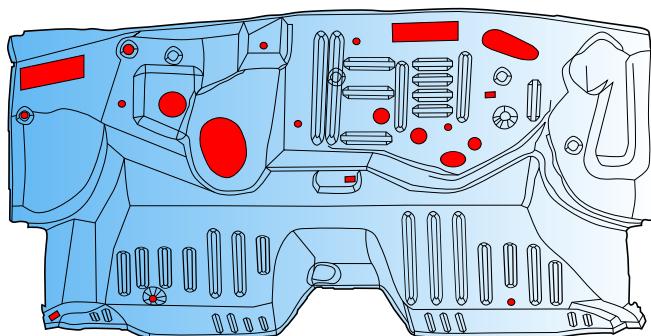


Figure 12-26. A trimmed dash panel (firewall) with holes auto-filled using LS PrePost 4.0 (original model courtesy of NCAC Taurus crash model).

Contours of shell thickness
min=0.478084, at elem# 3210698
max=1.10908, at elem# 3211511

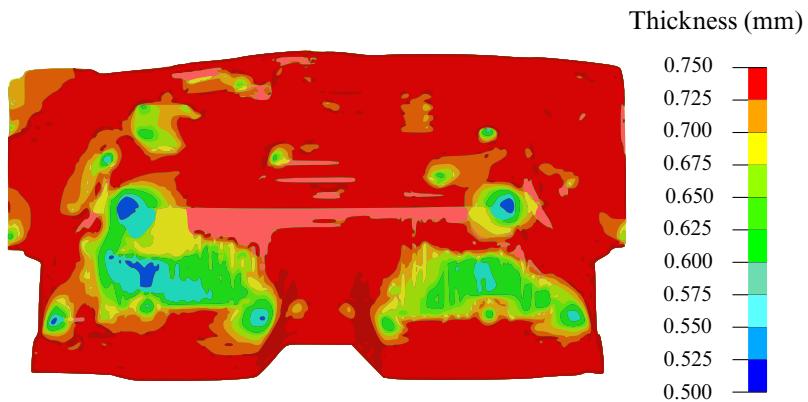


Figure 12-27. Shell thickness prediction ($t_0 = 0.75$ mm).

Contours of plastic strain
max ipt. value
min=0, at elem# 3008783
max=0.46, at elem# 3210698

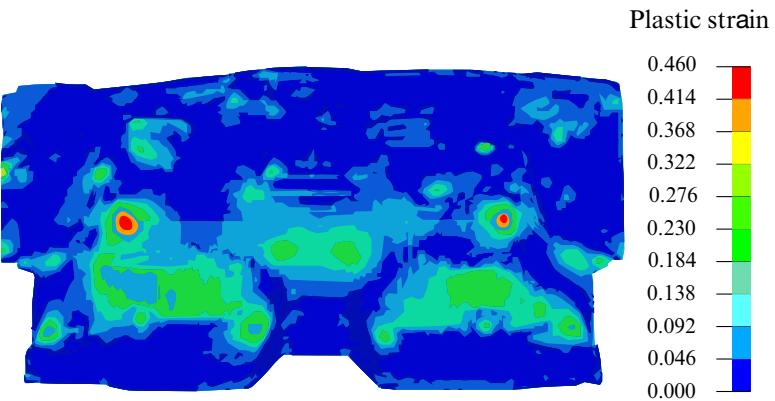


Figure 12-28. Effective plastic strain prediction.

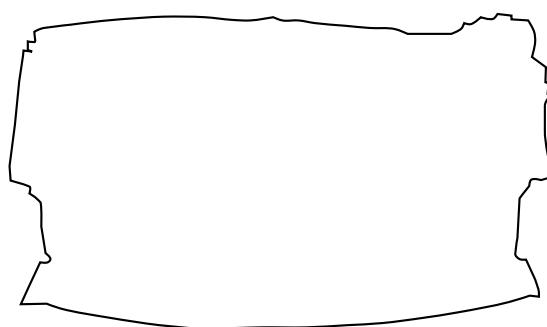


Figure 12-29. Initial blank size prediction (flat, not to scale).

*CONTROL

*CONTROL_FORMING_ONESTEP

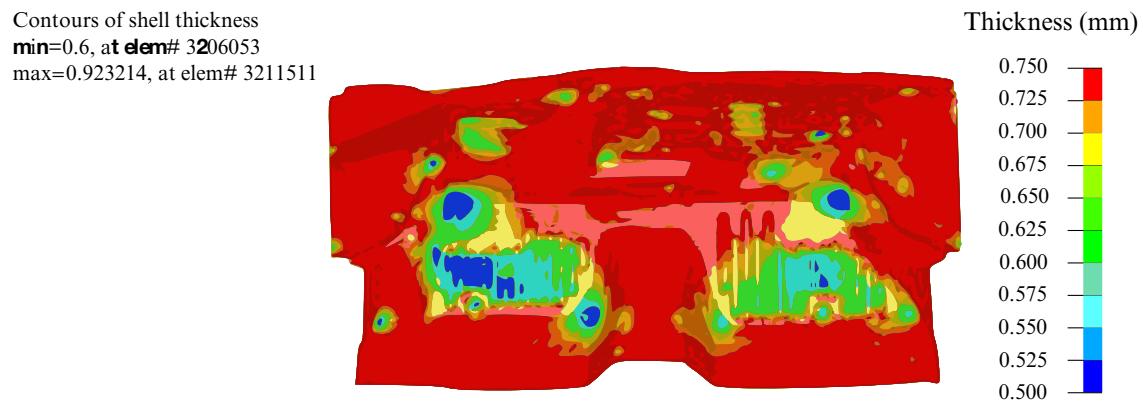


Figure 12-30. Blank thickness prediction with TSCLMIN = 0.8.

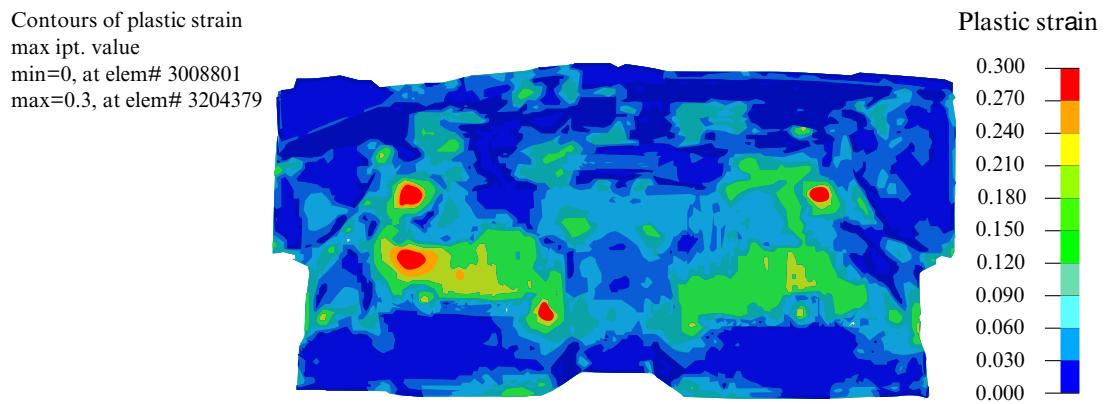


Figure 12-31. Effective plastic strain with EPSMAX = 0.3.

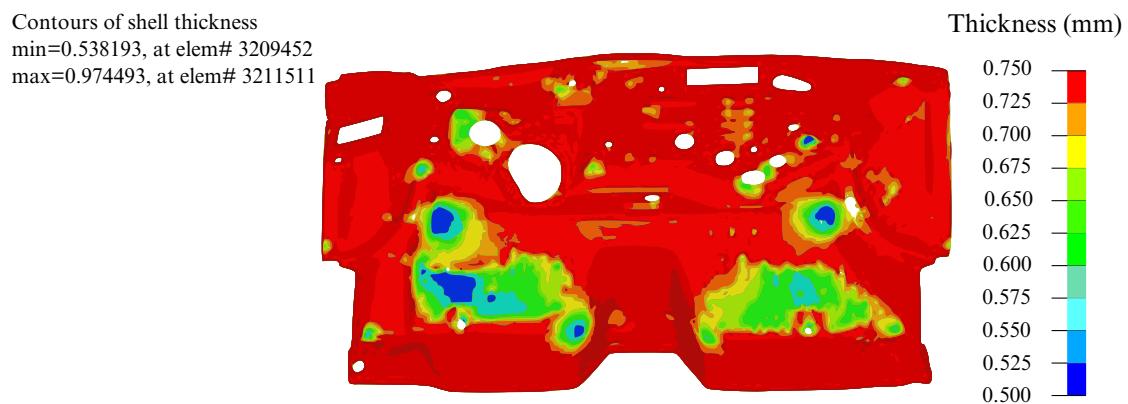


Figure 12-32. Blank thickness with trimmed holes ($t_0 = 0.75$ mm).

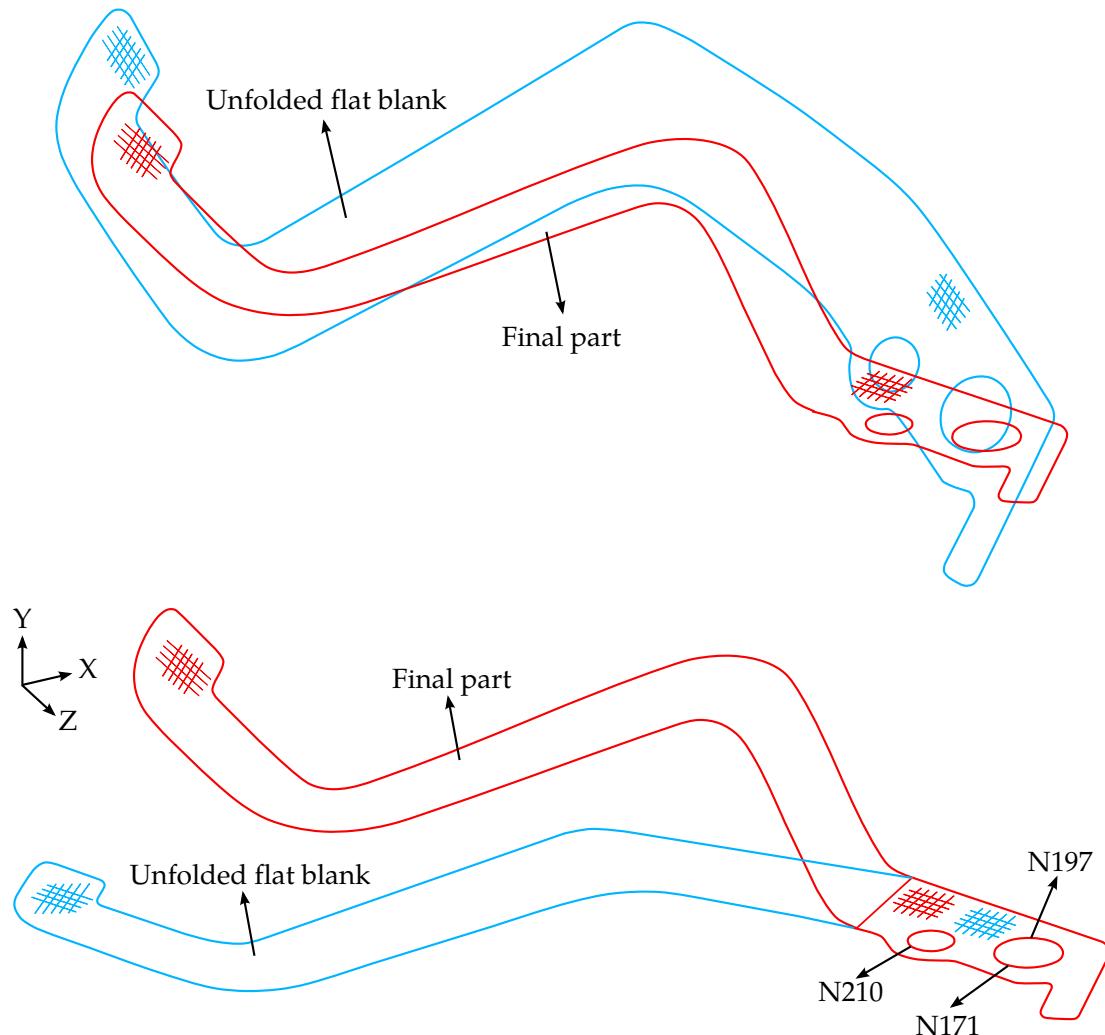


Figure 12-33. An example of the results when using the NODE1, NODE2, and NODE3 feature (bottom) and without using the feature (top), courtesy of Kainet Technologies Pvt Ltd, India.

*CONTROL

*CONTROL_FORMING_OUTPUT

*CONTROL_FORMING_OUTPUT_{OPTION}

Available options include:

<BLANK>

INTFOR

Purpose: This card defines the times at which states are written to the d3plot and intfor files based on the tooling's distances from the home (final) position. When the INTFOR option is set, this keyword card controls when states are written to the intfor file, otherwise it controls the d3plot file. This feature may be combined with parameterized input and/or automatic positioning of the stamping tools using the *CONTROL_FORMING_AUTOPOSITION_PARAMETER card.

NOTE: When this card is present, no states are written except for those specified on this card. This card supersedes the *DATABASE_BINARY_D3PLOT card.

Forming Output Cards. Repeat as many times as needed to define additional outputs in separate tooling kinematics curves. The next keyword ("**") card terminates the input.

Card 1	1	2	3	4	5	6	7	8
Variable	CID	NOUT	TBEG	TEND	Y1/LCID	Y2/CIDT	Y3	Y4
Type	I	I	F	F	F/I	F/I	F	F
Default	none	0	0.0	none	Rem 3	Rem 3	Rem 3	Rem 3

VARIABLE	DESCRIPTION
CID	ID of a tooling kinematics curve. This curve is integrated so that the specified output distances can be mapped to times. For correct distance-to-time mapping CID must be applied to the tool of interest using a *BOUNDARY_PRESCRIBED_MOTION_RIGID card. The ordinate scale factor SFO in the *DEFINE_CURVE is supported in this keyword starting from Dev Revision 82755.
NOUT	Total number of states written to the d3plot or intfor databases for the tooling kinematics curve, CID, excluding the beginning and final states. If NOUT is larger than the number of states specified by either LCID or Y_i fields (5 through 8), the remaining states are

VARIABLE	DESCRIPTION
	evenly distributed between TBEG and the time corresponding to the biggest Y_i from the home position, as shown in Figure 12-34 . If NOUT is left as blank or as “0”, the total number of output states will be determined by either LCID or Y_i ’s.
	Starting in Dev Revision 124051, NOUT works alone, without the need to define LCID or Y_i ’s.
TBEG	Start time of the curve. This time should be consistent with the BIRTH in *BOUNDARY_PRESCRIBED_MOTION_RIGID.
TEND	End time of the curve. This time should be consistent with the DEATH in *BOUNDARY_PRESCRIBED_MOTION_RIGID. This time is automatically reset backward removing any idling time if the tool finishes traveling early, so output distances can start from the reset time. A state is written at TEND.
Y1/LCID, Y2, Y3, Y4	<p>Y1/LCID.GT.0: All four variables (Y_1, Y_2, Y_3, Y_4) are taken to be the distances from the punch home, where d3plot files will be output.</p> <p>Y1/LCID.LT.0: The absolute value of Y1/LCID (must be an integer) is taken as a load curve ID (see *DEFINE_CURVE). Only the abscissas in the load curve, which are the distances to punch home, are used. These distances specify the states that are written to the d3plot files. Ordinates of the curve are ignored. This case accommodates more states than is possible with the four variables Y_1, Y_2, Y_3, Y_4. Furthermore, when $Y_1/LCID < 0$, Y_2, Y_3, and Y_4 are ignored.</p> <p>Available starting from Dev Revision 112604, the output will be skipped for any negative abscissa in the load curve. Note a curve with <i>only</i> negative abscissas is not allowed.</p>
Y2/CIDT	<p>Y2/CIDT.GT.0: The input is taken as the distance from the punch home, where a d3plot file will be output.</p> <p>Y2/CIDT.LT.0: The absolute value of Y2/CIDT (must be an integer) is taken as a load curve ID (see *DEFINE_CURVE). Only the abscissas in the load curve, which are the simulation times, are used. These times specify the states that are written to the d3plot files. Ordinates of the curve are ignored.</p>

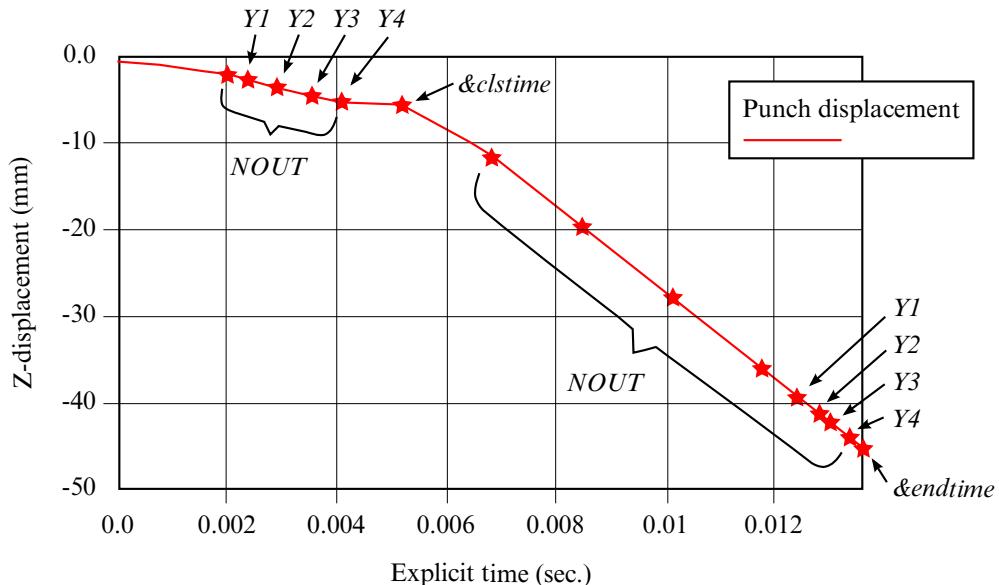


Figure 12-34. An output example for closing and drawing. See the example provided at the end of this section.

VARIABLE**DESCRIPTION**

Note this time-dependent load curve will output additional d3plot files on top of the d3plot files already written in case $Y1/LCID < 0$ (if specified). Furthermore, when $Y2/CIDT < 0$, $Y3$ and $Y4$ are ignored. See the [example Using CIDT](#) below.

Motivation:

In stamping simulations not all time steps are of equal interest. This feature allows you to save special states, usually those for which wrinkling and thinning conditions arise as the punch approaches its home position.

Remarks:

- Related Keywords.** Keywords *DATABASE_BINARY_D3PLOT and *DATABASE_BINARY_INTFOR are not required (ignored if present) to output d3plot and intfor files when this keyword is present.
- CID.** *CONTROL_FORMING_OUTPUT and *CONTROL_FORMING_OUTPUT_INTFOR can share the same CIDs.
- Output Distribution.** The following are rules for output distribution:

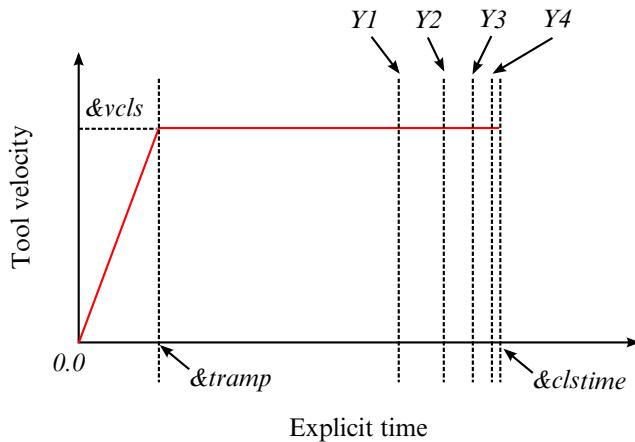


Figure 12-35. Specifying d3plot/intfor output at specific distances to punch home.

- a) If columns 5 through 8 are left blank, output (NOUT) will be evenly distributed through the travel.
- b) The variable NOUT has priority over the number of points on the LCID.
- c) Distances input (in LCID) that are greater than the actual tool travel will be ignored.
- d) Distance input (in LCID) does not necessarily have to be in a descending or ascending order.

Applicability:

Velocity must be the prescribed motion in *BOUNDARY_PRESCRIBED_MOTION_RIGID (VAD = 0) for this keyword to apply. This keyword only works for explicit dynamics. Tooling kinematics profiles of various trapezoids (including right trapezoid) are all supported. Local coordinate systems are supported.

Air Draw Example:

In the sample keyword input below (air draw, referring to [Figures 12-34](#) and [12-36](#)), a total of five states will be output during a binder closing. The kinematics are specified by the curve of ID 1113, which defines tooling kinematics starting time 0.0 and ending at time $\&clstime$.

Curve 1113 is used to associate the specified distances to the appropriate time step. In this example NOUT is set to 5. Of these five output states the last four will be output at upper die distance to closing of 3.0, 2.0, 1.0, and 0.5 mm according to the values specified in the Y_1 , Y_2 , Y_3 , and Y_4 fields.

*CONTROL

*CONTROL_FORMING_OUTPUT

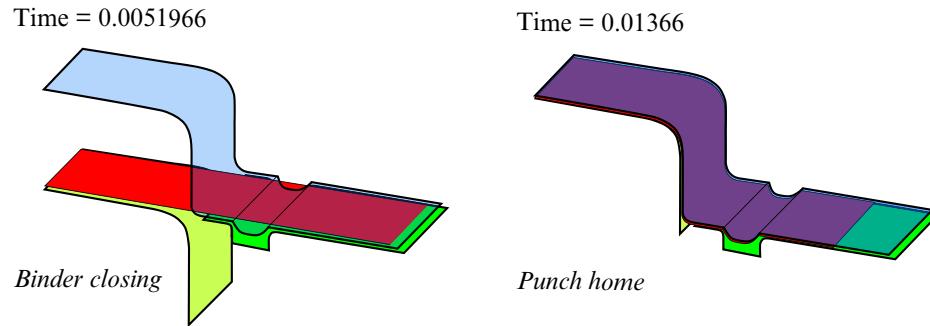


Figure 12-36. An air draw example with closing and drawing.

Similarly, a total of eight states will be written to the d3plot file made during draw forming according curve ID 1115, which defines tooling kinematics starting at time &clstime, and ending at time &endtime. Of the eight states the last four will be output at punch distance to draw home of 6.0, 4.0, 3.0, and 1.0 mm; the remaining four outputs will be evenly distributed between starting punch distance to home and punch distance of 6.0mm to home.

Likewise, for intfor, 15 states will be written before closing and 18 states after the closing. The d3plot and intfor files will always be output for the first and last states as a default; and at where the two curves meet at &clstime, only one d3plot and intfor will be output.

To output intfor, “S=*filename*” needs to be specified on the command line, and SAPR and SBPR need to be set to “1” on the *CONTACT_... cards.

```
$-----1-----2-----3-----4-----5-----6-----7-----8
*CONTROL_FORMING_OUTPUT
$    CID      NOUT      TBEG      TEND      y1      y2      y3      y4
      1113      5          &clstime      3.0      2.0      1.0      0.5
      1115      8          &clstime      6.0      4.0      3.0      1.0
*CONTROL_FORMING_OUTPUT_INTFOR
$    CID      NOUT      TBEG      TEND      y1      y2      y3      y4
      1113     15          &clstime      3.5      2.1      1.3      0.7
      1115     18          &clstime      16.0     4.4      2.1      1.3
*BOUNDARY_PRESCRIBED_MOTION_RIGID
$    typeID      DOF      VAD      LCID      SF      VID      DEATH      BIRTH
&udiepid      3          0      1113     -1.0      0      &clstime      0.0
&bindpid      3          0      1114      1.0      0      &clstime      0.0
&udiepid      3          0      1115     -1.0      0      &endtime      &clstime
&bindpid      3          0      1115     -1.0      0      &endtime      &clstime
$-----1-----2-----3-----4-----5-----6-----7-----8
*DEFINE_CURVE
1113
0.0,0.0
&clsramp,&vcls
&clstime,0.0
*DEFINE_CURVE
1114
0.0,0.0
10.0,0.0
*DEFINE_CURVE
1115
0.0,0.0
&drwramp,&vdraw
```

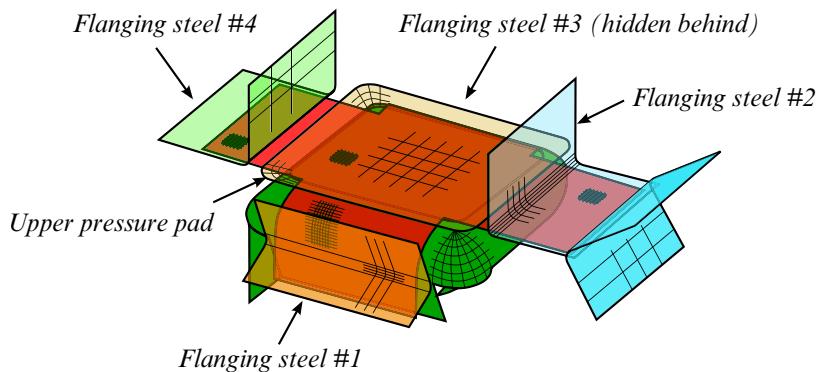


Figure 12-37. An example of multiple flanging process.

&drwtime, &vdraw

The keyword example below illustrates the use of load curves 3213 and 3124 to specify the states written to the d3plot and intfor files respectively. In addition to the eight states specified by curve 3213, five additional outputs will be generated. Similarly, in addition to the 10 intfor states defined by curve 3214, eight additional states will be output.

```
*CONTROL_FORMING_OUTPUT
$      CID      NOUT      TBEG      TEND      y1      y2      y3      y4
      1113      13          &clstime   -3213
*CONTROL_FORMING_OUTPUT_INTFOR
$      CID      NOUT      TBEG      TEND      y1      y2      y3      y4
      1113      18          &clstime   -3214
*DEFINE_CURVE
3213
88.0
63.0
42.0
21.5
9.8
5.2
3.1
1.0
*DEFINE_CURVE
3214
74.0
68.0
53.0
32.0
25.5
7.8
4.2
2.1
1.4
0.7
```

Multiple Flanging Process Example:

Referring to [Figure 12-37](#) and a partial keyword example listed below, flanging steels #1 through #4 are defined as parameters &flg1pid through &flg4pid, respectively, which are moving in their own local coordinate systems. The termination time &endtime is defined as pad closing time, &clstime, plus the maximum travel time of all four flanging

*CONTROL

*CONTROL_FORMING_OUTPUT

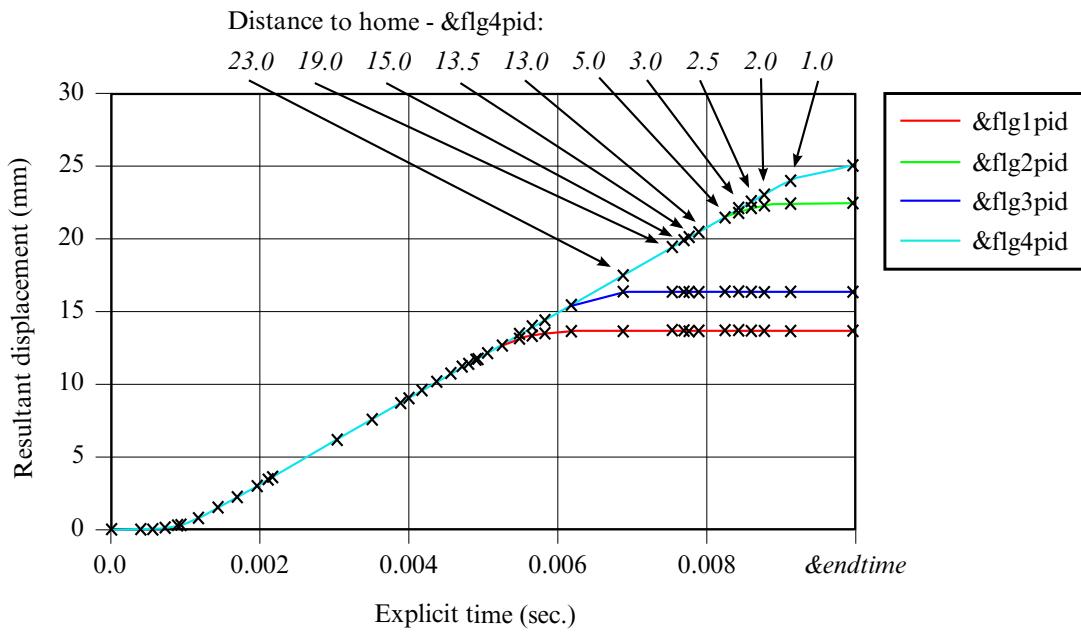


Figure 12-38. D3PLOT/INTFOR output in case of multiple flanging process.

steels. A total of ten d3plot states and ten intfor states are defined for each flanging steel using curve IDs 980 and 981, respectively. Curve values outside of the last 10 states (distances) are ignored; and reversed points are automatically adjusted.

In Figure 12-38, locations of d3plot states are indicated by "x" markers for each flanging steel move. Note that for flanging steels with longer travel distances, there may be additional d3plot states between the defined points, controlled by distance output defined for other flanging steels with shorter travals. The total number of d3plot (and intfor) states is the sum of all nout defined for each flanging steel so care should be taken to limit the total d3plot (and intfor) states, especially if large number of flanging steels are present.

```
*KEYWORD
$ -----closing
*BOUNDARY_PRESCRIBED_MOTION_RIGID
$ typeID      DOF      VAD      LCID      SF      VID      DEATH      BIRTH
&upid1       3         0        1113    &padvdir     0      &clstime
*BOUNDARY_PRESCRIBED_MOTION_RIGID_local
  &flg1pid    3         0        1114    1.0      0      &clstime
  &flg2pid    3         0        1114    1.0      0      &clstime
  &flg3pid    3         0        1114    1.0      0      &clstime
  &flg4pid    3         0        1114    1.0      0      &clstime
$ -----flanging
*BOUNDARY_PRESCRIBED_MOTION_RIGID
$ typeID      DOF      VAD      LCID      SF      VID      DEATH      BIRTH
&upid1       3         0        1115    &padvdir     0      &clstime
*BOUNDARY_PRESCRIBED_MOTION_RIGID_local
  &flg1pid    3         0        1116    1.0      0      &clstime
  &flg2pid    3         0        1117    1.0      0      &clstime
  &flg3pid    3         0        1118    1.0      0      &clstime
  &flg4pid    3         0        1119    1.0      0      &clstime
-----1-----2-----3-----4-----5-----6-----7-----8
*DEFINE_CURVE
  1116
    0.0          0.0
```

```
&tdrwup          &vdrw
&tdown1          &vdrw
&drw1tim        0.0
1.0E+20          0.0

*DEFINE_CURVE
1117
0.0              0.0
&tdrwup          &vdrw
&tdown2          &vdrw
&drw2tim        0.0
1.0E+20          0.0

*DEFINE_CURVE
1118
0.0              0.0
&tdrwup          &vdrw
&tdown3          &vdrw
&drw3tim        0.0
1.0E+20          0.0

*DEFINE_CURVE
1119
0.0              0.0
&tdrwup          &vdrw
&tdown4          &vdrw
&drw4tim        0.0
1.0E+20          0.0

$-----1-----2-----3-----4-----5-----6-----7-----8
*DEFINE_CURVE
980
60.0
55.0
42.0
40.0
38.0
31.0
23.0
19.0
15.0
13.0
13.5
5.0
3.0
2.0
2.5
1.0

*DEFINE_CURVE
981
23.0
19.0
15.0
13.0
13.5
:
*CONTROL_FORMING_OUTPUT
$ -----1-----2-----3-----4-----5-----6-----7-----8
$     CID      NOUT      TBEG      TEND    Y1/LCID
  1116      10  &clstime  &endtime   -980
  1117      10  &clstime  &endtime   -980
  1118      10  &clstime  &endtime   -980
  1119      10  &clstime  &endtime   -980

*CONTROL_FORMING_OUTPUT_INTFOR
$ -----1-----2-----3-----4-----5-----6-----7-----8
$     CID      NOUT      TBEG      TEND    Y1/LCID
  1116      10  &clstime  &endtime   -981
  1117      10  &clstime  &endtime   -981
  1118      10  &clstime  &endtime   -981
  1119      10  &clstime  &endtime   -981
```

: : : :

Using CIDT Example:

The example below shows in addition to the 7 states output based on various distances from punch home, defined by load curve 980, 4 more states are output based on simulation time, defined by load curve 999.

```
$---+---1---+---2---+---3---+---4---+---5---+---6---+---7---+---8
*DEFINE_CURVE
999
1.0e-03
2.0e-03
3.0e-03
4.0e-03
*DEFINE_CURVE
980
13.5,0.0
13.0,0.0
5.0,0.0
3.0,0.0
2.5,0.0
2.0,0.0
1.0,0.0
*CONTROL_FORMING_OUTPUT
$-----1-----2-----3-----4-----5-----6-----7-----8
$      CID      NOUT      TBEG      TEND      Y1/LCID      Y2/CIDT
    1116      0  &clstime  &endtime      -980      -999
    1117      0  &clstime  &endtime      -980      -999
    1118      0  &clstime  &endtime      -980      -999
    1119      0  &clstime  &endtime      -980      -999
```

***CONTROL_FORMING_PARAMETER_READ**

Purpose: This feature allows for reading of a numerical number from an existing file to store in a defined parameter. The parameter can be used and referenced in the current simulation. The file to be read may be a result from a previous simulation. The file may also simply contain a list of numbers defined beforehand and to be used for the current simulation.

Card 1	1	2	3	4	5	6	7	8
Variable				FILENAME				
Type				C				

Parameter Cards. Include one card for each parameter. The next keyword ("*") card terminates the input.

Card 2	1	2	3	4	5	6	7	8
Variable	PARNAME	METHOD	LINE #	BEGIN	END			
Type	C	I	I	I	I			
Default	none	0	0	0	0			

VARIABLE	DESCRIPTION
FILENAME	Name of the file to be read.
PARNAME	Parameter name. Maximum character length is 7.
METHOD	Read instruction: EQ.1: read, follow definition by LINE #, BEGIN and END definition
LINE #	Line number in the file.
BEGIN	Beginning column number in the line number defined above.
END	Ending column number in the line number defined above.

Remarks:

1. **Input Order.** Keyword input order is sensitive. Recommended order is to define variables in *PARAMETER first, followed with this keyword, using the defined variables.
2. **Multiple Variables.** Multiple variables can be defined with one such keyword, with the file name needed to be defined only once. If there are variables located in multiple files, the keyword needs to be repeated for each file.

Examples:

An example provided below shows that multiple PIDs for individual tools and blank are defined in files data.k and data1.k. In the main input file, sim.dyn, used for LS-DYNA execution, variables (integer) are first initialized for PIDS of all tools and blank with *PARAMETER. These variables are updated with integers read from files data.k and data1.k from respective line number and column number through the use of this keyword. In the *SET_PART_LIST definition, these PIDs are used to define the part set.

Below is file data.k, to be read into sim.dyn:

```
$$$$$$$$$$$$$$$$$$$$$$$$$$  
$$$ define PIDs  
$$$$$$$$$$$$$$$$$$$$$$$$  
-----+---1-----2-----3-----4-----5-----6-----7-----  
upper die pid: 3  
lower post pid: 2  
Below is file "data1.k", also to be read into "sim.dyn":  
$$$$$$$$$$$$$$$$$$$$$$$$  
$$$$$$$$$$$$$$$$$$$$$$$$  
$$$$$$$$$$$$$$$$$$$$$$$$  
$$$ define PIDs  
$$$$$$$$$$$$$$$$$$$$$$$$  
-----+---1-----2-----3-----4-----5-----6-----7-----  
lower binder pid: 4  
blank pid: 1
```

Below is partial input for the main input file sim.dyn:

```
$-----+---1-----2-----3-----4-----5-----6-----7-----  
*INCLUDE  
blank.k  
*INCLUDE  
tool.k  
-----+---1-----2-----3-----4-----5-----6-----7-----  
*PARAMETER  
Iblankp,0  
Iupdiep,0  
Ipunchp,0  
Ilbindp,0  
Rblankmv,0.0  
Rpunchmv,0.0  
Rupdiemv,0.0
```

```
Rbindmv,0.0
Rbthick,1.6
$---+----1-----2-----3-----4-----5-----6-----7-----+
*CONTROL_FORMING_PARAMETER_READ
data.k
updiep,1,5,30,30
punchp,1,6,30,30
*CONTROL_FORMING_PARAMETER_READ
data1.k
lbindp,1,7,30,30
blankp,1,8,30,30
$---+----1-----2-----3-----4-----5-----6-----7-----+
*SET_PART_LIST
1
&blankp
*SET_PART_LIST
2
&punchp
*SET_PART_LIST
3
&updiep
*SET_PART_LIST
4
&lbindp
$---+----1-----2-----3-----4-----5-----6-----7-----+
*CONTROL_FORMING_AUTOPOSITION_PARAMETER_SET
$# psid cid dir mpsid position premove thick parname
    1     0     3     2       1      0.000 &bthick blankmv
    3     0     3     1       1      0.000           updiemv
    4     0     3     1      -1      0.000           bindmv
$---+----1-----2-----3-----4-----5-----6-----7-----+
*PART_MOVE
$pid,xmov,ymov,zmov,cid,ifset
1,0.0,0.0,&blankmv,,1
3,0.0,0.0,&updiemv,,1
4,0.0,0.0,&bindmv,,1
```

Revision Information:

This feature is available in LS-DYNA R5 Revision 55035 and later releases.

*CONTROL

*CONTROL_FORMING_POSITION

*CONTROL_FORMING_POSITION

Purpose: This keyword allows user to position tools and a blank in setting up a stamping process simulation. All tools must be pre-positioned at their home positions. For tools that are positioned above the sheet blank (or below the blank) and ready for forming, *CONTROL_FORMING_TRAVEL should be used. This keyword is used together with *CONTROL_FORMING_USER. One *CONTROL_FORMING_POSITION card may be needed for each part.

NOTE: This option has been deprecated in favor of *CONTROL_FORMING_AUTOPOSITION_PARAMETER).

Positioning Cards. For each part to be positioned include an additional card. The next “**” card terminates the input.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	PREMOVE	TARGET					
Type	I	F	I					
Default	none	none	I					

VARIABLE	DESCRIPTION
PID	Part ID of a tool to be moved, as in *PART
PREMOV	The distance to pre-move the tool, in the reverse direction of forming.
TARGET	Target tool PID, as in *PART. The tool (PID) will be moved in the reverse direction of the forming and positioned to clear the interference with the blank, then traveled to its home position with a distance GAP (*CONTROL_FORMING_USER) away from the TARGET tool to complete the forming.

Remarks:

When this keyword is used, all stamping tools must be in their respective home positions, which is also the position of each tool at its maximum stroke. From the home position each tool will be moved to its start position, clearing interference between the blank and

tool yet maintaining the minimum separation needed to avoid initial penetration. Currently the tools can only be moved and travels in the direction of the global Z-axis.

A partial keyword example is provided in manual pages under *CONTROL_FORMING_USER.

Revision information:

This feature is available starting in Revision 24641.

*CONTROL

*CONTROL_FORMING_PRE_BENDING

*CONTROL_FORMING_PRE_BENDING_{OPTION}

Available options include:

<BLANK>

LOCAL

Purpose: Bend an initially flat sheet metal blank with a user-specified radius to control the blank's gravity loaded shape during sheet metal forming.

Card 1	1	2	3	4	5	6	7	8
Variable	PSET	RADIUS	VX	VY	VZ	XC	YC	ZC
Type	I	F	F	F	F	F	F	F
Default	none	none	none	none	none	↓	↓	↓

Local Coordinate System Card. This card is required if the option LOCAL is used.

Card 2	1	2	3	4	5	6	7	8
Variable	CID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
PSET	Part set ID to be included in the pre-bending
RADIUS	Radius of the pre-bending: GT.0.0: bending center is on the same side as the element normal. LT.0.0: bending center is on the reverse side of the element normals. See Figure 12-39 .

VARIABLE	DESCRIPTION
VX, VY, VZ	Vector components of an axis about which the flat blank will be bent
XC, YC, ZC	(X, Y, Z) coordinates of the center of most-bent location. If undefined, center of gravity of the blank will be used as a default.
CID	ID for a local coordinate system in which the blank will be bent

About Pre-Bending due to Gravity:

In some situations, gravity loading upon a flat blank will result in a “concave” shape in a die. This mostly happens in cases where there is little or no punch support in the middle of the die cavity and in large stamping dies. Although the gravity loaded blank shape is correct, the end result is undesirable. For example, buckles may result during the ensuing closing and forming simulations. In practice, a true flat blank rarely exists. Typically, the blank is either manipulated (shaking or bending) by die makers in the tryout stage or by suction cups in a stamping press to get an initial convex shape prior to the binder closing and punch forming. This keyword allows this bending to be performed.

Example:

A partial keyword example (NUMISHEET2002 fender outer) is provided below, where blank part set ID variable &BLKSID defined previously is to be bent with a radius value of -10000.0 mm with the Z-axis as the bending axis on the reverse side of the blank positive normal (see [Figure 12-39](#)). The bending is off the center of gravity at (234.0, 161.0, 81.6) (to the right along positive X-axis). Only a slight pre-bending on the blank is needed to ensure a convex gravity-loaded shape. Note this keyword is input order sensitive and requires all model information to be placed ahead, therefore it is best to place the keyword at the end of an input deck.

```
*KEYWORD
:
*:CONTROL_IMPLICIT_FORMING
1
*:CONTROL_FORMING_PRE_BENDING
$    PSET      RADIUS      VX      VY      VZ      XC      YC      ZC
  &BLKSID   -10000.     0.00     0.00     1.0    234.000  161.000   81.60
...
*END
```

In [Figures 12-40](#), initial blank shape without pre-bending is shown. Without pre-bending, the gravity loaded blank sags in the middle of the die cavity (see [Figure 12-41](#)) which is likely unrealistic and would lead to predictions of surface quality issues. With pre-bending applied (see [Figure 12-42](#)), the blank bends slightly into a convex shape before loading. This shape results in an overall convex shape after gravity completes loading

(see [Figure 12-43](#)), leading to a much shorter binder closing distance, and a more realistic surface quality assessment.

Bending in a local coordinate system:

A partial keyword example below bends the blank (part set ID 5) in local coordinate system #8, about an axis 45 degrees from both the local X- and Y- axes, and perpendicular to the local Z-axis, in a radius of 1000 mm. The bending center is located in the negative surface normal side of the blank.

```
*KEYWORD
:
*DEFINE_COORDINATE_SYSTEM
$#    cid      xo      yo      zo      xl      yl      zl      cidl
      8     490.902   1.26663   98.7325   500.0   1.26663   98.7325       0
$#    xp       yp      zp
      0.0    71.9769   167.651
*CONTROL_FORMING_PRE_BENDING_LOCAL
$    PSET    RADIUS      VX      VY      VZ      XC      YC      ZC
      5    -1000.0     1.00     1.00
$    CID
      8
:
*END
```

Revision Information:

This feature is available in double precision starting in LS-DYNA DEV 66094. It is also available in LS-PrePost *eZ-Setup* for metal forming application (<http://ftp.lstc.com/-anonymous/outgoing/lsprepost/>). The option LOCAL is available starting in DEV 139260.

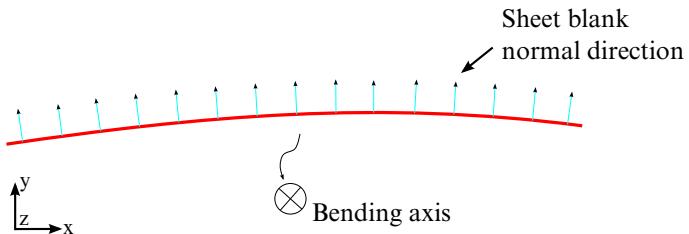


Figure 12-39. Negative "R" puts center of bending on the opposite side of the positive blank normal.

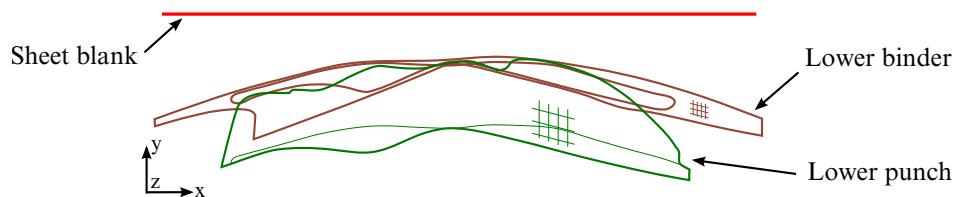


Figure 12-40. Initial model before auto-positioning.

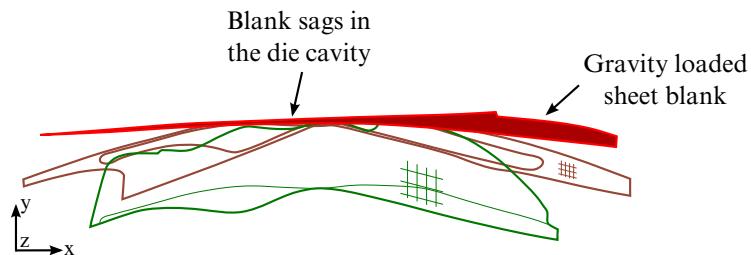


Figure 12-41. Gravity loaded blank without using this keyword.

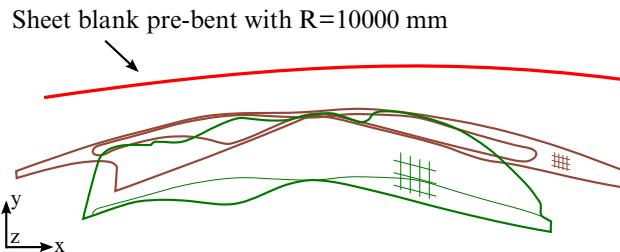


Figure 12-42. Pre-bending using this keyword (1st state in d3plot).

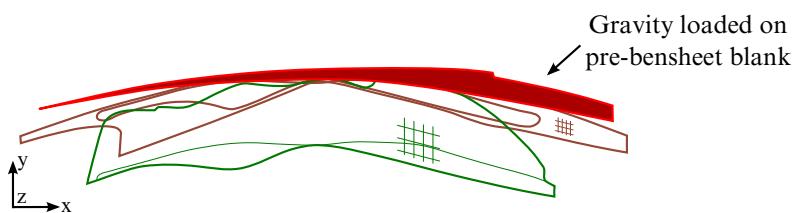


Figure 12-43. Gravity loaded shape (final state in d3plot) with convex shape.

***CONTROL_FORMING_PROJECTION**

Purpose: Remove initial penetrations between the blank and the tooling (shell elements only) by projecting the penetrated blank nodes along a normal direction to the surface of the blank or to the surface of the tool with the specified gap between the node and the tooling surface. This is useful for line die simulations of a previously formed panel to reduce tool travel, therefore saving simulation time.

Define Projection Card. This card may not be repeated.

Card 1	1	2	3	4	5	6	7	8
Variable	PIDB	PIDT	GAP	NRBST	NRTST			
Type	I	I	F	I	I			

VARIABLE	DESCRIPTION
PIDB	Part ID of the blank
PIDT	Part ID for the tool
GAP	A distance, which defines the minimum gap required
NRBST	Specify whether the blank will move along its normal direction. If its moves along the normal of blank, then this flag also specifies the direction the normal is pointing with respect to the tool. EQ.0: Move the blank's nodes along the blank's normal. The normal to the surface of the blank is pointing towards the tool. EQ.1: Move the blank's nodes along the blank's normal. The normal to the surface of the blank is pointing away from the tool. EQ.2: Move the blank nodes along the tool's normal direction. This case is useful for contact between a guide pin and blank.
NRTST	Normal direction of the tool: EQ.0: The normal to the surface of the tool is pointing towards the blank. EQ.1: The normal to the surface of the tool is pointing away from blank.

***CONTROL**

***CONTROL_FORMING_PROJECTION**

Remarks:

This feature requires consistent normal vectors for both the rigid tooling surface and the blank surface.

***CONTROL_FORMING_REMOVE_ADAPTIVE_CONSTRAINTS**

Purpose: Convert an adaptive mesh into a fully connected mesh. This feature removes adaptive constraints and connects the mesh with triangular elements.

Card 1	1	2	3	4	5	6	7	8
Variable	PID							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
PID	Part ID (see *PART) of the part whose adaptive mesh constraints are to be removed and its mesh converted into connected meshes.

Remarks:

In some applications of sheet metal forming, such as stoning or springback simulations, adaptive refinement of the sheet blank may affect the accuracy of the calculation. To avoid this problem, a non-adapted mesh is required. However, an adaptively refined mesh has the optimal mesh density that is tailored to the tooling geometry; the resulting mesh, in its initial shape, either flat or deformed, has fewer elements than a blank with non-adapted and uniformly-sized elements and thus is the most efficient for simulation. If the parameter IOFLAG in *CONTROL_ADAPTIVE is turned on, such a mesh adapt.msh will be generated at the end of each simulation, with its shape conforming to the initial input blank shape.

This keyword takes the adapted mesh, removes the adaptive constraints, and use triangular elements to connect the otherwise disconnected mesh. The resulting mesh is a fully connected mesh, with the optimal mesh density, to be used to re-run the simulation (without mesh adaptivity) for better accuracy.

Note that the original adapt.msh file from an LS-DYNA run will include not only the blank but the tooling mesh as well. For this keyword to be used, the original file can be read into LS-PrePost with the blank shown in active display only; the menu option *File* → *Save As* → *Save Active Keyword As* can be used to write out the adapted blank mesh only.

*CONTROL

*CONTROL_FORMING_REMOVE_ADAPTIVE_CONSTRAINTS

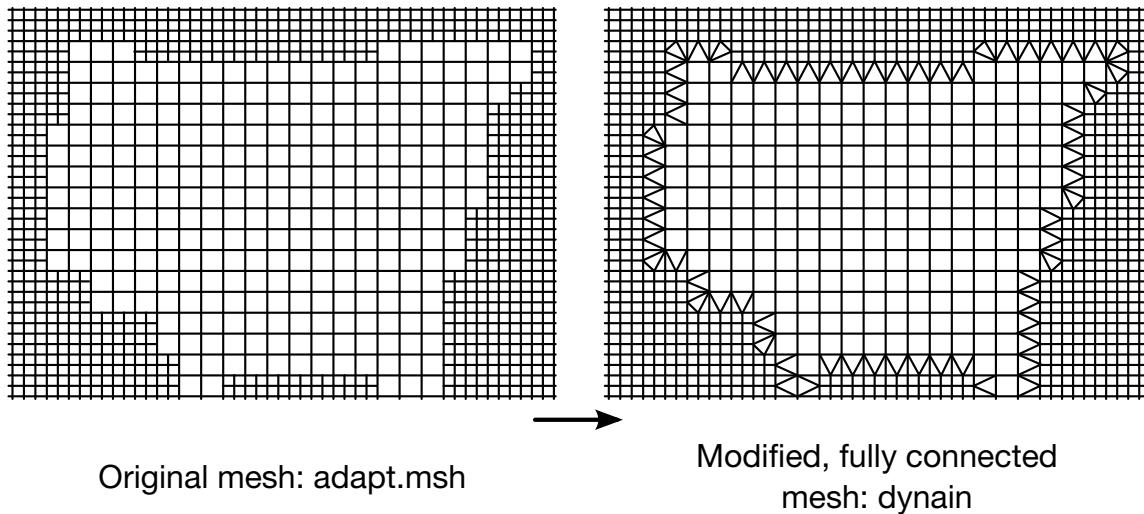


Figure 12-44. Converting an adaptive mesh to a fully connected mesh.

Example:

The following complete input file converts an adaptive mesh file blankadaptmsh.k (Figure 12-44 left) with the PID of 1 into a connected mesh (Figure 12-44 right). The resulting mesh will be in the dynain file.

```
*KEYWORD
*INCLUDE
blankadaptmsh.k
*PARAMETER
I blkpid          1
$-----1-----2-----
*CONTROL_TERMINATION
0.0
*CONTROL_FORMING_REMOVE_ADAPTIVE_CONSTRAINTS
$   PID
&blkpid
*set_part_list
1
&blkpid
*INTERFACE_SPRINGBACK_LSDYNA_NOTHICKNESS
1
*INTERFACE_SPRINGBACK_EXCLUDE
INITIAL_STRAIN_SHELL
INITIAL_STRESS_SHELL
*PART
$       PID      SID      MID
    &blkpid      1      1
*MAT_037
...
*SECTION_SHELL
$     SECID      ELMFORM      SHRF      NIP
        1           2  0.000E+00      3
1.0,1.0,1.0,1.0
*END
```

***CONTROL_FORMING_SCRAP_FALL**

Purpose: Direct and aerial trimming of a sheet metal part by trim steels in a trim die. According to the trim steels and trim vectors defined, the sheet metal part will be trimmed into a parent piece and multiple scrap pieces. The parent piece is defined as a fixed rigid body. Trimmed scraps (deformable shells) are constrained along trim edges until they come into contact with the trim steel; the edge constraints are gradually released as the trim steel's edge contacts the scrap piece, allowing for simulating contact-based scrap fall. This keyword applies to shell elements only.

Card Sets. Include Card 1 columns 1-6 only per each scrap piece for the *constraint release method* (see [Remarks](#)). For the *scrap trimming method* include one set of Cards 1, 2, and 3 per trim steel. The next keyword ("*") card terminates the input.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	VECTID	NDSET	LCID	DEPTH	DIST	IDRGD	IFSEED
Type	I	I	I	I	F	F	I	I
Default	none	global z	none	none	none	none	none	none

Card 2	1	2	3	4	5	6	7	8
Variable	NDBEAD	SEEDX	SEEDY	SEEDZ	EFFSET	GAP	IPSET	EXTEND
Type	I	F	F	F	F	F	I	F
Default	none	↓	↓	↓	none	none	none	none

Card 3	1	2	3	4	5	6	7	8
Variable	NEWID							
Type	I							
Default	none							

*CONTROL

*CONTROL_FORMING_SCRAP_FALL

VARIABLE	DESCRIPTION
PID	Part ID of a scrap piece. This part ID becomes a dummy ID if all trimmed scrap pieces are defined by NEWID. See definition for NEWID and Figure 12-47 .
VECTID	Vector ID for a trim steel movement, as defined by *DEFINE_VECTOR. If left undefined (blank), global z-direction is assumed.
NDSET	A node set consisting of all nodes along the cutting edge of the trim steel. Note that prior to Revision 90339 the nodes in the set must be defined in consecutive order. See Remarks (LS-PrePost) below on how to define a node set along a path in LS-PrePost. This node set, together with VECTID, is projected to the sheet metal to form a trim curve. To trim a scrap out of a parent piece involving a neighboring trim steel, which also serves as a scrap cutter, the node set needs to be defined for the scrap cutter portion only for the scrap, see Figure 12-47 .
LCID	Load curve ID governing the trim steel kinematics, as defined by *DEFINE_CURVE. GT.0: velocity-controlled kinematics LT.0: displacement-controlled kinematics
DEPTH	A small penetrating distance between the cutting edge of the trim steel and the scrap piece, as shown in Figure 12-46 . Nodes along the scrap edge are released from automatically added constraints at the simulation start and are free to move after this distance is reached.
DIST	A distance tolerance measured in the plane normal to the trim steel moving direction, between nodes along the cutting edge of the trim steel defined by NDSET and nodes along an edge of the scrap, as shown in Figure 12-45 . This tolerance is used to determine if the constraints need to be added at the simulation start to the nodes along the trim edge of the scrap piece.
IDRGD	Part ID of a parent piece, which is the remaining sheet metal after the scrap is successfully trimmed out of a large sheet metal. Note the usual *PART needs to be defined somewhere in the input deck, along with *MAT_20 and totally fixed translational and rotational DOFs. See Figure 12-47 .

VARIABLE	DESCRIPTION
IFSEED	A flag to indicate the location of the scrap piece. EQ.0: automatically determined. The trim steel defined will be responsible to trim as well as to push (have contact with) the scrap piece. EQ.1: automatically determined; however, the trim steel in definition will only be used to trim out the scrap, not to push (have contact with) the scrap piece. EQ.-1: user specified by defining SEEDX, SEEDY, and SEEDZ
NDBEAD	A node set to be excluded from initially imposed constraints after trimming. This node set typically consists of nodes in the scrap draw bead region where due to modeling problems the beads on the scrap initially interfere with the beads on the rigid tooling; it causes scrap to get stuck later in the simulation if left as is. See Figure 12-48 .
SEEDX, SEEDY, SEEDZ	x, y, z coordinates of the seed node on the scrap side; define only when IFSEED is set to "-1". See Figure 12-47 .
EFFSET	Scrap edge offset amount away from the trim steel edge, towards the scrap seed node side. This is useful to remove initial interference between the trimmed scrap (because of poorly modeled trim steel) and coarsely modeled lower trim post. See Figure 12-47 .
GAP	Scrap piece offset amount from the part set defined by IPSET (e.g. top surfaces of the scrap cutters), in the direction of the element normals of the IPSET. This parameter makes it easier to remove initial interference between the scrap and other die components. See Figure 12-50 .
IPSET	A part set ID from which the scrap will be offset to remove the initial interference; works together only with GAP. The part set ID should only include portions of tool parts that are directly underneath the scrap (top surface portion of the tools). The normals of the IPSET must point toward the scrap. The parts that should belong to IPSET are typically of those elements on the top surface of the scrap cutter; see Figure 12-50 .
EXTEND	An amount to extend a trim steel's edge based on the NDSET defined, so it can form a continuous trim line together with a

*CONTROL

*CONTROL_FORMING_SCRAP_FALL

VARIABLE	DESCRIPTION
	neighboring trim steel, whose edge may also be extended, to trim out the scrap piece. See Figure 12-47 .
NEWID	New part ID of a scrap piece for the scrap area defined by the seed location. If this is not defined (left blank) or input as "0", the scrap piece will retain original PID as its part ID. See Figure 12-47 . This is useful when one original scrap is trimmed into multiple smaller pieces, and contacts between these smaller pieces need to be defined.

Background:

Sheet metal trimming and the resulting scrap fall are top factors in affecting the efficiency of stamping plants worldwide. Difficult trimming conditions, such as those multiple direct trims, a mixture of direct and cam trims, and multiple cam trims involving bypass condition, can cause trimmed scraps to get stuck around and never separate from the trim edge of the upper trim steels or lower trim post. Inappropriate design of die structure and scrap chute can slow down or prevent scraps from tumbling out to the scrap collectors. Smaller scrap pieces (especially aluminum) can sometimes shoot straight up, and get stuck and gather in areas of the die structure. All these problems result in shutdowns of stamping presses, reducing stroke-per-minute (SPM) and causing hundreds of thousands of dollars in lost productivity.

With this keyword, engineers can consider the trimming details, manage the scrap trim and the drop energy, study different trimming sequences, explore better die structure and scrap chutes design and layout before a trim die is even built. This feature is developed in conjunction with the *Ford Motor Company*.

The constraint release method:

Prior to Revision 91471 (see [Revision](#)), simulating the scrap trim and fall uses the "constraint release" method, where only the scrap piece is modeled and defined.

As shown in [Figure 12-45](#), the scrap piece is modeled as a deformable body and the trim steel and trim post as rigid shell elements, while the parent piece does not need to be modeled at all. Between the trim edge of the scrap piece and the post there should be a gap (indicated by GAP in the figure). The gap ensures that the contact interface (to be explained later) correctly accounts for the shell thickness along the edge. A gap that is too small may cause initial penetration between the scrap and the post which may manifest as unphysical adhesion between the scrap and the post.

The edge of the scrap piece should initially be flush with that of the trim post (perpendicular to the trim direction), just as exactly what happens in the production environment.

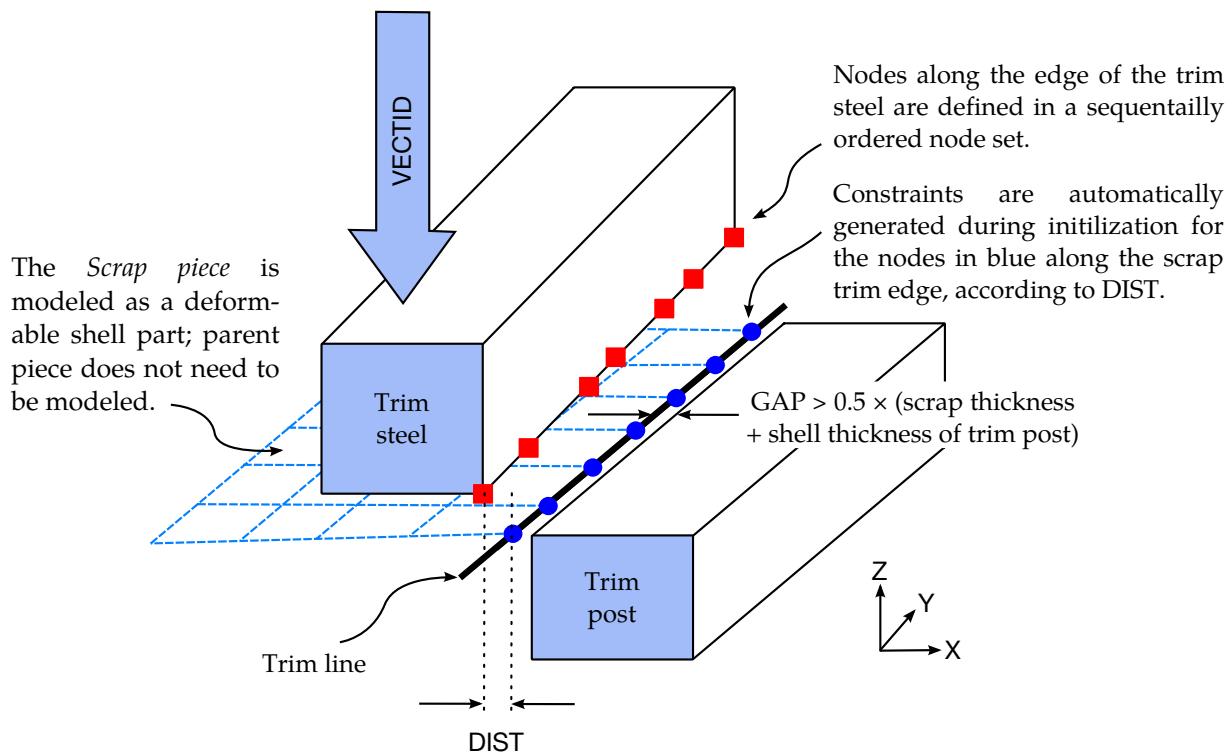


Figure 12-45. Modeling details of the constraint release method. *Drawing modified from the original sketches courtesy of the Ford Motor Company.*

If the scrap is unrealistically positioned above the trim post edge, the scrap may be permanently caught between the trim steel and the post under a combination of uncertain trimming forces as the trim steel moves down.

During initialization, constraints are added automatically on the nodes along the scrap trim edge corresponding to the node set (NDSET) along the trim steel, based on the supplied tolerance variable DIST and trim vector VECTID. Although the direction of the path is not important, prior to Revision 90339, the NDSET must be arranged so that the nodes are in a sequential order (LS-PrePost 4.0 creating node set by path). As the edge of the trim steel comes within DEPTH distance of the trim line, the constraints are removed. The contact interfaces serve to project the motion of the trim steel onto the scrap piece, see [Figure 12-46](#).

The scrap trimming method:

The original simplified method has the following drawbacks:

1. No scrap trimming – the scrap piece cannot be trimmed directly from a parent piece; an exact scrap piece after trimming must be modeled.

*CONTROL

*CONTROL_FORMING_SCRAP_FALL

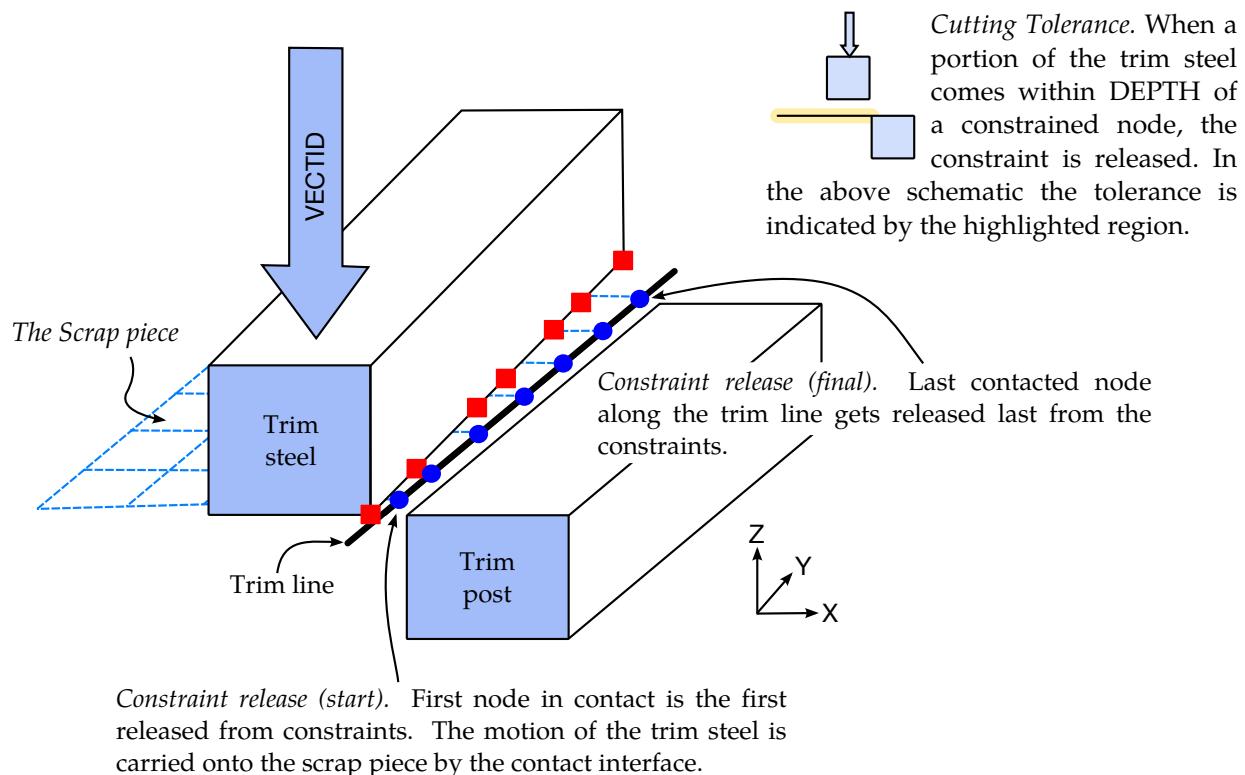


Figure 12-46. Contact-based separation and contact-driven kinematics and dynamics in the constraint release method. *Drawing modified from the original sketches courtesy of the Ford Motor Company.*

2. Poorly (or coarsely) modeled draw beads in the scrap piece do not fit properly in badly modeled draw beads on the tooling, resulting in initial interferences between the two and therefore affecting the simulation results.
3. For poorly (or coarsely) modeled scrap edges and trim posts, users have to manually modify the scrap trim edges to clear the initial interference with the trim posts.
4. Users must clear all other initial interferences (e.g. between scrap and scrap cutter) manually.

Based on users' feedback, a new method "scrap trimming" (after Revision 91471) has been developed to address the above issues and to, furthermore, reduce the effort involved in preparing the model. The new method ([Figure 12-47](#)) involves trimming scrap from an initially large piece of sheet metal, leaving the parent piece as a new fixed rigid body, defined by user (along with *SECTION_SHELL and *MAT_20). If one large piece of sheet metal is trimmed in many different places to form several scrap pieces, a new PID can be defined for each of the scrap pieces (NEWID), along with *SECTION_SHELL and *MAT_037, for example. The trim lines are obtained from the trim steel edge node set NDSET and the trim vector VECTID.

Note: The trim steel cannot have walls that are parallel to the trim direction; otherwise the scrap will not be trimmed properly from the parent piece. This is illustrated in [Figure 12-49](#).

Parameters related to the constraint release method:

1. The value of DEPTH is typically set to one-half of the scrap thickness.
2. The initial gap separating the scrap from the post must be greater than the average of the scrap and post thickness values; see [Figure 12-45](#).
3. The input parameter DIST should be set larger than the maximum distance between nodes along the trim steel edge and scrap edge in the view along the trim direction; see [Figure 12-45](#).

Parameters related to the scrap trimming method:

1. Similar to DEPTH, EFFSET should be typically set to one-half of the scrap thickness, although it may be larger for some poorly modeled trim steels and trim posts.

Contact:

Only *CONTACT_FORMING contact interfaces are allowed for contact between the scrap piece and the trim steel. In particular, *CONTACT_FORMING_SURFACE_TO_SURFACE is recommended. A negative contact offset must be used; this is done typically by setting the variable SBST in *CONTACT_FORMING_SURFACE_TO_SURFACE to the negative thickness value of the scrap piece.

For contact between the scrap piece and the shell elements in all the other die structures, *CONTACT_AUTOMATIC_GENERAL should be used for the edge-to-edge contact frequently encountered during the fall of the scrap piece. All friction coefficients should be small. The explicit time integrator is recommended for the modeling of scrap trim and fall. Mass scaling is not recommended.

LS-PrePost:

The node set (NDSET) defined along the trim steel edge can be created with *LS-PrePost 4.0*, via *Model/CreEnt/Cre, Set Data, *SET_NODE, ByPath*, then select nodes along the trim edge continuously until finish and then hit *Apply*.

*CONTROL

*CONTROL_FORMING_SCRAP_FALL

Keyword examples – the constraint release method:

A partial example of using the keyword below includes a node set ID 9991 along the trim steel (PID 2) edge used to release the constraints between the scrap piece with PID 1, and the parent piece. The LCID for the trim steel kinematics is (+)33 (load curve is controlled by velocity) moving in the $-z$ direction. The trimming velocity is defined as 1000 mm/s, and the retracting velocity is 4000 mm/s. The variables DEPTH and DIST are set to 0.01 and 2.5, respectively. The contact interface between the trim steel and scrap piece is defined using *CONTACT_FORMING_SURFACE_TO_SURFACE and contact between the scrap and all other die structures are defined using *CONTACT_AUTOMATIC_GENERAL.

```
*KEYWORD
*CONTROL_TERMINATION
&endtime
*CONTROL_FORMING_SCRAP_FALL
$      PID      VECTID      NDSET      LCID      DEPTH      DIST
      1          9991       33        0.75       2.0
*SET_NODE_LIST
 9991
 24592    24591    24590    24589    24593    24594    24595    24596
*BOUNDARY_PRESCRIBED_MOTION_rigid
$pid,dof,vad,lcid,sf,vid,dt,bt
2,3,0,33,-1.0
*DEFINE_CURVE
33
0.0,0.0
0.216,1000.0
0.31,-4000.0
0.32,0.0
0.5,0.0
$-----1-----2-----3-----4-----5-----6-----7-----8
*CONTACT_forming_surface_to_surface_ID
 1
   1      2      3      3      0      0      0      0
   0.02    0.0    0.0    0.0    20.0     0    0.01.0000E+20
$#  sfsa    sfsb    sast    sbst    sfsat    sfsbt    fsf      vsf
   0.0    0.0    0.0    &bst     1.0     1.0    1.0     1.0
$-----1-----2-----3-----4-----5-----6-----7-----8
*CONTACT_AUTOMATIC_GENERAL_ID
 2
```

*END

For the negative option of LCID, displacement will be used as input to control the tool kinematics. A partial example is provided below, where LCID is defined as a negative integer of a load curve, controlling the trim steel kinematics. The trim steel is moving down for 27.6075 mm in 0.2 sec to trim, and moving up for the same distance to its original position in 0.3 sec to retract. Although this option is easier to use, the corresponding velocity from the input time and displacement must be realistic for a realistic simulation.

```
*CONTROL_FORMING_SCRAP_FALL
$ LCID<0: trimming steel kinematics is controlled by displacement.
$      PID      VECTID      NDSET      LCID      DEPTH      DIST
      1        44         1     -33332      0.70      2.00
```

```
*DEFINE_VECTOR
44,587.5,422.093,733.083,471.104,380.456,681.412
*BOUNDARY_PRESCRIBED_MOTION_RIGID_LOCAL
$pid,dof,vad,lcid,sf,vid,dt,bt
11,3,2,33332,1.0,44
*DEFINE_CURVE
33332
0.0,0.0
0.2,-27.6075
0.5,0.0
```

A keyword example – the scrap trimming method:

The keyword example below shows three scrap pieces, with part ID &spid1 (PID, also is the original blank to be trimmed) and new part IDs 1001 and 1002 (NEWID), being trimmed out of the original blank &spid1 (PID); the remaining parent piece is defined as a fixed rigid body with part ID 110 (IDRGD). Each scrap piece is indicated by a point location defined by SEEDX, SEEDY and SEEDZ. Each scrap is also to be trimmed by three different trim steels. For example, scrap piece &spid1 is to be trimmed by trim steels with part IDs 22, 23 and 24 (as referenced by *BOUNDARY_PRESCRIBED_MOTION_RIGID_LOCAL) in trim directions defined respectively by vector IDs (from VID in *DEFINE_VECTOR) &cord1, &cord2 and &cord3 (VECTID) and with trim kinematics defined respectively by load curve IDs 1800, 1801 and 1802 (LCID). The scrap pieces &spid1, 1001 and 1002 are each offset by 0.60mm (GAP), in the direction normal to the elements defined by part set IDs 887, 888, and 889 (IPSET), respectively. The trim edge offset is 0.90mm (EFFSET) away from the trim edge for all scraps. The draw bead node sets to be released are, 987, 988 and 989 for each scrap &spid1, 1001, and 1002, respectively.

```
*CONTROL_FORMING_SCRAP_FALL
$ The next three cards define scrap piece &spid1 being trimming simultaneously
$ with three trim steels with VLCIDs of 1800, 1801 and 1802:
$     PID      VECTID      NDSET      LCID      DEPTH      DIST      IDRGD      IFSEED
$     &spid1    &cord1    &nset1    1800    &depth1    2.00      110       -1
$     NDBEAD    SEEDX      SEEDY      SEEDZ      EFFSET      GAP      IPSET      EXTEND
$         987   -528.046   373.40    710.000    0.90      0.60      887       8.0
$     NEWID
$         0
$     &spid1    &cord2    &nset2    1801    &depth1    2.00      110       -1
$         987   -528.046   373.40    710.000    0.90      0.60      887       8.0
$         0
$     &spid1    &cord3    &nset3    1802    &depth1    2.00      110       -1
$         987   -528.046   373.40    710.000    0.90      0.60      887       8.0
$         0
$ The next three cards define scrap piece 1001 being trimming simultaneously
$ with three trim steels with VLCIDs of 1802, 1803 and 1804:
$     &spid1    &cord3    &nset33   1802    &depth1    2.00      110       -1
$         988   -252.452   383.322   799.974    0.90      0.60      888       8.0
$         1001
$     &spid1    &cord4    &nset4    1803    &depth1    2.00      110       -1
$         988   -252.452   383.322   799.974    0.90      0.60      888       8.0
$         1001
$     &spid1    &cord5    &nset5    1804    &depth1    2.00      110       -1
$         988   -252.452   383.322   799.974    0.90      0.60      888       8.0
$         1001
$ The next three cards define scrap piece 1002 being trimming simultaneously
$ with three trim steels with VLCIDs of 1804, 1805 and 1806:
```

*CONTROL

*CONTROL_FORMING_SCRAP_FALL

```
&spid1  &cord5  &nset55    1804  &depth1   2.00    110     -1
989    74.452  404.522  857.974  0.90    0.60    889     8.0
1002
&spid1  &cord6  &nset6     1805  &depth1   2.00    110     -1
989    74.452  404.522  857.974  0.90    0.60    889     8.0
1002
&spid1  &cord7  &nset7     1806  &depth1   2.00    110     -1
989    74.452  404.522  857.974  0.90    0.60    889     8.0
1002
*BOUNDARY_PRESCRIBED_MOTION_rigid
$      PID      DOF      VAD      LCID      SF      VID      DT      BT
22          3          0      1800      -1.0      &cord1
*DEFINE_CURVE
1800
0.0,0.0
0.216,1000.0
0.31,-4000.0
0.32,0.0
1.0,0.0
*DEFINE_VECTOR
$      VID      XT      YT      ZT      XH      YH      ZH
&cord1  587.5  422.093  733.083  471.104  380.456  681.412
*BOUNDARY_PRESCRIBED_MOTION_rigid
$      PID      DOF      VAD      LCID      SF      VID      DT      BT
23          3          0      1801      -1.0      &cord2
*DEFINE_CURVE
1801
0.0,0.0
0.326,1000.0
0.44,-4000.0
0.60,0.0
1.0,0.0
*DEFINE_VECTOR
$      VID      XT      YT      ZT      XH      YH      ZH
&cord2  587.5  522.0    763.0    10.104   80.6    1.1
*BOUNDARY_PRESCRIBED_MOTION_rigid
$      PID      DOF      VAD      LCID      SF      VID      DT      BT
24          3          0      1802      -1.0      &cord3
*DEFINE_CURVE
1802
0.0,0.0
0.36,1000.0
0.42,-4000.0
0.64,0.0
1.0,0.0
*DEFINE_VECTOR
$      VID      XT      YT      ZT      XH      YH      ZH
&cord3  85.5    42.93   3.93     7.4      30.4    41.2
```

Revision/Other information:

A graphical user interface capable of setting up a complete input deck for the original simplified method is now available in *LS-PrePost* under *APPLICATION/Scrap Trim* (<http://ftp.lstc.com/anonymous/outgoing/lsprepost/>). A reference paper regarding the development and application of this keyword for the constraint release method can be found in the *proceedings of the 12th International LS-DYNA User's Conference*. The following provides a list of revision history for the keyword:

1. The *constraint release method* is available between LS-DYNA Revision 63618 and 91471.
2. The *scrap trimming method* is available starting in Revision 91471.
3. The parameter NEWID is available starting in Revision 92578.

The restriction that NDSET must be defined in a consecutive order is lifted starting in Revision 90339.

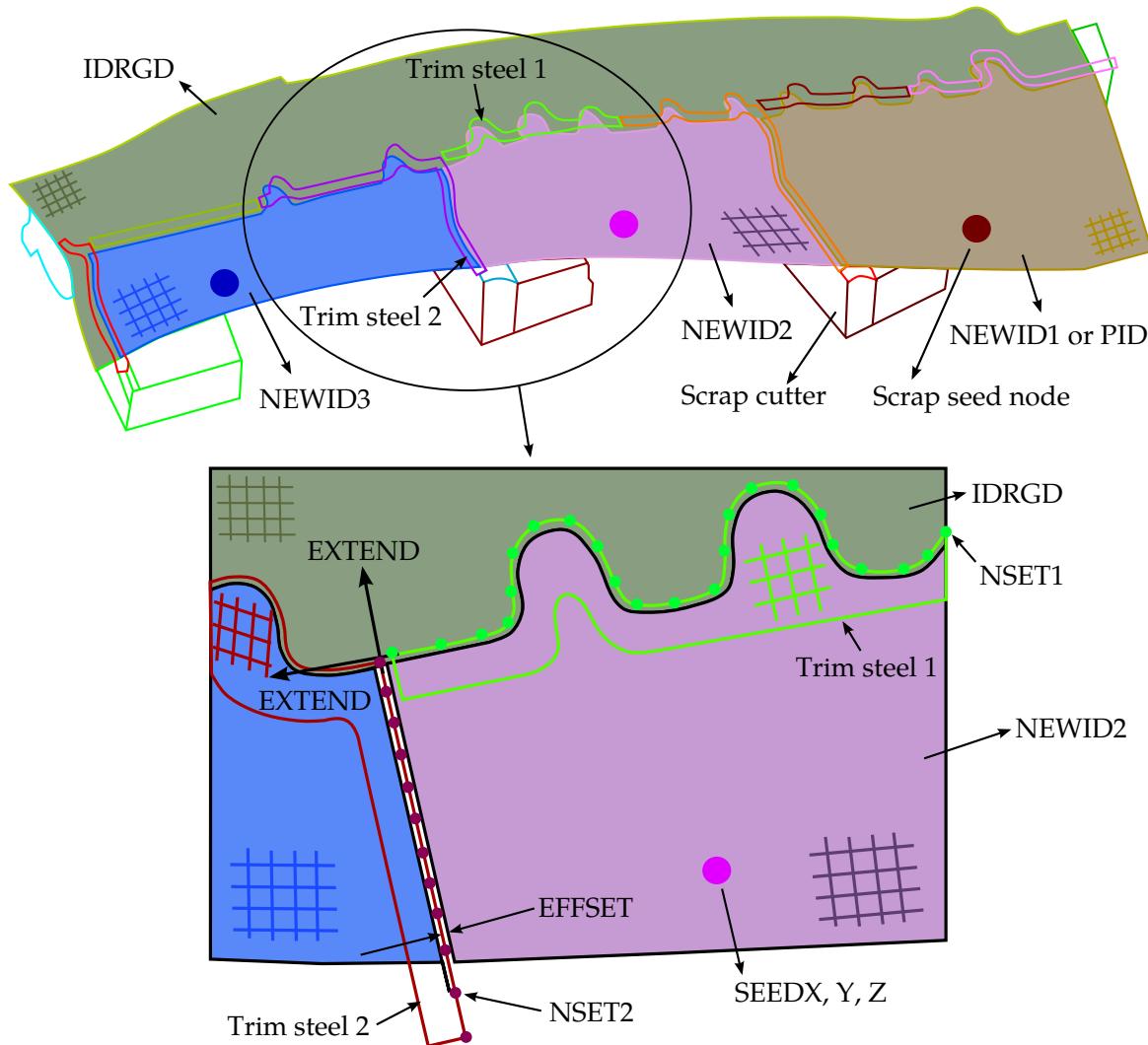


Figure 12-47. Trimming of multiple scraps and parameter definitions in the scrap trimming method. *Model courtesy of the Ford Motor Company.*

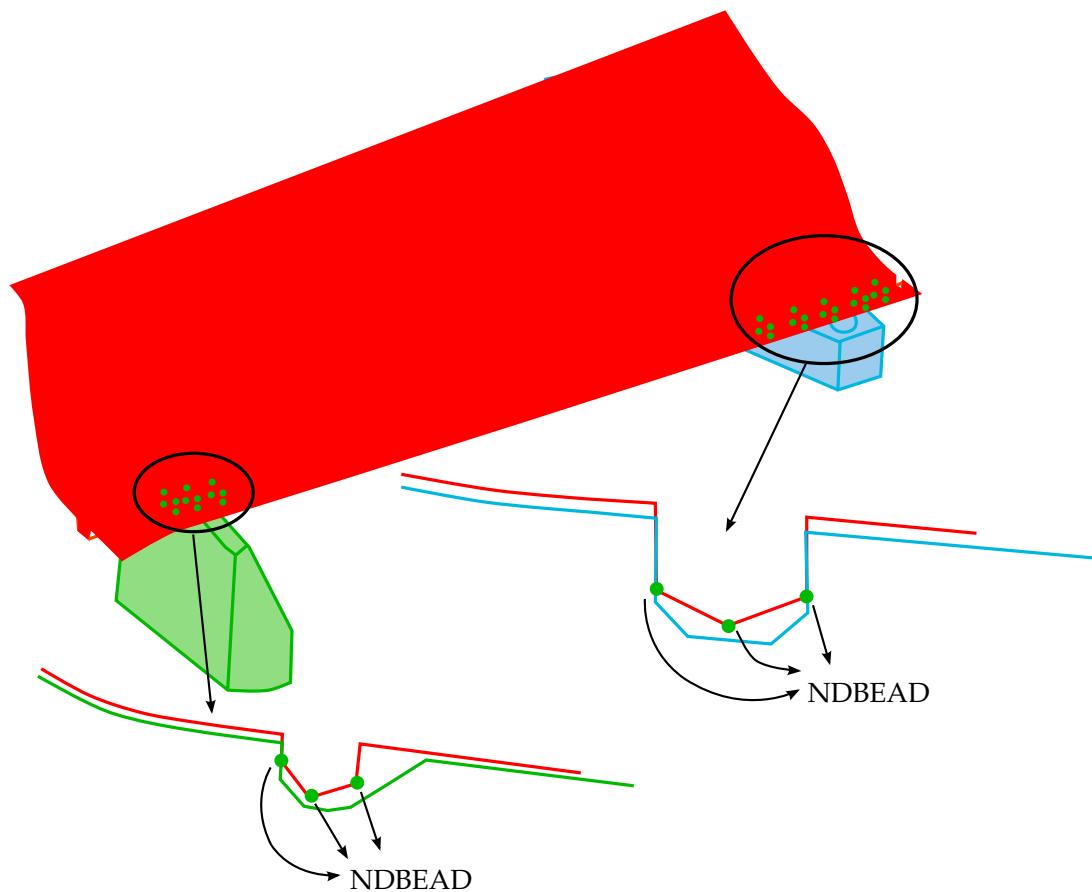


Figure 12-48. Definition of NDBEAD in the scrap trimming method. *Model courtesy of the Ford Motor Company.*

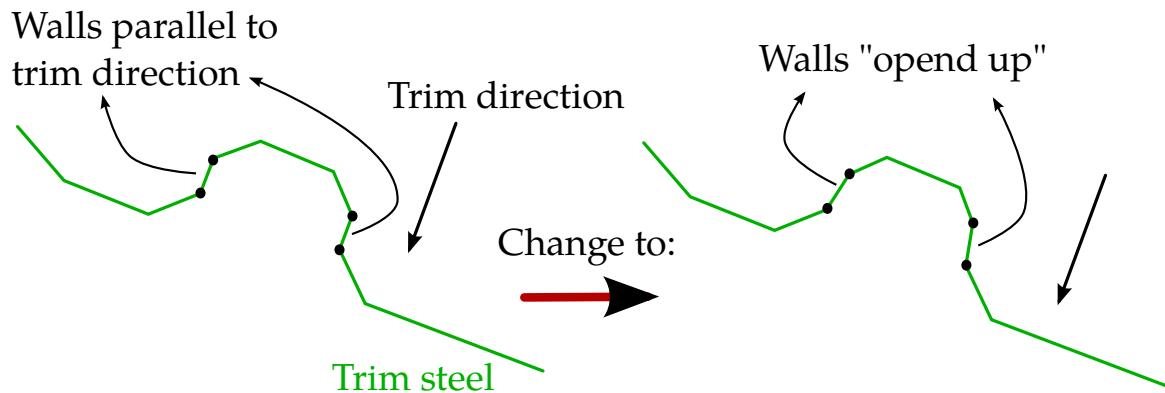


Figure 12-49. Walls that are parallel to trim direction anywhere in a trim steel (left) are not allowed. Wall needs to be opened up in relation to the trim direction for a successful simulation (right).

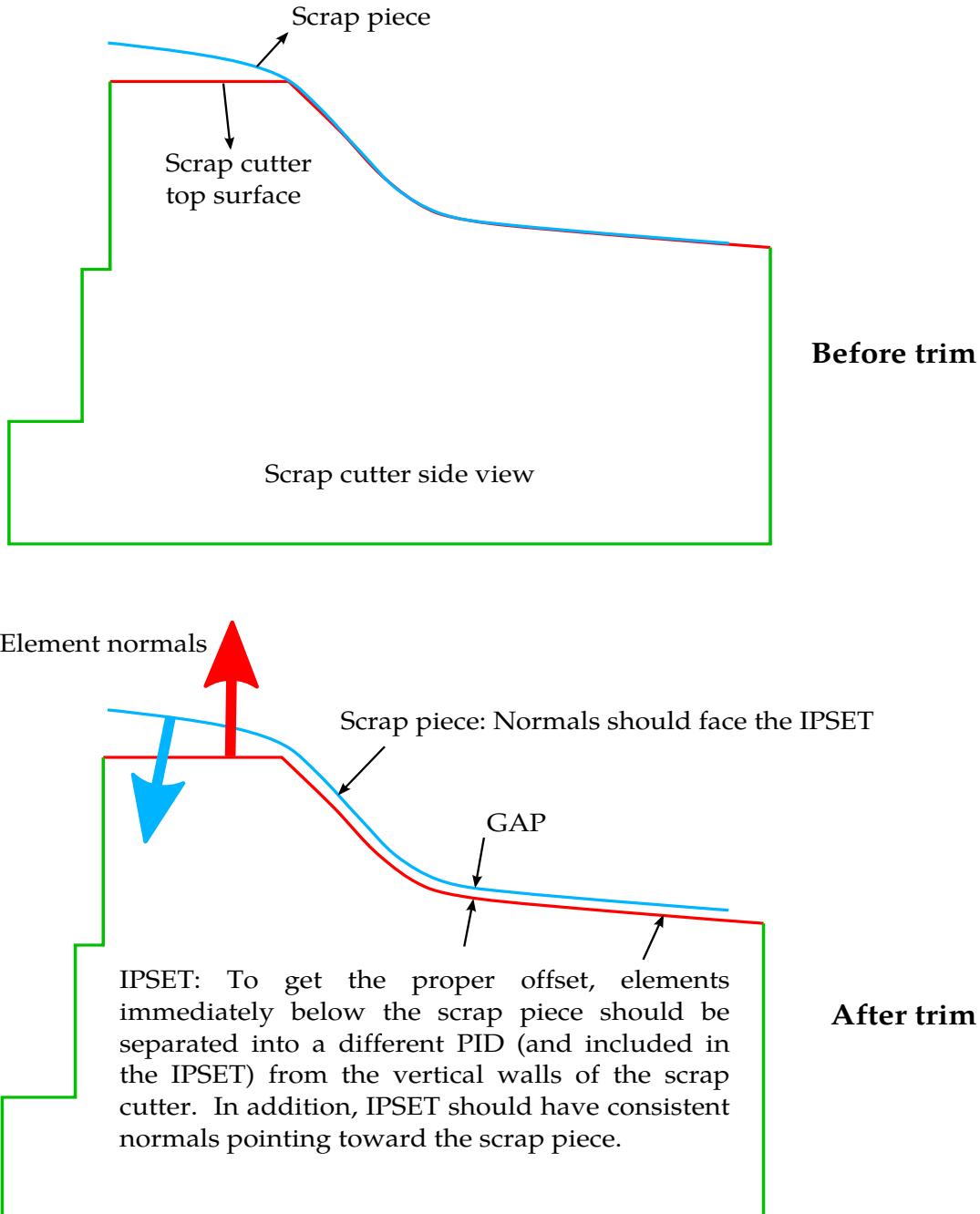


Figure 12-50. Element normal of the IPSET in the scrap trimming method.
Model courtesy of the Ford Motor Company.

*CONTROL

*CONTROL_FORMING_SHELL_TO_TSHELL

*CONTROL_FORMING_SHELL_TO_TSHELL

Purpose: This keyword converts thin shell elements (*SECTION_SHELL) to thick shell elements (*SECTION_TSHELL). It generates segments on both the top and bottom sides of the thick shells for contact. Mesh adaptivity is also supported.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	THICK	MIDSF	IDSEGB	IDSEGTT			
Type	I	I	I	I	I			
Default	none	none	0	Rem 4	Rem 4			

VARIABLE	DESCRIPTION
PID	Part ID of the thin shell elements.
THICK	Thickness of the thick shell elements (TSHELL).
MIDSF	TSHELL's mid-plane position definition (see Figure 12-51 and Remark 4): EQ.0: Mid-plane is at thin shell surface. EQ.1: Mid-plane is at one half of THICK above thin shell surface. EQ.-1: Mid-plane is at one half of THICK below thin shell surface.
IDSEGB	Set ID of the segments to be generated at the bottom layer of the TSHELLs, which can be used for defining a contact surface with segments. The bottom layer of the TSHELLs has an outward normal that points in the opposite direction to the positive normal side of thin shells; see Figure 12-51 . Note the default normal of the generated segments are consistent with the thin shells' normal. To reverse this default normal, set the IDSEGB to a negative number. See Remark 2 .
IDSEGTT	Set ID of the segments to be generated at the top layer of the TSHELLs, which can be used for defining a contact surface with segments. The top side of a TSHELL has an outward normal that points in the same direction as the positive normal side of the thin shells; see Figure 12-51 .

VARIABLE	DESCRIPTION
	The normal of the generated segments are consistent with the thin shells' normal. To reverse this default normal, set the IDSEGT to a negative number. See Remark 2 .

Remarks:

1. **Node and Element Numbering.** Node IDs of the thick shell elements will be the same as those for the thin shells. Element IDs of the thick shell elements will start at 2 (so renumber element IDs of other parts accordingly). Only one layer of thick shells will be created.
2. **Contact.** For FORMING contacts, if the generated segments are used as a SURFA member in contact with a SURFB member of a rigid body, the normal vectors for the rigid body must be consistent and facing the SURFA segments. The normal vectors for the SURFA segments are not required to point at the rigid bodies, although they should be made consistent.
3. **Adaptivity.** New nodes generated adaptively from their parent nodes with *BOUNDARY_SPC are automatically constrained accordingly.
4. **Contact and Segment Sets.** If no contact for the TSHELLs is defined in the input deck, then the fields IDSEGB and IDSEGT are ignored and the field MIDSF will be set to 0, regardless of the user set value. See the first example.

Examples:

1. A standalone part of thin shell elements can be changed to thick shell elements with a simplified small input deck. The following will convert shell elements with PID 100 of thickness 1.5 mm to thick shell elements of PID 100 with thickness of 2.0 mm, with thick shell meshes stored in the file dynain.geo. Note that MIDSF, IDSEGB and IDSEGT are ignored in this case and do not need to be set.

```
*KEYWORD
*CONTROL_TERMINATION
0.0
*INCLUDE
shellupr.k
*SET_PART_LIST
1,
100
*PART
Sheet blank
100,100,100
*SECTION_SHELL
$      SID      EIFORM      SHRF      NIP      PROPT
        100          2       0.833       3         1.0
$      T1       T2       T3       T4      NLOC
1.5,1.5,1.5,1.5
```

*CONTROL

*CONTROL_FORMING_SHELL_TO_TSHELL

```
*MAT_024
$     MID      RO      E      PR      SIGY      ETAN      FAIL      TDEL
$     100    7.85E-09  2.07E+05  0.28    382.8      0.0      0.0      0.0
$     C       P       LCSS    LCSR      VP
$     0.0     0.0      92      0      0.0
$     EPS1    EPS2    EPS3    EPS4    EPS5    EPS6    EPS7    EPS8
$     ES1     ES2     ES3     ES4     ES5     ES6     ES7     ES8

*DEFINE_CURVE
92,,,0.5
 0.0000000000E+00  3.8276000000E+02
 4.0000000000E-03  3.9616000000E+02
 8.0000000000E-03  4.0695000000E+02
  :
  :
*DEFINE_CURVE
90905
 0.0000000000E+00  0.380000000E+03
 0.30000003E-02   0.392489226E+03
 0.60000005E-02   0.403294737E+03
 0.899999961E-02  0.412847886E+03
 0.120000001E-01  0.421429900E+03
 0.150000006E-01  0.429234916E+03
 0.179999992E-01  0.436402911E+03
 0.209999997E-01  0.443038343E+03
*INTERFACE_SPRINGBACK_LSDYNA
1
OPTCARD,,,1
*CONTROL_FORMING_SHELL_TO_TSHELL
$     PID      THICK
$     100     2.0
*END
```

2. The conversion can also be done in an input deck set up for a complete metal forming simulation with thin shell elements that are part of a sheet blank. The conversion happens at the beginning of the simulation, as shown in an example below. This partial deck is of a thin shell sheet blank with PID 1 that is to be converted to a thick shell sheet blank with thickness of 1.6 mm. The commented out keywords were originally for the thin shells but have been changed due to the conversion. For example, the *SECTION_TSHELL is defined instead of *SECTION_SHELL for the sheet blank. The corresponding material type for the sheet blank (*MAT_037) is also changed to a type that supports solid element simulations (*MAT_024). The mid-plane of the thick shells is one half of 1.6 mm below the thin shells' surface. Segment IDs 10 (IDSEG1) and 11 (IDSEG2) are created at the bottom and top side of the thick shells, respectively, as shown in [Figure 12-51](#). Segment set 10 is defined to contact with the lower punch (part set ID 2), while segment set 11 is used for contact with the upper die cavity (part set ID 3).

```
*KEYWORD
...
*PART
Sheet blank
1,1,1
$-----1-----2-----3-----4-----5-----6-----7-----8
$           Blank property
$-----1-----2-----3-----4-----5-----6-----7-----8
$*SECTION_SHELL
```

***CONTROL_FORMING_SHELL_TO_TSHELL**
***CONTROL**

```

$      SID    elform      SHRF      nip      PROPT      QR/IRID      ICOMP      SETYP
$      1        2       0.833      3        1.0
$      T1        T2       T3        T4      NLOC
$&blthick,&blthick,&blthick,&blthick
*SECTION_TSHELL
$      SID    elform      SHRF      nip      PROPT      QR/IRID      ICOMP      SETYP
$      1        1       0.833    &nip      1.0
$-----1-----2-----3-----4-----5-----6-----7-----8
*MAT_024
$      MID      RO       E        PR       SIGY      ETAN      FAIL      TDEL
$      1   7.85E-09  2.07E+05    0.28    382.8      0.0      0.0      0.0
$      C       P       LCSS      LCSR      VP
$      0.0       0.0       92        0       0.0
$      EPS1     EPS2     EPS3      EPS4      EPS5     EPS6      EPS7      EPS8
$      ES1      ES2      ES3      ES4      ES5      ES6      ES7      ES8

*DEFINE_CURVE
92,,,0.5
      0.000000000E+00      3.8276000000E+02
      4.000000000E-03      3.9616000000E+02
      8.000000000E-03      4.0695000000E+02
      :
      :
*INTERFACE_SPRINGBACK_LSDYNA
1
OPTCARD,,,1
*CONTROL_FORMING_SHELL_TO_TSHELL
$      PID      THICK      MDSF      IDSEGB      IDSEG7
      100       1.6       -1        10        11
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE
$      SURFA     SURFB     SURFATYP     SURFBTYP
      11         2         0         2
      :
      :
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE
$      SURFA     SURFB     SURFATYP     SURFBTYP
      10         3         0         2
      :
      ...
*END

```

*CONTROL

*CONTROL_FORMING_SHELL_TO_TSHELL

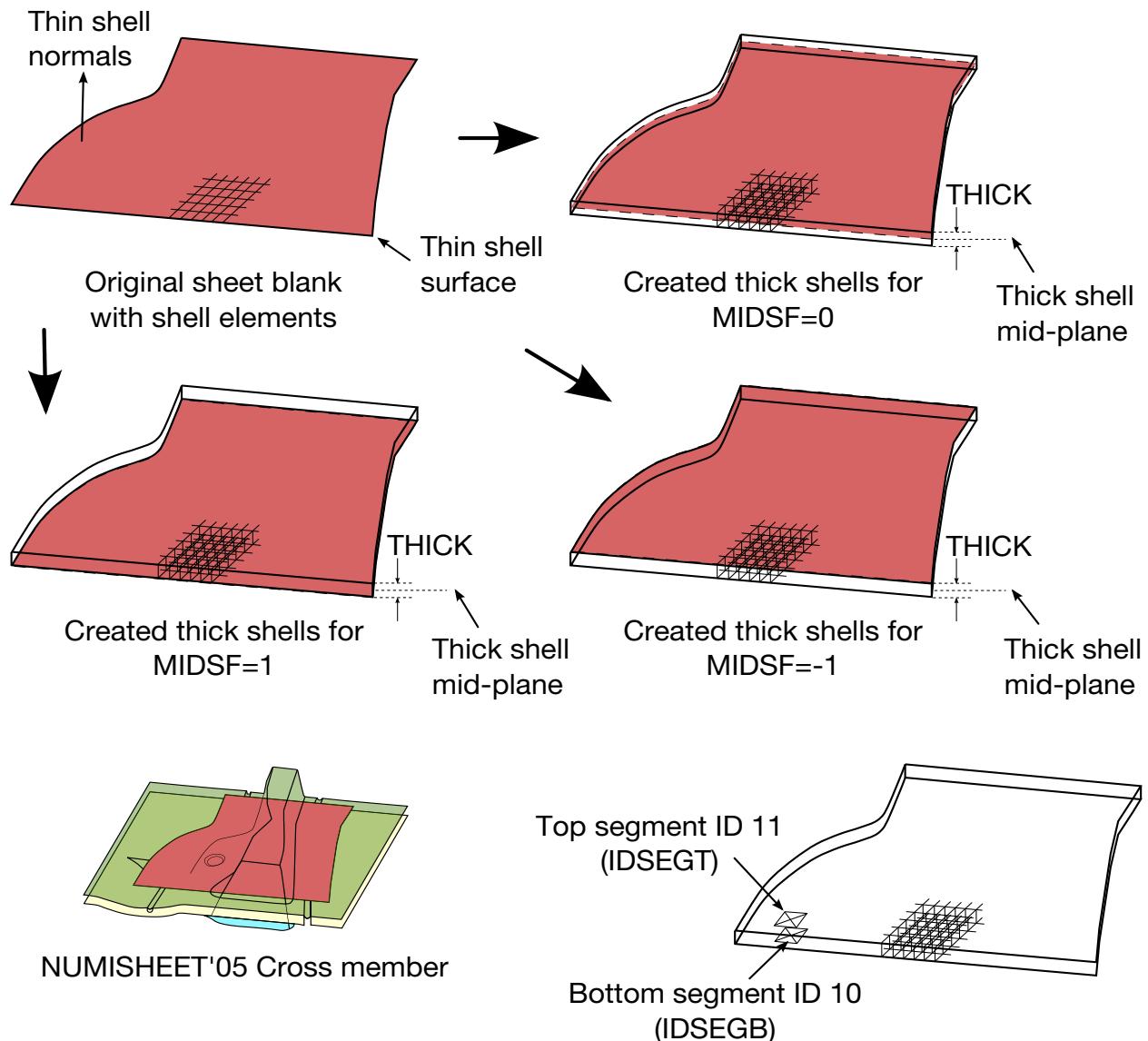


Figure 12-51. Converting thin shells to thick shells in sheet metal forming

***CONTROL_FORMING_STONING**

Purpose: Detect surface lows or surface defects formed during metal stamping. This calculation is typically performed after a springback simulation. A curvature-based method is implemented with the feature. Users have the option to check an entire part or just a few local areas, defined by node set or shell element set. In each area, direction of the stoning action can be specified or the program can automatically determine the stoning direction.

Card 1	1	2	3	4	5	6	7	8
Variable	ISTONE	LENGTH	WIDTH	STEP	DIRECT	REVERSE	METHOD	
Type	I	F	F	F	F	I	I	
Default	none	0.0	0.0	0.0	0.0	0	0	

Card 2	1	2	3	4	5	6	7	8
Variable	NODE1	NODE2	SID	ITYPE	V1	V2	V3	
Type	I	I	I	I	F	F	F	
Default	0	0	Rem 4	Rem 4	0.0	0.0	0.0	

VARIABLE	DESCRIPTION
ISTONE	Stoning calculation option. EQ.1: calculate panel surface quality using stoning method.
LENGTH	Length of the stone. See Remark 1 .
WIDTH	Width of the stone. See Remark 1 .
STEP	Stepping size of the moving stone. See Remark 1 .
DIRECT	Number of automatically determined stoning direction(s). See Remark 2 .

*CONTROL

*CONTROL_FORMING_STONING

VARIABLE	DESCRIPTION
REVERSE	Surface normal reversing option (see Remark 3): EQ.0: do not reverse surface normals. EQ.1: reverse surface normals.
METHOD	Stoning method. EQ.0: curvature-based method.
NODE1	Tail node defining stoning moving direction. See Remark 2 .
NODE2	Head node defining stoning moving direction. See Remark 2 .
SID	Node or shell set ID.
ITYPE	Set type designation: EQ.1: node set EQ.2: element set
V1, V2, V3	Vector components defining stoning direction (optional). See Remark 2 .

About Stoning:

Stoning is a quality checking process on class-A exterior stamping panels. Typically the long and wider surfaces of an oil stone of a brick shape are used to slide and scratch in a given direction against a localized area of concern on a stamped panel. Surface “lows” are shown where scratch marks are not visible and “highs” are shown in a form of scratch marks. This keyword is capable of predicting both the surface “lows” and “highs”. Since stoning process is carried out after the stamping (either drawn or trimmed) panels are removed from the stamping dies, a springback simulation needs to be performed prior to conducting a stoning analysis.

Remarks:

1. **Stone Reference Sizes.** As a reference, typical stone length and width can be set at 150.0 and 30.0 mm, respectively. The step size of the moving stone is typically set about the same order of magnitude of the element length. The smallest element length can be selected as the step size.
2. **Stoning Direction.** Stoning direction can be set in three different ways:

- a) The variables NODE1 and NODE2 are used to define a specific stoning direction. The stone is moved in the direction defined by NODE1 to NODE2.
 - b) Alternatively, one can leave NODE1 and NODE2 blank and define the number of automatically determined stoning directions by using the variable DIRECT. Any number can be selected but typically 2 is used. Although CPU time required for the stoning calculation is trivial, a larger DIRECT consumes more CPU time.
 - c) Furthermore, stoning direction can also be defined using a vector by defining the variables V1, V2, and V3.
3. **Element Normal Vectors.** Stoning is performed on the outward normal side of the mesh. Element normals must be consistent and oriented accordingly. Element normal can be automatically made consistent in LS-PrePost4.0 under *Ele-Tol/Normal* menu. Alternatively, the variable REVERSE provides in the solver an easy way to reverse a part with consistent element normals before the computation.
4. **Region for Stoning Analysis.** The blank model intended for analysis can be included using keyword *INCLUDE. If nothing is defined for SID and ITYPE, then the entire blank model included will be used for stoning analysis.
- A large area mesh can be included in the input file. An ELSET may also be included, which defines a local area that requires stoning computation. Alternatively, an ELSET can define several local areas to be used for the computation. Furthermore, an ELSET should not include meshes that have reversed curvatures. An ELSET can be easily generated using LS-PrePost4.0, under *Model → CreEnt → Cre → Set_Data → *SET_SHELL*.
5. **Modeling Guidelines.** Since stoning requires high level of accuracy in spring-back prediction, it is recommended that the SMOOTH option in keyword *CONTACT_FORMING_ONE WAY_SURFACE_TO_SURFACE to be used during the draw forming simulation. Not all areas require SMOOTH contact, only areas of interest may apply. In addition, meshes in the areas of concern need to be very fine, with average element size of 1 to 2 mm. Mesh adaptivity is not recommended in the SMOOTH/stoning areas. Also, mass scaling with DT2MS needs to be sufficiently small to reduce the dynamic effect during forming. For binder closing of large exterior panels, implicit static method using *CONTROL_IMPLICIT_FORMING type 2 is recommended, to further reduce potential buckles caused by the inertia effect.
6. **Stoning Output.** It is recommended that double precision version of LS-DYNA be used for this application. The output of the stoning simulation results is in a file named filename.output, where “filename” is the name of the LS-DYNA stoning input file containing this keyword, without the file extension. The stoning

results can be viewed using LS-PrePost4.0, under *MPost* → *FCOMP* → *Shell_Thickness*.

Example:

An example of a stoning analysis on a Ford Econoline door outer panel is provided for reference. The original part model comes from National Crash Analysis Center at The George Washington University. The original part was modified heavily in LS-PrePost4.0 for demonstration purposes. Binder and addendum were created and sheet blank size was assumed. The blank is assigned 0.65 mm thickness and a BH210 properties with *MAT_037. Shell thickness contour plots for the drawn and trimmed panels are shown in [Figures 12-52](#) and [12-53](#), respectively. Springback amount in the Z-direction is plotted in [Figure 12-54](#). The complete input deck used for the stoning simulation is provided below for reference; where, a local area mesh of the door handle after springback simulation Doorhandle.k and an element set elset1.k are included in the deck. Locations of the ELSETs are defined for the upper right ([Figure 12-55](#) left) and lower right corners ([Figure 12-56](#) left) of the door handle, where “mouse ears” are expected.

```
*KEYWORD
*TITLE
Stoning Analysis
*INCLUDE
Doorhandle.k
*INCLUDE
elset1.k
*CONTROL_FORMING_STONING
$ ISTONE LENGTH WIDTH STEP DIRECT REVERSE METHOD
   1    150.0    4.0   1.0      9       0       0
$ NODE1     NODE2     SID ITYPE
                  1        2
*END
```

Stoning results are shown in [Figures 12-55](#) (right) and [12-56](#) (right) for the upper right and lower right corners, respectively. “Mouse ears” are predicted where anticipated.

Revision Information:

The stoning feature is available in LS-DYNA Revision 54398 and later releases. Vector component option is available in Revision 60829 and later releases.

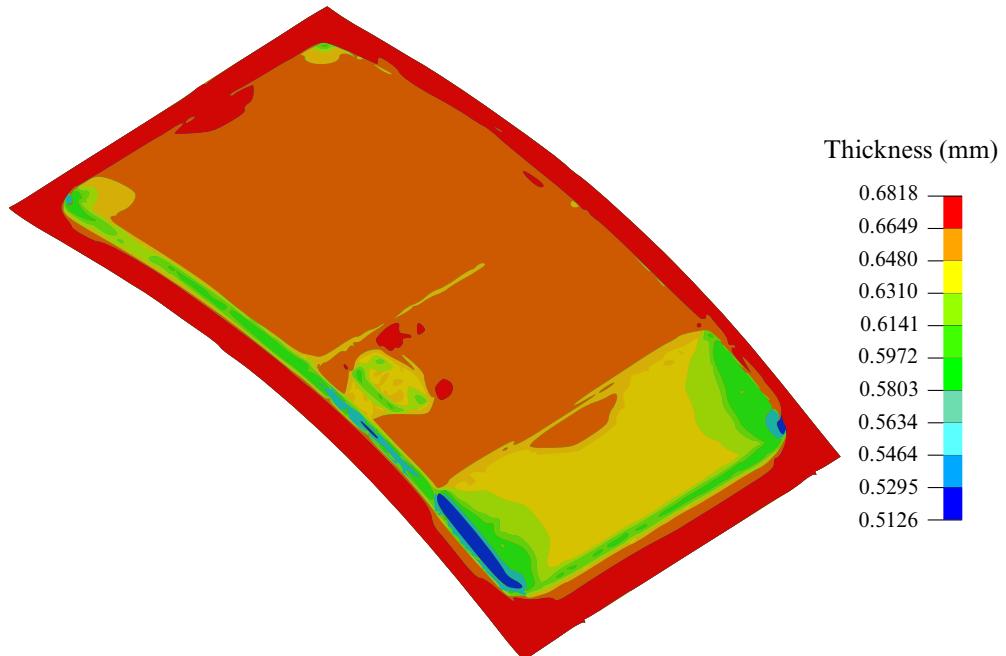


Figure 12-52. Thickness contour of the panel after draw simulation.

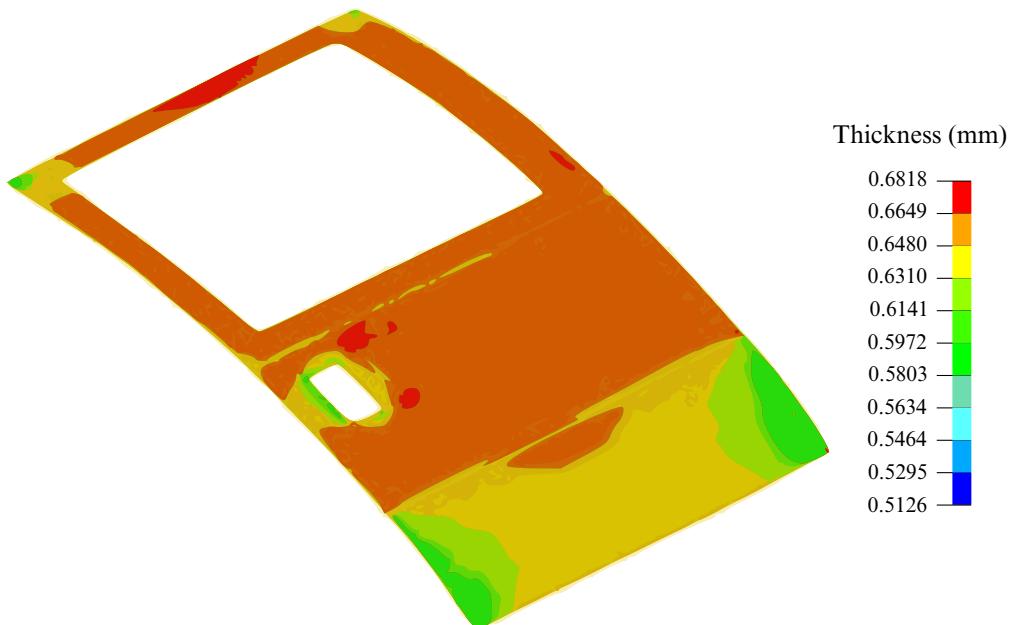


Figure 12-53. Thickness contour of the panel after trimming.

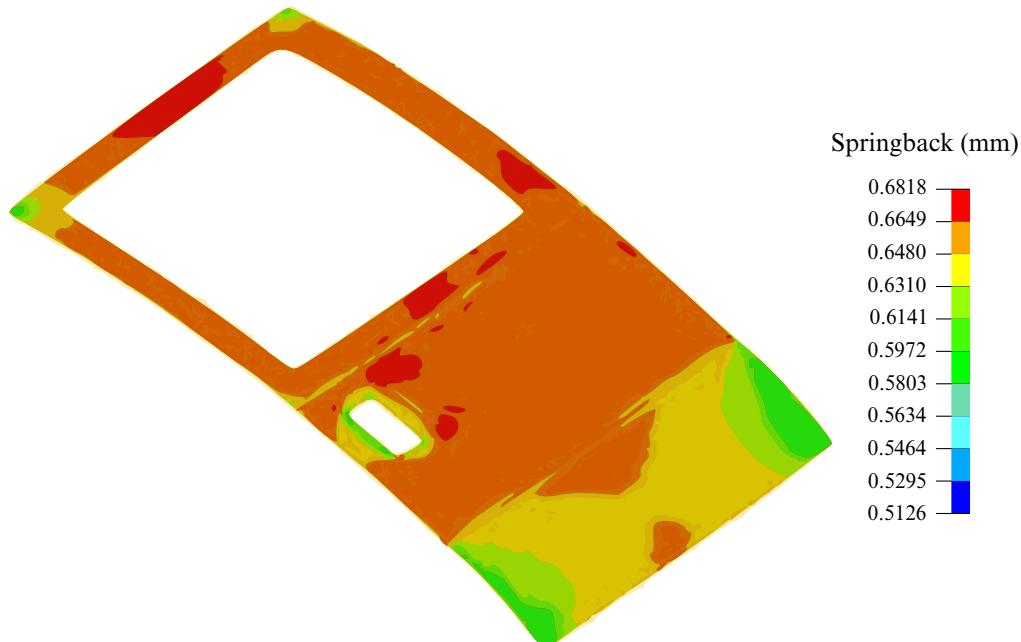


Figure 12-54. Springback amount (mm).



Figure 12-55. Stoning simulation for the upper right door corner.

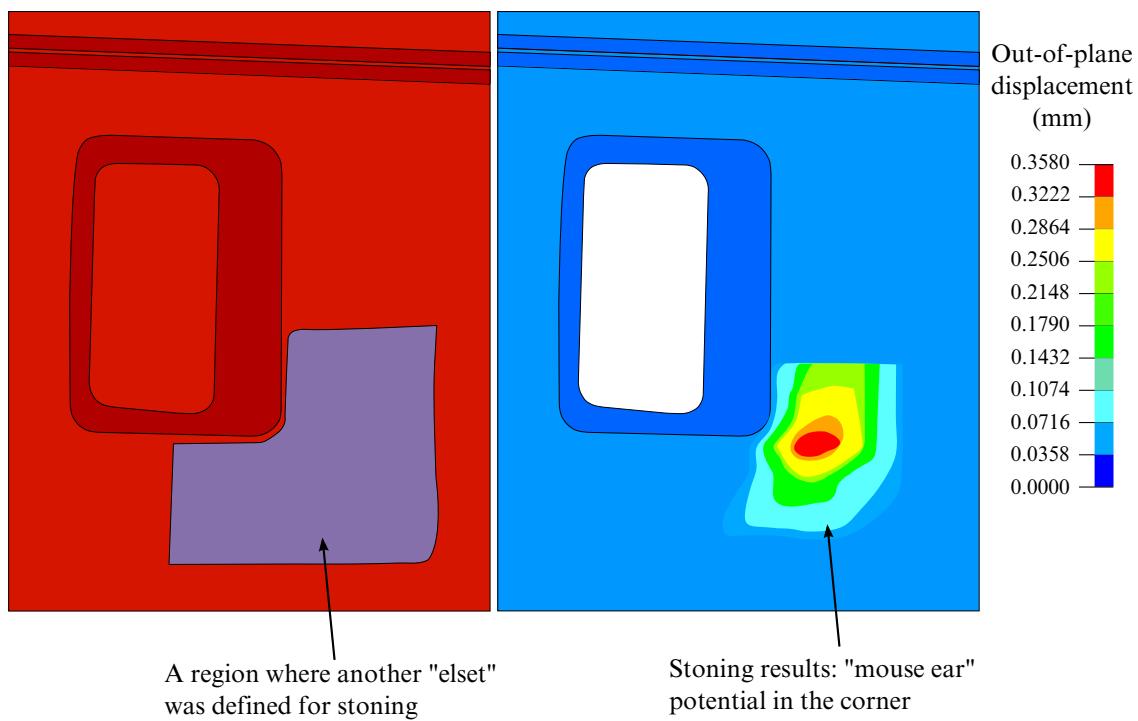


Figure 12-56. Stoning simulation for the lower right door corner.

*CONTROL

*CONTROL_FORMING_STRAIN_RATIO_SMOOTH

*CONTROL_FORMING_STRAIN_RATIO_SMOOTH

Purpose: In explicit methods, the output strain path looks smooth, but the strain incremental ratio, β ($= d\epsilon_2/d\epsilon_1$) can oscillate significantly during each step and will result in significant error in formability prediction. This keyword uses a smoothing algorithm to smooth the beta value when calculating the Formability Index, which is designed to better calculate formability by considering a non-linear strain path.

This keyword must be used with the NLP option in *MAT_36, *MAT_37, *MAT_125, or *MAT_226 for shell elements only. It is jointly developed with Ford Motor Company.

Card 1	1	2	3	4	5	6	7	8
Variable	DT/CYCLE	WEIGHT						
Type	F	F						
Default	none	none						

VARIABLE

DESCRIPTION

DT/CYCLE

Flag for output option (time interval or cycle number).

LT.0: the absolute value is the time interval between outputs.

GT.0: number of cycles between outputs

WEIGHT

Coefficient α in equation below.

Remarks:

The incremental change of in-plane major and minor strains are smoothed according to the following formula:

$$\begin{aligned}\Delta\epsilon_{1(n-1)} \times (1 - \alpha) + d\epsilon_{1(n)} \times \alpha \\ \Delta\epsilon_{2(n-1)} \times (1 - \alpha) + d\epsilon_{2(n)} \times \alpha\end{aligned}$$

where, $d\epsilon_{1(n)}$ and $d\epsilon_{2(n)}$ are incremental changes of ϵ_1 and ϵ_2 in the current time step n , $\Delta\epsilon_{1(n-1)}$ and $\Delta\epsilon_{2(n-1)}$ are incremental changes of ϵ_1 and ϵ_2 in the previous time step $n - 1$. The weighting coefficient α regulates the smoothness of the incremental changes in ϵ_1 and ϵ_2 .

With the smoothed incremental major and minor strains, β is:

$$\beta = \frac{\Delta\epsilon_{2(n-1)} \times (1 - \alpha) + d\epsilon_{2(n)} \times \alpha}{\Delta\epsilon_{1(n-1)} \times (1 - \alpha) + d\epsilon_{1(n)} \times \alpha} .$$

The lower limit of β is the minimum strain ratio defined in the FLD curve.

Note that β is stored in history variable #2.

A partial keyword deck that illustrates using this keyword is shown below. The material model is *MAT_36. Note NEIPS is set to 3 so three history variables that include formability index (F.I.), strain ratio β and effective plastic strain $\bar{\epsilon}$ are output.

```
*DATABASE_EXTENT_BINARY
$    NEIPH      NEIPS      MAXINT      STRFLG      SIGFLG      EPSFLG      RLTFLG      ENGFLG
      3          &nip          1
*CONTROL_FORMING_STRAIN_RATIO_SMOOTH
$ DT/CYCLE      WEIGHT      OUTPUT
      -0.001      0.35          1
*MAT_3-PARAMETER_BARLAT_NLP
$      MID      RO      E      PR      HR
      13      7.8E-09      2.07E+05      0.30      3.000
$      M      R00      R45      R90      LCID
      6.000      1.200      1.450      1.090      99
$      AOPT      C      P      VLCID
      2.000
$                                A1      A2      A3
$                                1.000      0.000      0.000
$      V1      V2      V3      D1      D2      D3
$                                0.000      1.000      0.000
*DEFINE_CURVE
      200
$FLD
      -0.7000      0.8309
      -0.4500      0.6805
      -0.2500      0.5081
      0.0000      0.2479
      0.2000      0.3487
      0.4000      0.3845
```

*CONTROL

*CONTROL_FORMING_TEMPLATE

*CONTROL_FORMING_TEMPLATE

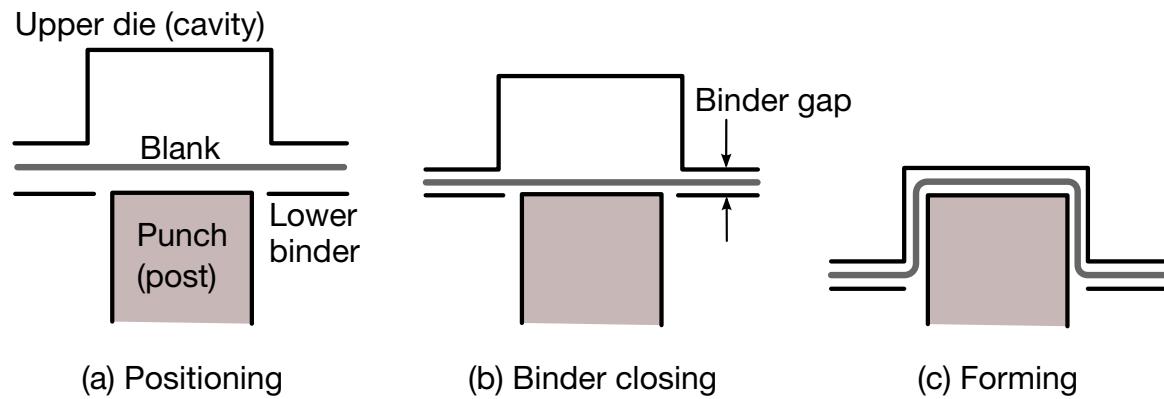
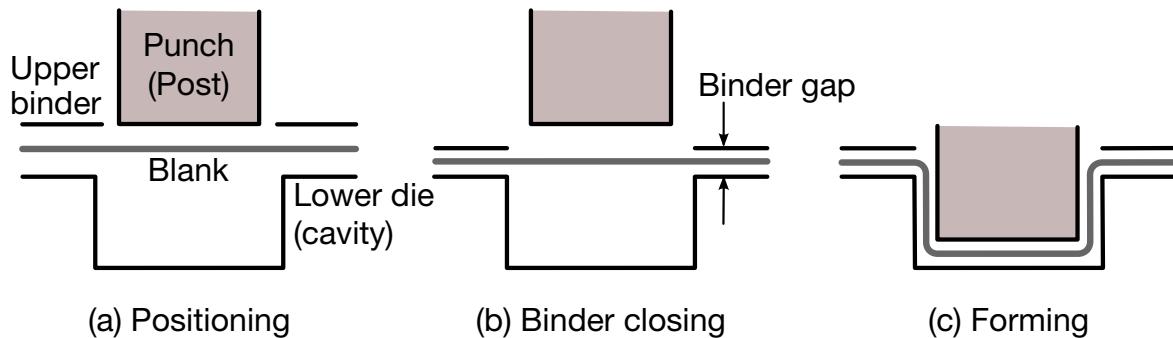
Purpose: This keyword is used to simplify the required input for sheet metal stamping simulations. With this keyword, five templates are given: three-piece air draw, three-piece toggle draw, four-piece stretch draw, trimming, and springback.

NOTE: This option has been deprecated in favor of *CONTROL_FORMING_AUTOPOSITION_PARAMETER.

Card 1	1	2	3	4	5	6	7	8
Variable	IDTEMP	BLKID	DIEID	PNCH	BNDU	Bndl	TYPE	PREBD
Type	I	I	I	I	I	I	I	F
Default	none	none	none	none	none	none	0	0.0

Card 2	1	2	3	4	5	6	7	8
Variable	LCSS	AL/FE	R00	R45	R90	E	DENSITY	PR
Type	I	C	F	F	F	F	F	F
Default	none	F	1.0	R00	R00	none	none	none

Card 3	1	2	3	4	5	6	7	8
Variable	K	N	MTYP	UNIT	THICK	GAP	FS	
Type	F	F	I	I	F	F	F	
Default	none	none	37	1	none	1.1t	0.1	

**Figure 12-57.** IDTEMP = 1: forming in 3-piece air draw.**Figure 12-58.** IDTEMP = 2: forming in 3-piece toggle draw.

Card 4	1	2	3	4	5	6	7	8
Variable	PATERN	VMAX	VX	VY	VZ	VID	AMAX	
Type	I	F	F	F	F	I	F	
Default	1	1000	0	0	-1	-z	10^6	

Card 5	1	2	3	4	5	6	7	8
Variable	LVLADA	SIZEADA	TIMSADA	D3PLT				
Type	I	F	I	I				
Default	1	none	20	10				

*CONTROL

*CONTROL_FORMING_TEMPLATE

VARIABLE	DESCRIPTION
IDTEMP	Type of forming process (see About IDTEMP below): EQ.1: 3-piece air-draw (Figure 12-57) EQ.2: 3-piece Toggle-draw (Figure 12-58) EQ.3: 4-piece stretch draw (Figure 12-60) EQ.4: springback EQ.5: trimming
BLKID	Part or part set ID (see TYPE) that defines the blank.
DIEID	Part or part set ID that defines the die. See Figures 12-57, 12-58 and 12-60 for more information
PNCHID	Part or part set ID that defines the punch.
BNDUID	Part or part set ID that defines the upper binder.
BNDLID	Part or part set ID that defines the lower binder.
TYPE	Flag for part or part set ID used in the definition of BLKID, DIEID, PNCHID, BNDUID, and BNDLID: EQ.0: Part ID EQ.1: Part set ID
PREBD	“Pull-over” distance, for 4 piece stretch draw only. This is the travel distance of both upper and lower binder together after they are fully closed. Typically, this distance is below 50 mm. See Figure 12-60 for more information.
LCSS	If the material (*MAT_XXX) for the blank is not defined, this curve ID will define the stress-strain relationship; otherwise, this curve is ignored.
AL/FE	This parameter is used to define the Young’s Modulus and density of the blank. If this parameter is defined, E and DENSITY will be defined in the units given by Table 12-59 . EQ.A: the blank is aluminum EQ.F: the blank is steel (default)
R00, R45, R90	Material anisotropic parameters. For transversely anisotropy the R value is set to the average value of R00, R45, and R90.

VARIABLE	DESCRIPTION
E	Young's Modulus. If AL/FE is user defined, E is unnecessary.
DENSITY	Material density of blank. If AL/FE is user defined, this parameter is unnecessary.
PR	Poisson's ratio.
K	Strength coefficient for exponential hardening ($\bar{\sigma} = k\bar{\varepsilon}^n$). If LCSS is defined, or if a blank material is user defined by *MAT_XXX, this parameter is ignored.
N	Exponent for exponential hardening ($\bar{\sigma} = k\bar{\varepsilon}^n$). If LCSS is defined, or if a blank material user defined, this parameter is ignored.
MTYP	Only material models *MAT_036 and *MAT_037 are supported.
UNIT	Define a number between 1 and 10 (Table 12-59) to indicate the UNIT used in this simulation. This unit is used to obtain proper punch velocity, acceleration, time step, and material properties.
THICK	Blank thickness. If the blank thickness is already defined with *SECTION_SHELL, this parameter is ignored.
GAP	The gap between rigid tools at their home position. If *BOUNDARY_PRESCRIBED_RIGID_BODY is user defined, this parameter is ignored. The default is 1.1 x blank thickness.
FS	Friction coefficient (default is 0.10). If the contact (*CONTACT) is user defined, this parameter is ignored.
PATTERN	Velocity profile of moving tool. If the velocity is user defined by *BOUNDARY_PRESCRIBED_RIGID_BODY, PATTERN is ignored. EQ.1: Ramped velocity profile EQ.2: Smooth velocity curve
VX, VY, VZ	Vector components defining the direction of the punch movement. The default direction is defined by VID.
VID	Vector ID defining the direction of the punch movement. This variable overrides the vector components (VX,VY,VZ). If VID and (VX,VY,VZ) are undefined, the punch is assumed to move in the negative z-direction.
AMAX	The maximum allowable acceleration.

VARIABLE	DESCRIPTION
LVLADA	Maximum adaptive level.
SIZEADA	Minimum element size permitted in the adaptive mesh.
TIMSADA	Total number of adaptive steps during the forming simulation.
D3PLT	The total number of output states in the D3PLOT database.

Applicable units:

UNIT	1	2	3	4	5	6	7	8	9	10
Mass	Ton	Gm	Gm	Gm	Gm	Kg	Kg	Kg	Kg	Kg
Length	Mm	Mm	Mm	Cm	Cm	Mm	Cm	Cm	Cm	m
Time	S	Ms	S	Us	S	Ms	Us	Ms	S	S
Force	N	N	μ N	1e7N	Dyne	KN	1e10N	1e4N	1e-2N	N

Table 12-59. Available units for metal stamping simulation.**About IDTEMP:**

When the variable IDTEMP is set to 1, it represents a 3-piece draw in air, as shown in [Figure 12-57](#). When IDTEMP is set to 2, a 3-piece toggle draw is assumed, as in [Figure 12-58](#). For IDTEMP of 1 or 2, LS-DYNA will automatically position the tools and minimize the punch travel (step a), calculate the binder and punch travel based on the blank thickness and the home gap (step b), set the termination time based on steps (a) and (b), define the rigid body motion of the tooling, establish all the contacts between the blank and rigid tools, and, select all necessary control parameters.

When IDTEMP is set to 3, a 4-piece stretch draw shown in [Figure 12-60](#) will be followed. The die action goes as follows: after upper binder moves down to fully close with lower binder, both pieces move together down a certain distance (usually ~50 mm) to “pull” the blank “over” the lower die; then the upper punch closes with the lower die; and finally the binders move down together to their home position.

Both toggle draw and 4-piece stretch draw are called “double action” processes which suffer from a slower stamping speed. As the metric of “hits per minute” (or “parts per minute”) becomes a stamping industry benchmark for efficiency, these types of draw are becoming less popular (especially the 4-piece stretch draw). Nevertheless, they remain

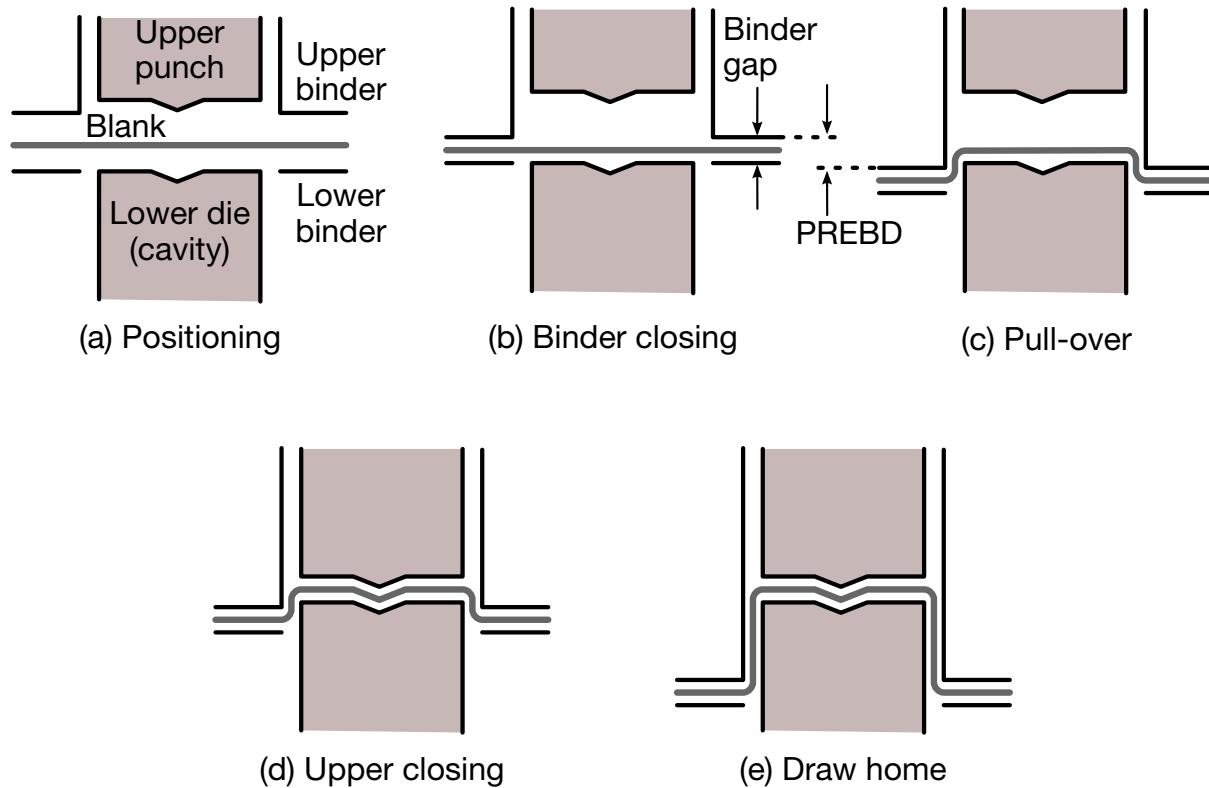


Figure 12-60. IDTEMP = 3: forming in 4-piece stretch draw.

important stamping processes for controlling wrinkles in difficult-to-form panels such as lift gate inners, door inners and floor pans. These two processes are also used in situation where deep drawn panels require draw depth of over 250 mm, the usual limit for automatic transfer presses.

For all the above IDTEMP values, users do not need to define additional keywords, such as *PART, *CONTROL, *SECTION, *MAT_..., *CONTACT_... (drawbead definition is an exception), and *BOUNDARY_PRESCRIPTION_RIGID. If any such keyword is defined, automatic default settings will be overridden.

When IDTEMP is set to 4, a springback simulation will be conducted. The only additional keyword required is *BOUNDARY_SPC_... to specify the constraints in the input deck.

When IDTEMP is set to 5, a trimming operation will be performed. The only additional keyword required is *DEFINE_CURVE_TRIM to specify the trim curves in the input deck.

Revision information:

This feature is available starting in Revision 45901 and later releases.

*CONTROL

*CONTROL_FORMING_TIPPING

*CONTROL_FORMING_TIPPING

Purpose: Reorient or reposition a part between the stamping dies. In a stamping line die simulation, panel tipping and translation between the die stations are frequently required. Typically, such transformations involve only a small rotation (less than 15 degrees) but a large translation. For example, a part may need to be tipped by an angle of 10 degrees along the Y-axis and translated 2000 mm along the X-axis between the current trimming die and next flanging die.

Card Summary:

Card Sets. For each rotated or translated part or set add one Card 1 (Tipping Card). If NMOVE ≠ -6, include NMOVE of Cards 2a and 2b combined. If NMOVE = -6, include Cards 2c.1 through 2c.6. The data set for this keyword ends at the next keyword ("*") card.

Card 1. This card is required.

PID/SID	ITYPE	ISTRAN	IFSTRSS	NMOVE			
---------	-------	--------	---------	-------	--	--	--

Card 2a. If NMOVE ≠ -6, this card may be included. It is recognized when the first field (ROT/TRAN) is 1. It defines a rotation transformation. Include a combination of this card and Card 2b to get NMOVE transformation cards.

ROT/TRAN	V11	V12	V13	X01	Y01	Z01	DISTA1
----------	-----	-----	-----	-----	-----	-----	--------

Card 2b. If NMOVE ≠ -6, this card may be included. It is recognized when the first field (ROT/TRAN) is 2. It defines a translation transformation. Include a combination of this card and Card 2a to get NMOVE transformation cards.

ROT/TRAN	DX	DY	DZ				
----------	----	----	----	--	--	--	--

Card 2c.1. This card is included if and only if NMOVE = -6.

P1	X1	Y1	Z1				
----	----	----	----	--	--	--	--

Card 2c.2. This card is included if and only if NMOVE = -6.

P2	X2	Y2	Z2				
----	----	----	----	--	--	--	--

Card 2c.3. This card is included if and only if NMOVE = -6.

P4	X3	Y3	Z3				
----	----	----	----	--	--	--	--

Card 2c.4. This card is included if and only if NMOVE = -6.

P4	X4	Y4	Z4				
----	----	----	----	--	--	--	--

Card 2c.5. This card is included if and only if NMOVE = -6.

P5	X5	Y5	Z5			
----	----	----	----	--	--	--

Card 2c.6. This card is included if and only if NMOVE = -6.

P6	X6	Y6	Z6			
----	----	----	----	--	--	--

Data Card Definitions:

Tipping Card. Specify a part or set ID to be tipped.

Card 1	1	2	3	4	5	6	7	8
Variable	PID/SID	ITYPE	ISTRAN	IFSTRSS	NMOVE			
Type	I	I	I	I	I			
Default	none	none	0	0	0			

VARIABLE	DESCRIPTION
PID/SID	Part ID or part set ID of part(s) that requires tipping and/or translation.
ITYPE	Part ID or part set ID indicator: EQ.1: PID means part ID, EQ.2: PID/SID means part set ID.
ISTRAN	Strain tensors inclusion option: EQ.1: include in tipping/translation.
IFSTRSS	Stress tensors inclusion option: EQ.1: include in tipping/translation.
NMOVE	Total number of tipping and translation transformations intended for this part/part set. However, when NMOVE: EQ.-6: Transformation is done based on user-specified coordinates of 3 points on the source, and the corresponding point coordinates on the target.

*CONTROL

*CONTROL_FORMING_TIPPING

Move Card 2a (Rotation). Include when first entry, ROT/TRAN, is set to 1.

Card 2a	1	2	3	4	5	6	7	8
Variable	ROT/TRAN	V11	V12	V13	X01	Y01	Z01	DSTA1
Type	I	F	F	F	F	F	F	F
Default	none	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Move Card 2b (Translation). Include when first entry, ROT/TRAN, is set to 2.

Card 2b	1	2	3	4	5	6	7	8
Variable	ROT/TRAN	DX	DY	DZ				
Type	I	F	F	F				
Default	none	0.0	0.0	0.0				

VARIABLE	DESCRIPTION
ROT/TRAN	Transformation type: EQ.1: rotation, EQ.2: translation.
V11, V12, V13	Vector components of an axis about which tipping is performed.
X01, Y01, Z01	X, Y, and Z coordinates of a point through which the tipping axis passes.
DSITA	Tipping angle in degrees. See Remark 2 .
DX, DY, DZ	Translation distances along global X-axis, Y-axis and Z-axis.

CONTROL_FORMING_TIPPING**CONTROL**

Move Card 2c.1 (3-point Transformation). Include when NMOVE is set to -6. (I8, 3E16.0)

Card 2c.1	1	2	3	4	5	6	7	8	9	10
Variable	P1		X1		Y1		Z1			
Type	I		F		F		F			
Default	none		0.0		0.0		0.0			

Move Card 2c.2 (3-point Transformation). Include when NMOVE is set to -6. (I8, 3E16.0)

Card 2c.2	1	2	3	4	5	6	7	8	9	10
Variable	P2		X2		Y2		Z2			
Type	I		F		F		F			
Default	none		0.0		0.0		0.0			

Move Card 2c.3 (3-point Transformation). Include when NMOVE is set to -6. (I8, 3E16.0)

Card 2c.3	1	2	3	4	5	6	7	8	9	10
Variable	P3		X3		Y3		Z3			
Type	I		F		F		F			
Default	none		0.0		0.0		0.0			

*CONTROL

*CONTROL_FORMING_TIPPING

Move Card 2c.4 (3-point Transformation). Include when NMOVE is set to -6. (I8, 3E16.0)

Card 2c.4	1	2	3	4	5	6	7	8	9	10
Variable	P4	X4		Y4	Z4					
Type	I	F		F	F					
Default	none	0.0		0.0	0.0					

Move Card 2c.5 (3-point Transformation). Include when NMOVE is set to -6. (I8, 3E16.0)

Card 2c.5	1	2	3	4	5	6	7	8	9	10
Variable	P5	X5		Y5	Z5					
Type	I	F		F	F					
Default	none	0.0		0.0	0.0					

Move Card 2c.6 (3-point Transformation). Include when NMOVE is set to -6. (I8, 3E16.0)

Card 2c.6	1	2	3	4	5	6	7	8	9	10
Variable	P6	X6		Y6	Z6					
Type	I	F		F	F					
Default	none	0.0		0.0	0.0					

VARIABLE

DESCRIPTION

P1, P2, P3

Unique point IDs from the target, where it will be transformed to.

X1, Y1, Z1

X2, Y2, Z2

X3, Y3, Z3

X, Y, and Z coordinates of the points P_1 , P_2 , and P_3 from the target, respectively.

VARIABLE	DESCRIPTION
P4, P5, P6	Unique point IDs from the source, where it will be transformed from.
X4, Y4, Z4 X5, Y5, Z5 X6, Y6, Z6	X, Y, and Z coordinates of the points P_4 , P_5 , and P_6 from the source, respectively. Note that points P_4 , P_5 , and P_6 correspond to points P_1 , P_2 , and P_3 , respectively.

Remarks:

- Include Keyword.** Keyword *INCLUDE can be used to include the file to be tipped or translated.
- Tipping Angle.** Tipping angle DISTA1 is defined in degrees. Signs of the tipping angles follow the right-hand rule.
- NMOVE ≠ -6 Example.** The keyword input below tips a part by $+23.0^\circ$, -31.0° , and $+8.0^\circ$ about the X-, Y-, and Z-axes that pass through the origin, respectively. The part also translates 12.0 mm, -6.0 mm and 91.0 mm along X-, Y-, and Z-axes, respectively.

```
*INCLUDE
trimmedpart.dynain
*CONTROL_FORMING_TIPPING
$ PID/PSID      ITYPE     ISTRAIN     ISTRSS      NMOVE
          1          0          1          1          4
$ ROT/TRAN      V11       V12       V13       X01       y01       z01      DSITA1
          1        1.000   0.000000   0.000   0.000   0.000   0.000   23.0
$ ROT/TRAN      V21       V22       V23       X21       y21       z21      DSITA2
          1        0.000   1.000000   0.000   0.000   0.000   0.000   -31.0
$ ROT/TRAN      V31       V32       V33       X31       y31       z31      DSITA3
          1        0.000000   0.000   1.000   0.000   0.000   0.000    8.0
$ ROT/TRAN      DX         DY         DZ
          2        12.0     -6.0      91.0
```

- NMOVE = -6 Example.** An example of the keyword input for NMOVE = -6 is included below. The part tips based on 3-points from the source and the corresponding 3-points from the target.

```
$-----1-----2-----3-----4-----5-----6-----7-----8
*CONTROL_FORMING_TIPPING
$ PID/PSID      ITYPE     ISTRAIN     ISTRSS      NMOVE
          2          1          1          1          -6
$ target coordinates (I8, 3E16.0):
          4        20.1878     -55.4753    -30.5631
          5        0.00092581    -55.4871    -30.5094
          6        7.61563E-4     -76.1135    -33.6276
$ source coordinates (I8, 3E16.0):
          1        20.1878     -55.4753    -30.5631
          2        0.00925064    -55.5290    -30.5749
          3        0.000757217   -52.4108    -51.2013
```

Revision Information:

This feature is available starting in LS-DYNA Dev53448, with major updates from Dev80261. It is also available in LS-PrePost *eZ-Setup* for metal forming applications (<http://ftp.lstc.com/anonymous/outgoing/lsprepost/>). Transformations based on 3-points (NMOVE = -6) is available in Dev123450.

***CONTROL_FORMING_TRAVEL**

Purpose: This keyword allows user to define tool travel for each phase in a stamping process simulation. The entire simulation process can be divided into multiple phases corresponding to the steps of an actual metal forming process. This keyword is to be used for tools that are pre-positioned above the sheet blank (or below the blank) and ready for forming. For tools that are pre-positioned at their home positions, *CONTROL_FORMING_TRAVEL should be used. This keyword is used together with *CONTROL_FORMING_USER.

NOTE: This option has been deprecated in favor of *CONTROL_FORMING_AUTOPOSITION_PARAMETER).

Define Travel Cards. Repeat Card as many times as needed to define travels in multiple phases. The next “*” card terminates the input.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	VID	TRAVEL	TARGET	GAP	PHASE	FOLLOW	
Type	I	I	F	I	F	I	I	
Default	none	none	none	none	1.0t	none	none	

VARIABLE	DESCRIPTION
PID	Part ID of a stamping tool, as defined in *PART.
VID	Vector ID defining the direction of travel for the tool defined by the PID.
TRAVEL	The distance in which the tool will be traveled to complete forming in the direction specified by the VID. If TRAVEL is defined, it is unnecessary to define TARGET.
TARGET	Target tool PID, as defined in *PART. The tool (defined by PID) will be traveled to where the TARGET tool is to complete forming.
GAP	The minimum distance between the tool (PID) and TARGET tool at the home position (forming complete). The GAP is by default the sheet blank thickness “t”.

***CONTROL**

***CONTROL_FORMING_TRAVEL**

VARIABLE	DESCRIPTION
PHASE	Phase number, starting sequentially from 1. For example, phase 1 is the binder closing, and phase 2 is the drawing operation.
FOLLOW	Part ID of a stamping tool to be followed by the tool (PID). When this variable is defined, the distance between the tool (PID) and part ID defined by FOLLOW, will remain constant during the phase.

Remarks:

FOLLOW can be used to reduce total simulation time. For example, in a toggle draw, the upper punch travels together with the upper binder during binder closing phase, thus reducing the upper travel distance during the draw, shortening the overall termination time.

An example is provided in manual pages under *CONTROL_FORMING_USER.

***CONTROL_FORMING_TRIM_MERGE**

Purpose: This feature allows for automatic close of any open trim loop curve for a successful trimming simulation. Previously, sheet metal trimming would fail if a trim curve does not form a closed loop. This keyword is used together with *DEFINE_CURVE_TRIM, *ELEMENT_TRIM, *DEFINE_VECTOR, *CONTROL_ADAPTIVE_CURVE, and *CONTROL_CHECK_SHELL. It applies to shell elements only.

Card 1	1	2	3	4	5	6	7	8
Variable	IMERGE	GAPM						
Type	I	F						
Default	1	0.0						

VARIABLE	DESCRIPTION
IMERGE	Activation flag. Set to '1' (default) to activate this feature.
GAPM	Gap distance between two open ends of a trim loop curve in the model. If the gap is smaller than GAPM, the two open ends of a trim curve will be closed automatically.

Remarks:

If multiple open trim loop curves exist, GAPM should be set to a value larger than any of the gap distances of any trim curves in the trim model.

Example:

The example provided below includes two trim curves, a three-dimensional curve (#90905) with an open gap of 2.3 mm and a two-dimensional trim curve (#90907) with an open gap 2.38 mm. An automatic merge operation is being performed with the GAPM set at 2.39 mm. Since this set value is larger than both gaps in the model, trimming will automatically close the gap for both curves to form two closed-loop curves for a successful trim. In [Figure 12-61](#), two different trimming results are illustrated with GAPM of 2.39 (successful) as well as 2.37 (fail).

```
*KEYWORD
*INCLUDE_TRIM
drawn2.dynain
:
*CONTROL_ADAPTIVE_CURVE
```

*CONTROL

*CONTROL_FORMING_TRIM_MERGE

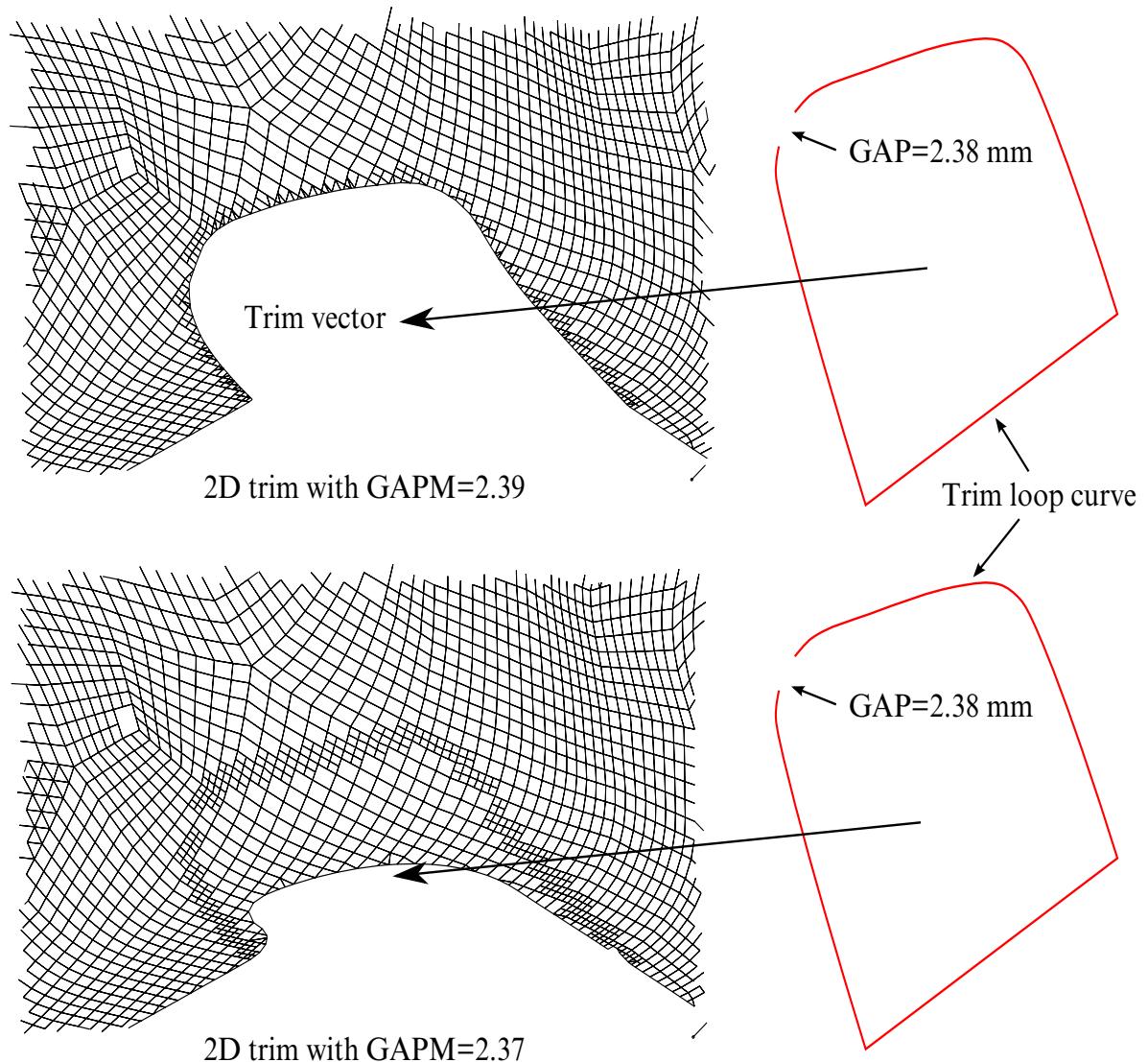


Figure 12-61. A 2D trimming with different GAPM values.

```
$ IDSET ITYPE N SMIN
  &blkSID 2 3 0.6
*CONTROL_CHECK_SHELL
$ PSID IFAUTO CONVEX ADPT ARATIO ANGLE SMIN
  &blkSID1 1 1 1 0.250000150.000000 0.000000
*DEFINE_CURVE_TRIM_3D
$# tcid tctype tflg tdir tctol tolN nseed1 nseed2
  90907 2 1 0 1.250000 2.500000 0 0
sim_trimline_03.igs
*DEFINE_CURVE_TRIM_NEW
$# tcid tctype tflg tdir tctol tolN nseed1 nseed2
  90905 2 0 2 1.250000 1.000000 0 0
$# filename
sim_trimline_02.igs
*DEFINE_VECTOR_TITLE
vector for Trim curve 90905
$# vid xt yt zt xh yh zh cid
  2 0.000 0.000 0.000 -0.170000 0.950000 -0.260000 0
*ELEMENT_TRIM
&blkSID
*DEFINE_TRIM_SEED_POINT_COORDINATES
```

```
$ NSEED,X1,Y1,Z1,X2,Y2,Z2
1,&seedx,&seedy,&seedz
$-----1-----2-----3-----4-----5-----6-----7-----8
*CONTROL_FORMING_TRIM_MERGE
$    IMERGE      GAPM
      1        2.39
$ Note that the 3D trim curve has a gap of 2.3 and the 2D trim curve has a gap of 2.38
*END
```

Revisions:

This feature is available starting in LS-DYNA Revision 84098.

*CONTROL

*CONTROL_FORMING_TRIM_SOLID_REFINEMENT

*CONTROL_FORMING_TRIM_SOLID_REFINEMENT

Purpose: Activate adaptive refinement along the trim curve for trimming a multi-layered sandwiched part.

Card 1	1	2	3	4	5	6	7	8
Variable	IREFINE	ILEVEL						
Type	I	I						
Default	1	0						

VARIABLE

DESCRIPTION

IREFINE

A flag to activate the adaptive trimming of a multi-layer sandwiched part. Currently setting this to either 0 or 1 will turn on the adaptive trimming.

EQ.1: Activate the adaptive trimming.

ILEVEL

Adaptive refinement level. Currently setting this variable to any integer other than 0 will refine the mesh one level down along the trim curve.

EQ.0: No refinement

EQ.1: Refine one level down.

Example:

A partial keyword input example is shown below where the included file incoming.dynain is a sandwiched part (ITYP = 1). The part is to be trimmed with adaptive refinement one level down (IREFINE = 1, ILEVEL = 2) along a 2D trim curve in the global X-direction that is defined in trimcurves2d.iges.

```
*Keyword
*Include_trim
incoming.dynain
*CONTROL_FORMING_TRIMMING
$      PSID          ITYP
      11            1
*CONTROL_FORMING_TRIM_SOLID_REFINEMENT
$  irefine    ilevel
      1            2
*SET_PART_LIST
11
100,101,102
```

```
*DEFINE_CURVE_TRIM_NEW
$#      tcid      tctype      tflg      tdir      tctol      tolN      nseed1      nseed2
      2          2          0          1      0.10000    1.000000      0          0
$# filename
trimcurves2d.iges
*DEFINE_VECTOR_TITLE
vector for Trim 1
$#      vid      xt      yt      zt      xh      yh      zh      cid
      1      0.000    0.000    0.000    1.000    0.000    0.000000      0
*DEFINE_TRIM_SEED_POINT_COORDINATES
$ NSEED,X1,Y1,Z1,X2,Y2,Z2
1,-69.5309,114.833,49.55
```

Note that for sandwiched part trimming, the keyword *INCLUDE_TRIM (not *INCLUDE) *must* be used to include the dynain file to be trimmed.

Revision information:

1. This feature is available starting in Dev 134513, in SMP and MPP.

*CONTROL

*CONTROL_FORMING_TRIMMING

*CONTROL_FORMING_TRIMMING

Purpose: Define a part subset to be trimmed by *DEFINE_CURVE_TRIM. This feature is intended for metal forming simulations. Currently trimming is enabled for shell elements, solid elements, adaptive sandwiched parts (one layer of solid elements with top and bottom layers of shell elements), non-adaptive sandwiched parts (multiple layers of solid elements with top and bottom layers of shell elements), thick shell elements (TSHELL, 2D trimming only) and isogeometric shell elements (*ELEMENT_SHELL-NURBS_PATCH, 3D trimming only). Note it is not applicable for axisymmetric solids or 2D plane strain/stress elements. For details, see *DEFINE_CURVE_TRIM.

NOTE: Before revision 87566 this card was called ELEMENT_TRIM.

Card 1	1	2	3	4	5	6	7	8
Variable	PSID		ITYP					
Type	I		I					
Default	none		0					

VARIABLE	DESCRIPTION
PSID	Part set ID for trimming, see *SET_PART.
ITYP	Activation flag for sandwiched parts (laminates) trimming: EQ.0: Trimming for solid elements. EQ.1: Trimming for laminates.

Remarks:

This keyword is used together with *DEFINE_CURVE_TRIM to trim the parts defined in PSID at time zero, that is, before any stamping process simulation begins. Elements in the part set will be automatically trimmed in the defined direction if they intersect the trim curves. See examples in keyword section *DEFINE_CURVE_TRIM.

Note for solid element trimming, the keyword *INCLUDE_TRIM (not *INCLUDE) *must* be used to include the dynain blank to be trimmed.

Revision information:

1. Revision 87566: *ELEMENT_TRIM was changed to the current name *CONTROL_FORMING_TRIMMING.
2. Revision 95745: *CONTROL_FORMING_TRIMMING was changed to *CONTROL_FORMING_TRIMMING.
3. Revision 92088: 2D trimming of solid elements is implemented.
4. Revision 92289: 2D and 3D trimming of laminates (ITYP) is added.
5. Revision 93467: 3D trimming of solid elements is added.
6. Latter Revisions may incorporate more improvements and are suggested to be used for trimming.

*CONTROL

*CONTROL_FORMING_UNFLANGING

*CONTROL_FORMING_UNFLANGING_{OPTION}

Available options include:

<BLANK>

OUTPUT

Purpose: This keyword unfolds flanges of a deformable blank onto a rigid tooling mesh using an implicit statics solver. This is typically used in trim line unfolding during a stamping die development process. The option OUTPUT must be used together with *CONTROL_FORMING_UNFLANGING to get the modified trim curves. Other keywords related to blank size development are *CONTROL_FORMING_ONESTEP and *INTERFACE_BLANKSIZE_DEVELOPMENT. The feature is available in double precision for SMP.

Card Summary:

Card 1a.1. This card is included if no keyword option is used.

NOPTION	DVID	NUNBEND	STFBEND	STFCNT	IFLIMIT	DIST	ILINEAR
---------	------	---------	---------	--------	---------	------	---------

Card 1a.2. This card is included if no keyword option is used.

NB1	NB2	NB3	CHARLEN	NDOUTER			
-----	-----	-----	---------	---------	--	--	--

Card 1b. This card is included if the keyword option OUTPUT is used.

THMX	THMN	EPSMX					
------	------	-------	--	--	--	--	--

Data Cards:

Card 1a.1	1	2	3	4	5	6	7	8
Variable	NOPTION	DVID	NUNBEND	STFBEND	STFCNT	IFLIMIT	DIST	ILINEAR
Type	I	I	I	F	F	I	F	I
Default	none	N/A	none	none	none	none	↓	2

VARIABLE	DESCRIPTION
NOPTION	Flag to turn on an unfolding simulation: EQ.1: Activate the unfolding simulation program.
DVID	This variable is currently not being used.
NUNBEND	Estimated number of unbending, ranging from 10 to 100.
STFBEND	Unflanging stiffness, ranging from 0.1 to 10.0.
STFCNT	Normal stiffness, ranging from 0.1 to 10.0.
IFLIMIT	Iteration limit for the first phase of unfolding, typically ranging from 11 to 400.
DIST	Distance tolerance for auto-SPC along flange root. DIST (Figure 12-63) is usually slightly more than $\frac{1}{2}$ of the flange thickness. <i>This field must be left blank for ILINEAR = 2.</i> Also, nodes along the root can be directly positioned on the rigid body surface (addendum), leaving a DIST of zero (Figure 12-63).
ILINEAR	Unfolding algorithm selection flag: EQ.0: Nonlinear unfolding. EQ.1: Linear unfolding. EQ.2: A hybrid unfolding method (Revision 87100 and later). The curved 3D meshes of the flange will first be mapped onto the tooling surface to be used as a starting porting for nonlinear iterations; unfolding completes when force balance is reached. (recommended).

Card 1a.2	1	2	3	4	5	6	7	8
Variable	NB1	NB2	NB3	CHARLEN	NDOUTER			
Type	I	I	I	F	I			
Default	none	↓	↓	150.0	none			

*CONTROL

*CONTROL_FORMING_UNFLANGING

VARIABLE	DESCRIPTION
NB1	The start node ID (Figure 12-64) on a flange root boundary (fixed end of the flange, see Figures 12-63 and 12-64). For closed-loop flange root boundary, only this parameter needs to be defined; for open-loop flange root boundary, define this parameter as well as NB2 and NB3. The solver will automatically identify and automatically impose the necessary boundary constraints on all the nodes along the entire three-dimensional flange root boundary.
NB2	The ID of a node in the middle of the flange root boundary, see Figure 12-64 . Define this parameter for open-loop flange root boundary only.
NB3	The end node ID on a flange root boundary. Define this parameter for open-loop flange root boundary only. The “path” formed by NB1, NB2 and NB3 can be in any direction, meaning NB1 and NB3 (Figure 12-64) can be interchangeable.
CHARLEN	Maximum flange height (Figure 12-64) to limit the search region for the boundary nodes along the flange root. This value should be set bigger than the longest width (height) of the flange; and it is needed in some cases. This parameter is now automatically calculated as of Revision 92860.
NDOUTER	A node ID on the outer flange (free end of the flange) boundary. This node helps search of nodes along the flange root, especially when holes are present in the flange area, see Figure 12-64 .

Card 1b	1	2	3	4	5	6	7	8
Variable	THMX	THMN	EPSMX					
Type	I	I	I					
Default	10^{20}	0.0	10^{20}					

VARIABLE	DESCRIPTION
THMX	Maximum thickness beyond which elements are deleted; this is useful in removing wrinkling areas of the flange (shrink flange). Modified, unfolded flange outlines based on this parameter are stored in a file called trimcurve_upd.k, written using the *DEFINE_CURVE_TRIM_3D keyword. The modified flanges (before unfolding) are in a keyword file called mdfiedflangedpart.k, and the unmodified flange (unfolded) is in trimcurve_nmd.k, also written using keyword *DEFINE_CURVE_TRIM_3D. See Figure 12-64 for an explanation. Currently the modified flange and curves are not smooth, which will be improved in the future. To convert between *DEFINE_CURVE_TRIM_3D and IGES format refer to Figures in *INTERFACE_BLANKSIZE.
THMN	Minimum thickness below which elements are deleted; this is useful in removing overly thinned areas of the flange (stretch flange). Updated flange information based on this parameter is stored in files listed above.
EPSMX	Maximum effective plastic strain beyond which elements are deleted; this is useful in removing flange areas with high effective plastic strains (stretch flange). Updated flange information based on this parameter is stored in files listed above.

Introduction:

Unfolding of flanges is one of the first steps in a stamping die development process. Immediately after tipping, binder and addendums are built for unfolding of flanges. According to process considerations (trim conditions, draw depth, and material utilization, etc.), the addendums are built either in parallel or perpendicular to the draw die axis, tangentially off the main surface off the breakline (see Appendix T), or any combinations of the three scenarios. Trim lines are developed by unfolding the flanges in finished (hemmed or flanged) position onto these addendums. Addendum length in some areas may have to be adjusted to accommodate the unfolded trim lines. Trim line development is very critical in hard tool development. Inaccurate trim lines lead to trim die rework.

Inputs and Outputs:

The inputs for the keyword are:

1. blank or flanges in the finished configuration, and,
2. the draw die surface in mesh.

Meshes for flanges should be of a quality similar to the blank mesh one would build for a forming simulation. In LS-PrePost 4.0, this kind of mesh can be created using *Mesh → Automesh → Size*. Element formulation 16 with NIP set to 5 is recommended for the blank. The output results, in terms of unfolding steps and final unfolded flanges, are stored in the d3plot files. The unfolded flange/trim curves can be created from the unfolded flanges using *Curve → Spline → From Mesh → By part*. Since the program uses an implicit statics solver, the double precision version of LS-DYNA must be used.

Other Modeling Guidelines:

1. All addendum and flanges need to be oriented as if they are in a draw position, with the drawing axis parallel to the global Z-axis; specifically the flanges need to be on top of the addendum, as noted in [Figures 12-62, 12-63](#) and [12-64](#).
2. Normals of the to-be-unfolded flange side and tool surface side must be consistent and must face toward each other when the flange is unfolded, see [Figure 12-64](#).
3. Holes in the blank are allowed only for ILINEAR = 2.
4. Adaptive re-meshing is not supported.
5. To-be-unfolded flange and tool meshes must not share the same nodes. This can be easily done using the mesh detaching feature under *EleTol → DetEle* in LS-PrePost.
6. Meshes of the flange part and rigid tool can slightly overlap each other, but *large amounts of overlap* (area of flange already on addendum surface) is *not allowed*. In LS-PrePost the *EleTol → PtTrim* feature can trim off the overlapped flange portion. The curves used for the trimming can be obtained from the flange tangent curves on the addendum (which has a more regulated mesh pattern) using LS-PrePost's *Curve → Spline → Method From Mesh → By Edge → Prop* feature with appropriate angle definition. Furthermore, any holes are not allowed in the overlapping area.
7. ***CONTACT_FORMING_ONE WAY_SURFACE_TO_SURFACE** should be used for the contact between the blank and tool. Negative tool offsets on the ***CONTACT...** keyword is *not supported*.
8. The rigid tool (total fixed in ***MAT_020**) must be larger than the unfolded flanges, especially along symmetric lines. This may be obvious, nevertheless it is sometimes overlooked.
9. Nodes along the flange root are automatically fixed by defining NB1, NB2 and NB3, as shown in [Figure 12-64](#).

10. No “zigzag” along the flange root boundary, meaning that the boundary along the flange root must be smooth. This restriction is removed as of R10.0.
11. Symmetric boundary conditions are supported.
12. Thickness and effective plastic strain are stored in a file unflanginfo.out, which can be plotted in LS-PrePost 4.0; see [Figure 12-64](#).

Application Examples:

A partial input deck is provided below for flange unfolding of a fender outer, modified from the original NCAC Taurus model. Shown in [Figure 12-62](#) are the progressions of the unfolding process, where the finished flanges are to be unfolded onto the addendum (rigid body). A section view of the same unfolding before and after is found in [Figure 12-63](#). ILINEAR is set at 2 while DIST is left blank. Total numbers of elements are 1251 on the blank and 6600 on the tooling. It took less than 3 minutes on an 8 CPU (SMP) machine. Note that additional keywords, such as *CONTROL_IMPLICIT_FORMING, etc. are used. Termination criterion is set using the variable DELTAU in *CONTROL_IMPLICIT_TERMINATION. Termination is reached when the relative displacement ratio criterion is met, as indicated in the messag file. Termination time of 10.0 (steps) is sufficient for most cases, but may need to be extended in some cases to satisfy the DELTAU in some cases.

```
$-----1-----2-----3-----4-----5-----6-----7-----8
*KEYWORD
*INCLUDE
toolblankmesh.k
*CONTROL_FORMING_UNFLANGING
$ NOPTION      DVID      NUNBEND      STFBEND      STFCNT      IFLIMIT      DIST      ILINEAR
      1           100          0.2         15.0        400             2
$   NB1       NB2       NB3     CHARLEN      NDOUTER
      321       451       322        60.0       6245
*CONTROL_IMPLICIT_FORMING
1
*CONTROL_IMPLICIT_TERMINATION
$   DELTAU
$   set between 0.0005~0.001
      0.0005
*CONTROL_IMPLICIT_GENERAL
$   IMFLAG      DT0
      1       .1000
*CONTROL_IMPLICIT SOLUTION
$   NSLOLVR      ILIMIT      MAXREF      DCTOL      ECTOL      RCTOL      LSTOL
      2           2        1100      0.100      1.e20      1.e20
$   dnorm      divflag      inistif
      0           2           0           1
*PARAMETER
R   ENDTIME      10.0
I   elform       16
I   nip          5
R   bthick       1.0
*PARAMETER_EXPRESSION
R   D3PLOTS      ENDTIME/60.0
*CONTROL_TERMINATION
&ENDTIME
*DATABSE_BINARY_D3PLOT
```

*CONTROL

*CONTROL_FORMING_UNFLANGING

```
&D3PLOTS
*CONTROL_RIGID...
*CONTROL_HOURGLASS...
*CONTROL_BULK_VISCOSITY...
*CONTROL_SHELL...
*CONTROL_CONTACT
$ SLSFAC      RWPNAL     ISLCHK    SHLTHK    PENOPT    THKCHG    ORIEN
   0.01        0.0          2           1           4           0           4
$ USRSTR      USRFAC      NSBCS     INTERM    XPENE     SSTHK     ECDT     TIEDPRJ
   0           0           10          0           2.0          0
*CONTROL_ENERGY...
*CONTROL_ACCURACY...
*DATABASE_EXTENT_BINARY...
*SECTION_SHELL_TITLE
BLANK/FLANGE thickness and elform/nip specs.
&blksec    &elform     0.833      &nip       1.0
&bthick,&bthick,&bthick,&bthick
*PART...
*MAT_TRANSVERSELY_ANISOTROPIC_ELASTIC_PLASTIC...
*MAT_RIGID...
*CONTACT_FORMING_ONE WAY_SURFACE_TO_SURFACE
$ SURFA      SURFB      SURFATYP   SURFBTYP   SABOXID   SBBOXID   SAPR     SBPR
  &blkpid   &diepid    3           3
$ FS         FD          DC          VC          VDC        PENCHK    BT        DT
  0.125      0.0         0.0         0.0         20.0       0          0.0      1.000E+20
$ SFSA       SFSB       SAST        SBST        SFSAT     SFSBT     FSF       VSF
  1.0         1.0         0.0
$ SOFT       SOFSCL     LCIDAB     MAXPAR     PENTOL    DEPTH     BSORT     FRCFRQ
  0
$ PENMAX     THKOPT     SHLTHK    SNLOG      ISYM      I2D3D     SLDTHK    SLDSTF
$ IGAP       IGNORE     DPRFAC    DTSTIF
  2
*END
```

In [Figure 12-64](#) (top), with THMN set at 0.4 mm, the stretch flange area of the corner, which has thickness less than 0.4 mm, is removed; and the modified flange outlines are created accordingly (bottom). The partial input used is listed below.

```
*CONTROL_FORMING_UNFLANGING
$ NOPTION      DVID      NUNBEND    STFBEND    STFCNT    IFLIMIT    DIST     ILINEAR
   1           100        0.2         15.0       400          2
$ NB1         NB2         NB3        CHARLEN   NDOUTER
   321        451        322        60.0       6245
*CONTROL_FORMING_UNFLANGING_OUTPUT
$ THMX        THMN        EPSMX
   0.4
```

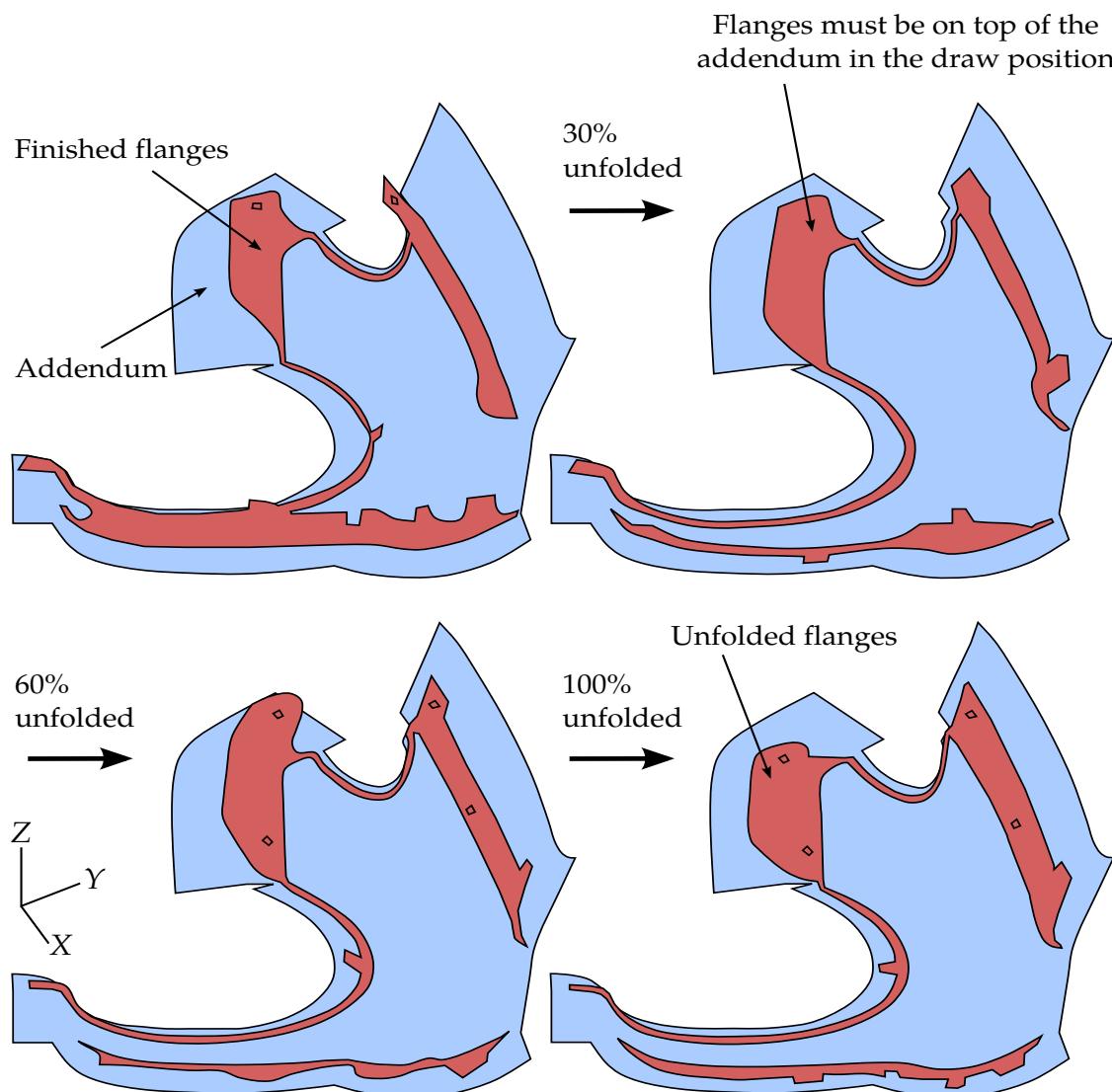


Figure 12-62. Flange unfolding progression of a fender outer (original model courtesy of NCAC at George Washington University).

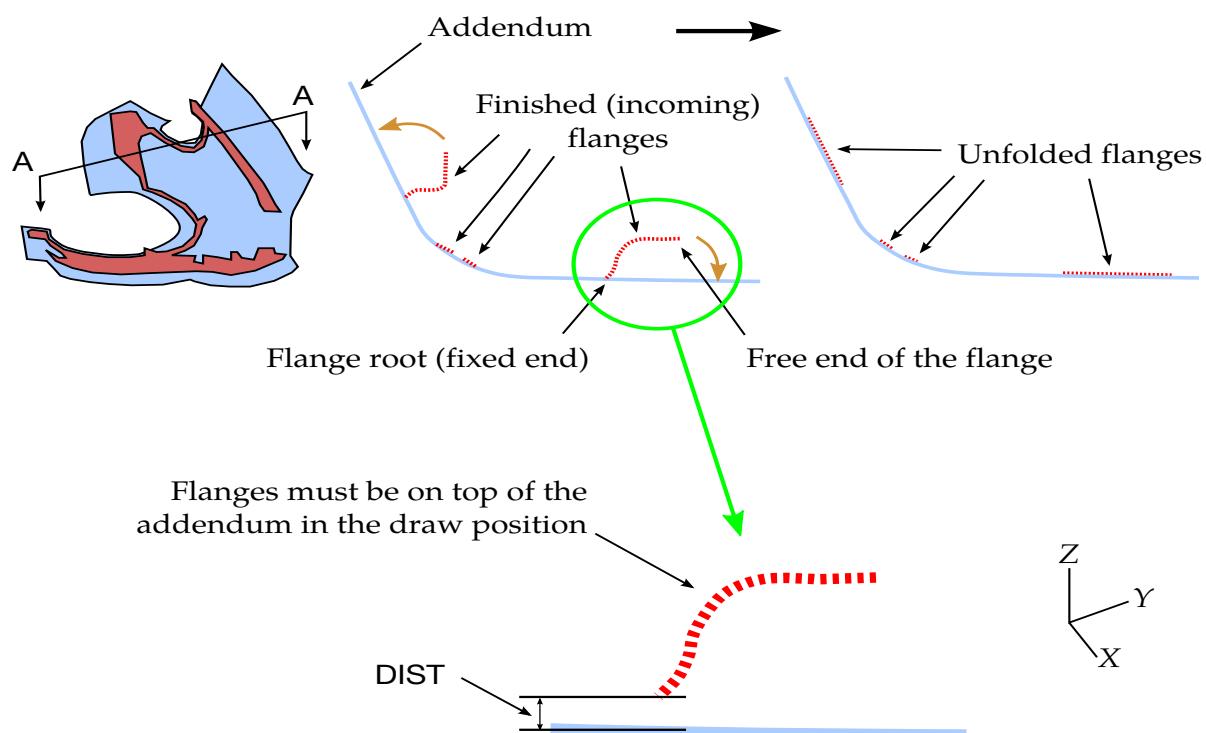
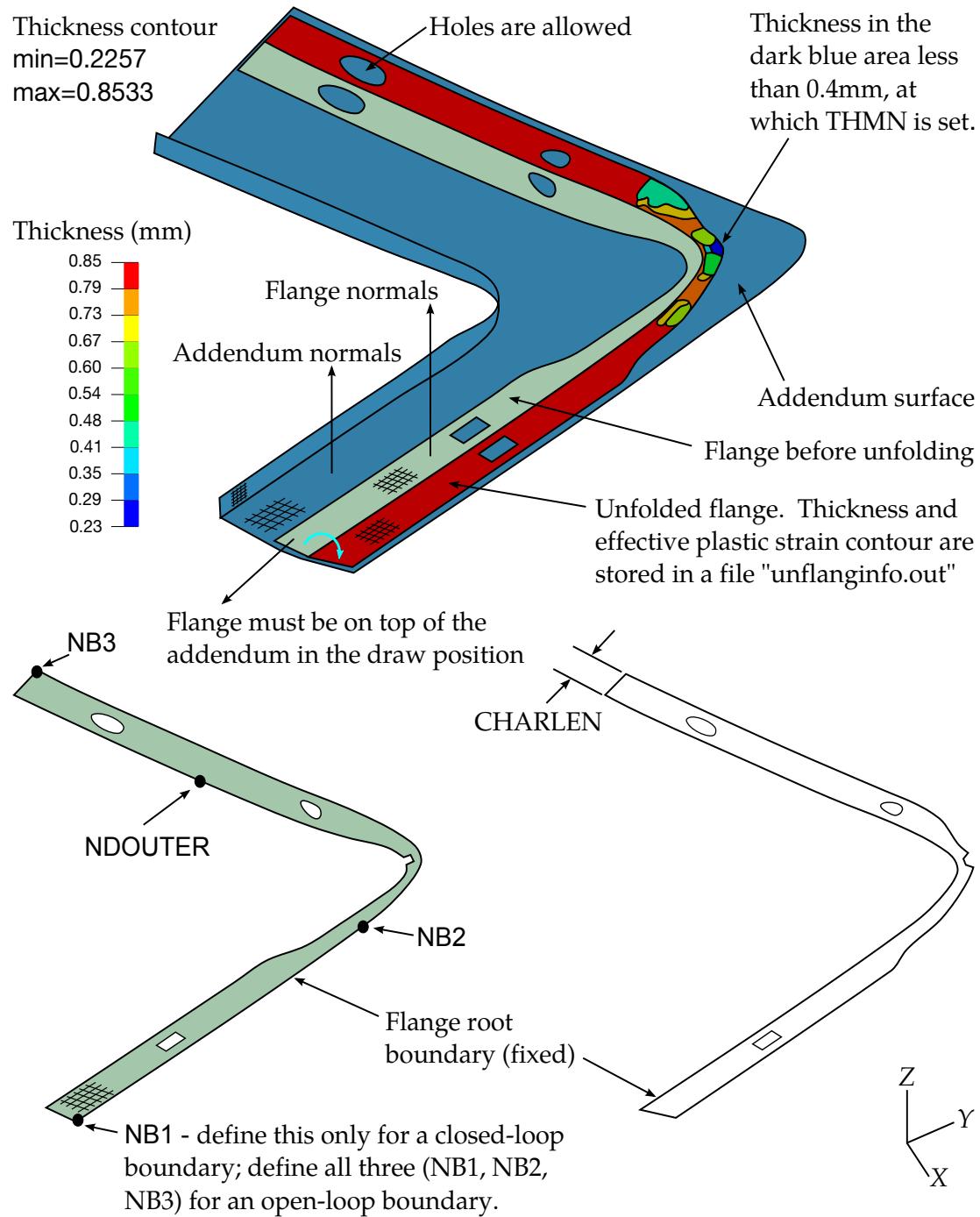


Figure 12-63. A section view showing flange unfolding before and after.



Original flange is modified based on THMN=0.4 and the mesh is stored in a file "mdfiedflangedpart.k". Boundary curves can be created using LSPP4.0 under Curve/Spline/From mesh/by part.

Modified boundary curves on unfolded flange are stored in a file "trimcurve_upd.k"; original boundary curves (without the corner cutout) is in "trimcurve_nmd.k".

Figure 12-64. Unfolding details and output files

*CONTROL

*CONTROL_FORMING_USER

*CONTROL_FORMING_USER

Purpose: This keyword, along with *CONTROL_FORMING_POSITION, or *CONTROL_FORMING_TRAVEL, allow user to set up a stamping process simulation. From this card various model parameters may be specified:

- material properties,
- material model,
- tooling kinematics,
- mesh adaptivity
- D3PLOT generation

NOTE: This option has been deprecated in favor of *CONTROL_FORMING_AUTOPOSITION_PARAMETER).

Card 1	1	2	3	4	5	6	7	8
Variable	BLANK	TYPE	THICK	R00	R45	R90	AL/FE	UNIT
Type	I	I	F	F	F	F	A	I
Default	none	0	none	1.0	R00	R00	F	1

Card 2	1	2	3	4	5	6	7	8
Variable	LCSS	K	N	E	DENSITY	PR	FS	MTYPE
Type	I	F	F	F	F	F	F	I
Default	none	none	none	none	none	none	0.1	37

Card 3	1	2	3	4	5	6	7	8
Variable	PATERN	VMAX	AMAX	LVLADA	SIZEADA	ADATIMS	D3PLT	GAP
Type	I	F	F	I	F	I	I	F
Default	1	1000.0	500000.	0	0	0	10	1.1t

VARIABLE	DESCRIPTION
BLANK	PID of a sheet blank, as in *PART.
TYPE	Flag of part or part set ID for the blank: EQ.0: Part ID. EQ.1: Part set ID.
THICK	Thickness of the blank. This variable is ignored if the thickness is already defined in *SECTION_SHELL.
R00, R45, R90	Material anisotropic parameters. For transverse anisotropy the R value is set to the average value of R00, R45, and R90.
AL/FE	This parameter is used to define the Young's Modulus, E, and density, ρ , for the sheet blank. If this variable is defined, E and ρ will be found by using the proper unit, as listed in Table 8.1, under *CONTROL_FORMING_TEMPLATE. EQ.A: the blank is aluminum EQ.F: the blank is steel (default)
UNIT	Units adopted in this simulation. Define a number between 1 and 10. Table 8.1 is used to determine the value for UNIT. This unit is used to obtain proper values for punch velocity, acceleration, time step, and physical and material properties.
LCSS	If the material for the blank has not been defined, this curve will be used to define the stress-strain relation. Otherwise, this variable is ignored.
PREBD	"Pull-over" distance for the upper and lower binders after closing in a 4-piece stretch draw, as shown in Figure 12-60 .

*CONTROL

*CONTROL_FORMING_USER

VARIABLE	DESCRIPTION
K	Strength coefficient for exponential hardening ($\bar{\sigma} = k\bar{\varepsilon}^n$). If LCSS is defined, or if a blank material is defined with *MAT_036 or *MAT_037, this variable is ignored.
N	Exponent for exponential hardening ($\bar{\sigma} = k\bar{\varepsilon}^n$). If LCSS is defined, or if a blank material is defined with *MAT_036 or *MAT_037, this variable is ignored.
E	Young's Modulus. If AL/FE is user defined, E is unnecessary.
DENSITY	Material density of the blank. If AL/FE is defined, this variable is unnecessary.
PR	Poisson's ratio. If AL/FE is user defined, this variable is unnecessary.
FS	Coulomb friction coefficient. If contact is defined with *CONTACT_FORMING_..., this variable is ignored.
MTYP	Material model identification number, for example, 36 for *MAT_036 and 37 for *MAT_037. Currently only material models 36 and 37 are supported.
PATERN	Velocity profile of the moving tool. If the velocity and the profile are defined by *BOUNDARY_PRESCRIBED_MOTION_RIGID, and *DEFINE_CURVE, this variable is ignored. EQ.1: Ramped velocity profile. EQ.2: Smooth velocity curve.
VMAX	The maximum allowable tool travel velocity.
AMAX	The maximum allowable tool acceleration.
LVLADA	Maximum mesh adaptive level.
SIZEADA	Minimum element size permitted during mesh adaptivity.
ADATIMS	Total number of adaptive steps during the simulation.
D3PLT	The total number of output states in the D3PLOT database.
GAP	Minimum gap between two closing tools at home position, in the travel direction of the moving tool. This variable will be used for *CONTROL_FORMING_POSITION.

Keyword examples:

A partial keyword example provided below is for tools in their home positions in a simple 2-piece crash forming die. A steel sheet blank PID 1, is assigned with a thickness of 0.76mm (UNIT = 1) and *MAT_037 with anisotropic values indicated, to follow hardening curve of 90903, form in a 'ramped' type of velocity profile with maximum velocity of 5000mm/s and acceleration of 500000.0 mm/s², adapt mesh 5 levels with smallest adapted element size of 0.9 for a total of 20 adaptive steps, create a total of 15 post-processing states, and to finish forming with a final gap of 1.1mm between the tools (PID3 and 5) at home position. The upper tool with PID 3 is to be moved back in Z axis to clear the interference with the blank before close toward the lower tool of target PID 5.

```
*CONTROL_FORMING_USER
$ BLANK      TYPE     THICK      R00      R45      R90      AL/FE      UNIT
    1          0        0.76     1.5       1.6      1.4        F          1
$ LCSS       K         N         E      DENSITY      PR      FS      MTYPE
90903
$ PATTERN    VMAX     AMAX     LVLADA    SIZEADA    ADATIMS    D3PLT      GAP
    1      5000.0   500000.0      5        0.9     20.0      15.0      1.1
$-----+-----+-----+-----+-----+-----+-----+-----+-----+
*CONTROL_FORMING_POSITION
$ This is for tools in home position.
$ PID      PREMOVE     TARGET
    3            5
```

The following partial keyword example is for tools already positioned in relationship to the blank and ready to close. All assigned properties for the blank remain the same. Here the upper tool PID3 is not going to be moved back, but instead it will move forward to close with the lower tool of target PID 5 in the direction specified by the vector ID 999.

```
*CONTROL_FORMING_USER
    1          0        1.0       1.5      1.6      1.4        F          1
90903
    1      5000.0   500000.0      5        0.9     20.0      15.0      1.1
*CONTROL_FORMING_TRAVEL
$ PID      VID      TRAVEL     TARGET      GAP      PHASE      FOLLOW
    3      999            5        1.1        1
```

Revision information:

This keyword is available starting in LS-DYNA Revision 48319.

*CONTROL

*CONTROL_FREQUENCY_DOMAIN

*CONTROL_FREQUENCY_DOMAIN

Purpose: Set global control flags and parameters for frequency domain analysis.

Card 1	1	2	3	4	5	6	7	8
Variable	REFGEO	MPN	MCF					
Type	I	F	I					
Default	0	0.0	0					

VARIABLE	DESCRIPTION
REFGEO	Flag for reference geometry in acoustic eigenvalue analysis: EQ.0: use original geometry ($t = 0$), EQ.1: use deformed geometry at the end of transient analysis.
MPN	Large mass added per node to be used in large mass method for enforced motion.
MCF	Flag for writing out MCF (Modal Coefficient File) from SSD analysis, to be used in fatigue analysis with nCode Designlife: EQ.0: Don't write out MCF. EQ.1: Write out MCF.

Remarks:

- 1. Deformed Geometry Eigenvalue Analysis.** For acoustic eigenvalue analysis (see keyword *FREQUENCY_DOMAIN_ACOUSTIC_FEM_EIGENVALUE), sometimes it is desired to extract the eigenvalues at the end of transient analysis, based on the deformed geometry. This is useful to study the effect of loading history on acoustic eigenvalues. In this case, one can set REFGEO = 1 to use the deformed geometry at the end of transient analysis.
- 2. Large Mass Method.** For enforced motion excitation (e.g. nodal acceleration, velocity, or displacement) in FRF, SSD, or random vibration analysis, one can use the large mass method to compute the response. With the large mass method, the user attaches a large mass to the nodes under excitation. LS-DYNA converts the enforced motion excitation to nodal force on the same nodes in the same direction to produce the desired enforced motion. MPN is the large mass

attached to each node under excitation (usually it is in the range of 10^5 - 10^7 times of the original mass of the entire structure). The large mass still needs to be applied to the nodes using the keyword *ELEMENT_MASS_{OPTION}.

The large nodal force p is computed as shown in the following table:

Nodal Motion	Nodal Force
Acceleration	$p = m_L \ddot{u}$
Velocity	$p = i\omega m_L \dot{u}$
Displacement	$p = -\omega^2 m_L u$

where ω is the round frequency, m_L is the large mass attached to each node (MPN), and \ddot{u} , \dot{u} and u are the enforced acceleration, velocity and displacement, respectively.

*CONTROL

*CONTROL_HOURGLASS

*CONTROL_HOURGLASS_{OPTION}

Available options include:

<BLANK>

936

The “936” option switches the hourglass formulation for shells so that it is identical to that used in LS-DYNA version 936. The modification in the hourglass control from version 936 was to ensure that all components of the hourglass force vector are orthogonal to rigid body rotations. However, problems that run under version 936 sometimes lead to different results in versions 940 and later. This difference in results is primarily due to the modifications in the hourglass force vector. Versions released after 936 should be more accurate.

Purpose: Redefine the default values of hourglass control type and coefficient.

Card 1	1	2	3	4	5	6	7	8
Variable	IHQ	QH						
Type	I	F						
Default	Rem 1	0.1						
Remarks	1,2	3,4						

VARIABLE

DESCRIPTION

IHQ

Default hourglass control type:

EQ.0: See [Remark 1](#),

EQ.1: Standard viscous form (may inhibit body rotation if solid element shapes are skewed),

EQ.2: Viscous form, Flanagan-Belytschko integration for solid elements,

EQ.3: Viscous form, Flanagan-Belytschko with exact volume integration for solid elements,

EQ.4: Stiffness form of type 2 (Flanagan-Belytschko),

VARIABLE	DESCRIPTION
	EQ.5: Stiffness form of type 3 (Flanagan-Belytschko) for solid elements,
	EQ.6: Belytschko-Bindeman [1993] assumed strain co-rotational stiffness form for 2D and 3D solid elements,
	EQ.7: Linear total strain form of type 6 hourglass control.
	EQ.8: Activates full projection warping stiffness for shell formulations 9, 16 and -16. A speed penalty of 25% is common for this option.
	EQ.9: Puso [2000] enhanced assumed strain stiffness form for 3D hexahedral elements,
	EQ.10: Cosserat Point Element (CPE) developed by Jabareen and Rubin [2008] for 3D hexahedral elements and Jabareen et.al [2013] for 10-noded tetrahedral elements. See Remark 6 .
QH	Default hourglass coefficient.

Remarks:

1. **Hourglass control types.** Hourglass control is viscosity or stiffness that is added to quadrilateral shell elements and hexahedral solid elements that use reduced integration. It also applies to formulation 1 tshells. Without hourglass control, these elements would have zero energy deformation modes which could grow large and destroy the solution. *CONTROL_HOURGLASS can be used to redefine the default values of the hourglass control type and coefficient. If omitted or if IHQ = 0, the default hourglass control types are as follows:
 - a) For shells: viscous type for explicit; stiffness type for implicit.
 - b) For solids: type 2 for explicit; type 6 for implicit.
 - c) For tshell formulation 1: type 2.

These default values are used unless HGID on *PART is used to point to *HOURGLASS data which overrides the default values for that part.

For explicit analysis, shell elements can be used with viscous hourglass control, (IHQ = 1 = 2 = 3) or stiffness hourglass control (IHQ = 4 = 5). Only shell forms 9, 16 and -16 use the warping stiffness invoked by IHQ = 8. For implicit analysis, the viscous form is unavailable.

For explicit analysis, hexahedral elements can be used with any of the hourglass control types except IHQ = 8. For implicit analysis, only IHQ = 6, 7, 9, and 10 are available.

If IHQ is set to a value that is invalid for some elements in a model, then the hourglass control type for those elements is automatically reset to a valid value. For explicit analysis, if IHQ = 6, 7, 9, or 10, then shell elements will be switched to type 4 except for forms 16 and -16 shells that are switched to type 8. If IHQ = 8, then solid elements and shell elements that are not form 16 will be switched to type 4. For implicit analysis, if IHQ = 1-5, then solid elements will be switched to type 6, and if IHQ = 1, 2, 3, 6, 7, 9, or 10, then shell elements will be switched to type 4.

2. **Viscous hourglass control.** Viscous hourglass control has been used successfully with shell elements when the response with stiffness based hourglass control was overly stiff. As models have grown more detailed and are better able to capture deformation modes, there is less need for viscous forms. To maintain back compatibility, viscous hourglass control remains the default for explicit analysis, but there may be better choices, particularly the newer forms for bricks (6, 7, 9, and 10).
3. **Hourglass coefficient stability.** QH is a coefficient that scales the hourglass viscosity or stiffness. With IHQ = 1 through 5 and IHQ = 8, values of QH that exceed 0.15 may cause instabilities. Hourglass types 6, 7, 9, and 10 will remain stable with larger QH and can work well with QH = 1.0 for many materials. However, for plasticity models, a smaller value such as QH = 0.1 may work better since the hourglass stiffness is based on elastic properties.
4. **Hourglass control for hexahedral elements.** Hourglass types 6, 7, 9, and 10 for hexahedral elements are based on physical stabilization using an enhanced assumed strain method. When element meshes are not particularly skewed or distorted, their behavior may be very similar and all can produce accurate coarse mesh bending results for elastic material with QH = 1.0. However, form 9 gives more accurate results for distorted or skewed elements. In addition, for materials 3, 18 and 24 there is the option to use a negative value of QH. With this option, the hourglass stiffness is based on the current material properties, i.e., the plastic tangent modulus, and scaled by $|QH|$.
5. **IHQ type 7.** Hourglass type 7 is a variation on form 6. Instead of updating the hourglass forces incrementally using the current stiffness and an increment of deformations, the total hourglass deformation is evaluated each cycle. This ensures that elements always spring back to their initial geometry if the load is removed and the material has not undergone inelastic deformation. Hourglass type 7 is recommended for foams that employ *INITIAL_FOAM_REFER-

ENCE_GEOMETRY. However, the CPU time for type 7 is roughly double that for type 6, so it is only recommended when needed.

6. **IHQ type 10.** Hourglass type 10 for 1-point solid elements or 10-noded tetrahedron of type 16 are structural elements based on Cosserat point theory that allows for accurate representation of elementary deformation modes (stretching, bending and torsion) for general element shapes and hyperelastic materials. To this end, the theory in Jabareen and Rubin [2008] and Jabareen et.al [2013] has been generalized in the implementation to account for any material response. The deformation is separated into a homogenous and an inhomogeneous part where the former is treated by the constitutive law and the latter by a hyperelastic formulation that is set up to match analytical results for the deformation modes mentioned above. Tests have shown that the element is giving more accurate results than other hexahedral elements for small deformation problems and more realistic behavior in general.

***CONTROL_IMPLICIT**

LS-DYNA's implicit mode may be activated in two ways. Using the *CONTROL_IMPLICIT_GENERAL keyword, a simulation may be flagged to run entirely in implicit mode. Alternatively, an explicit simulation may be seamlessly switched into implicit mode at the termination time using the *INTERFACE_SPRINGBACK_SEAMLESS keyword. The seamless switching feature is intended to simplify metal forming springback calculations, where the forming phase can be run in explicit mode, followed immediately by an implicit static springback simulation. In case of difficulty, restart capability is supported.

Several control cards are available to support implicit analysis. Default values are carefully selected to minimize input necessary for most simulations. The implicit control cards follow:

- *CONTROL_IMPLICIT_AUTO
- *CONTROL_IMPLICIT_BUCKLE
- *CONTROL_IMPLICIT_CONSISTENT_MASS
- *CONTROL_IMPLICIT_DYNAMICS
- *CONTROL_IMPLICIT_EIGENVALUE
- *CONTROL_IMPLICIT_FORMING
- *CONTROL_IMPLICIT_GENERAL
- *CONTROL_IMPLICIT_INTERA_RELIEF
- *CONTROL_IMPLICIT_JOINTS
- *CONTROL_IMPLICIT_MODAL_DYNAMIC
- *CONTROL_IMPLICIT_MODAL_DYNAMIC_DAMPING
- *CONTROL_IMPLICIT_MODAL_DYNAMIC_MODE
- *CONTROL_IMPLICIT_MODES
- *CONTROL_IMPLICIT_ORDERING
- *CONTROL_IMPLICIT_RESIDUAL_VECTOR
- *CONTROL_IMPLICIT_ROTATIONAL_DYNAMICS

*CONTROL_IMPLICIT_SOLUTION
*CONTROL_IMPLICIT_SOLVER
*CONTROL_IMPLICIT_SSD_DIRECT
*CONTROL_IMPLICIT_STABILIZATION
*CONTROL_IMPLICIT_STATIC_CONDENSATION
*CONTROL_IMPLICIT_TERMINATION

*CONTROL

*CONTROL_IMPLICIT_AUTO

*CONTROL_IMPLICIT_AUTO_{OPTION}

Available options for *OPTION* include:

<BLANK>

DYN

SPR

Purpose: Define parameters for automatic time step control during implicit analysis (see also *CONTROL_IMPLICIT_GENERAL). The DYN option allows setting controls specifically for the dynamic relaxation phase. The SPR option allows setting controls specifically for the springback phase.

Card 1	1	2	3	4	5	6	7	8
Variable	IAUTO	ITEOPT	ITEWIN	DTMIN	DTMAX	DTEXP	KFAIL	KCYCLE
Type	I	I	I	F	F	F		
Default	0	11	5	DT/1000.	DT×10.	none		

Mid-Step Residual Optional Card. Define this card if and only if IAUTO.EQ.3

Card 2	1	2	3	4	5	6	7	8
Variable	HCMIN	HCMAX	HMMIN	HMMAX	HNTMAX	HNRMAX	HRTMAX	HRRMAX
Type	F	F	F	F	F	F	F	F
Default	2× RCTOL	10× RCTOL	2× RMTOL	10× RMTOL	Infinity	Infinity	Infinity	Infinity

VARIABLE

DESCRIPTION

IAUTO

Automatic time step control flag

EQ.0: Constant time step size

EQ.1: Automatically adjust time step size

EQ.2: Automatically adjust time step size and synchronize with

VARIABLE	DESCRIPTION
	thermal mechanical time step.
	EQ.3: Same as 1, but accounting for mid step residual values with respect to parameters on card 2 and according to the Remark for IAUTO.
	LT.0: Curve ID = (-IAUTO) gives time step size as a function of time. If specified, DTMIN and DTMAX will still be applied.
ITEOPT	Optimum equilibrium iteration count per time step. See Figure 12-65 .
ITEWIN	Allowable iteration window. If iteration count is within ITEWIN iterations of ITEOPT, step size will not be adjusted for the next step.
DTMIN	Minimum allowable time step size. Simulation stops with error termination if time step falls below DTMIN. LT.0: Enable automatic key point generation. Minimum allowable time step is DTMIN .
DTMAX	Maximum allowable time step size. LT.0: Curve ID = (-DTMAX) gives max step size as a function of time. Also, the step size is adjusted automatically so that the time value of each point in the curve is reached exactly (see Figures 12-66 and 0-3).
DTEXP	Time interval to run in explicit mode before returning to implicit mode. Applies only when automatic implicit-explicit switching is active (IMFLAG = 4 or 5 on *CONTROL_IMPLICIT_GENERAL). Also, see KCYCLE. EQ.0: Defaults to the current implicit time step size. LT.0: Curve ID = (-DTEXP) gives the time interval as a function of time.
KFAIL	Number of failed attempts to converge implicitly for the current time step before automatically switching to explicit time integration. Applies only when automatic implicit-explicit switching is active. The default is one attempt. If IAUTO = 0, any input value is reset to unity.
KCYCLE	Number of explicit cycles to run in explicit mode before returning to the implicit mode. The actual time interval that is used will be

*CONTROL

*CONTROL_IMPLICIT_AUTO

VARIABLE	DESCRIPTION
	the maximum between DTEXP and KCYCLE*(latest estimate of the explicit time step size).
HCMIN, HCMAX	Mid-point relative Euclidian residual norm min and max tolerance, to be seen as a confidence interval, see Remark for IAUTO = 3. Only active if RCTOL on *CONTROL_IMPLICIT_SOLUTION is set.
HMMIN, HMMAX	Mid-point relative maximum residual norm min and max tolerance, to be seen as a confidence interval, see Remark for IAUTO = 3. Only active if RMTOL on *CONTROL_IMPLICIT_SOLUTION is set.
HNTMAX	Mid-point absolute Nodal Translational norm tolerance, see Remark for IAUTO = 3. Only active if NTTOL on *CONTROL_IMPLICIT_SOLUTION is set.
HNRMAX	Mid-point absolute Nodal Rotational norm tolerance, see Remark for IAUTO = 3. Only active if NRTOL on *CONTROL_IMPLICIT_SOLUTION is set.
HRTMAX	Mid-point absolute Rigid body Translational norm tolerance, see Remark for IAUTO = 3. Only active if RTTOL on *CONTROL_IMPLICIT_SOLUTION is set.
HRRMAX	Mid-point absolute Rigid body Rotational norm tolerance, see Remark for IAUTO = 3. Only active if RRTOL on *CONTROL_IMPLICIT_SOLUTION is set.

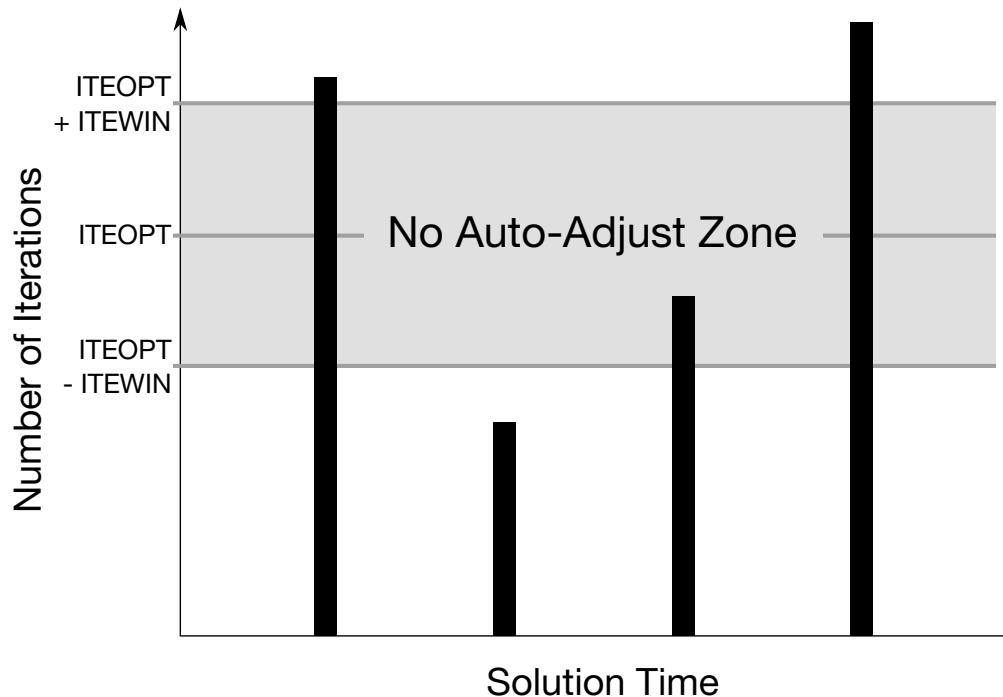


Figure 12-65. Iteration Window as defined by ITEOPT and ITEWIN.

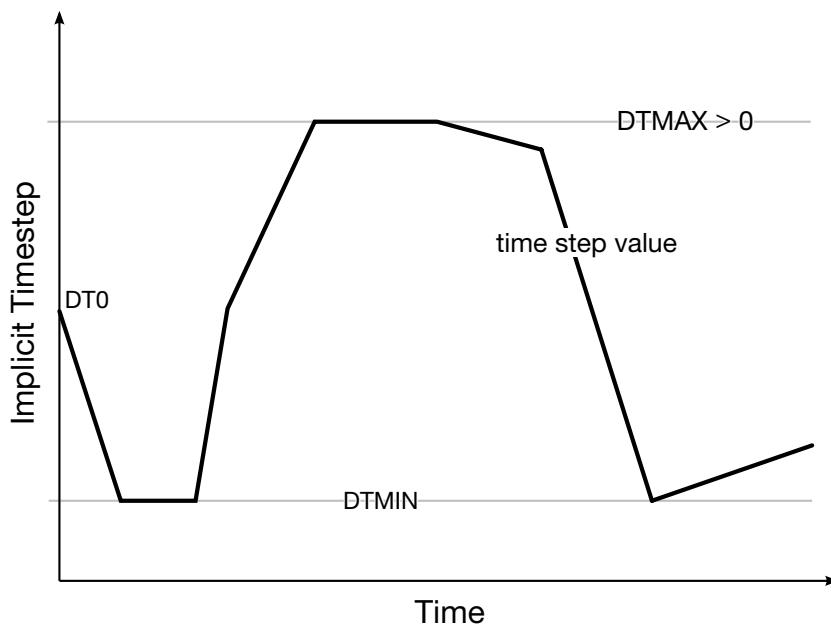


Figure 12-66. The implicit time step size changes continuously as a function of convergence within the bounds set by DT_{MIN} and DT_{MAX}

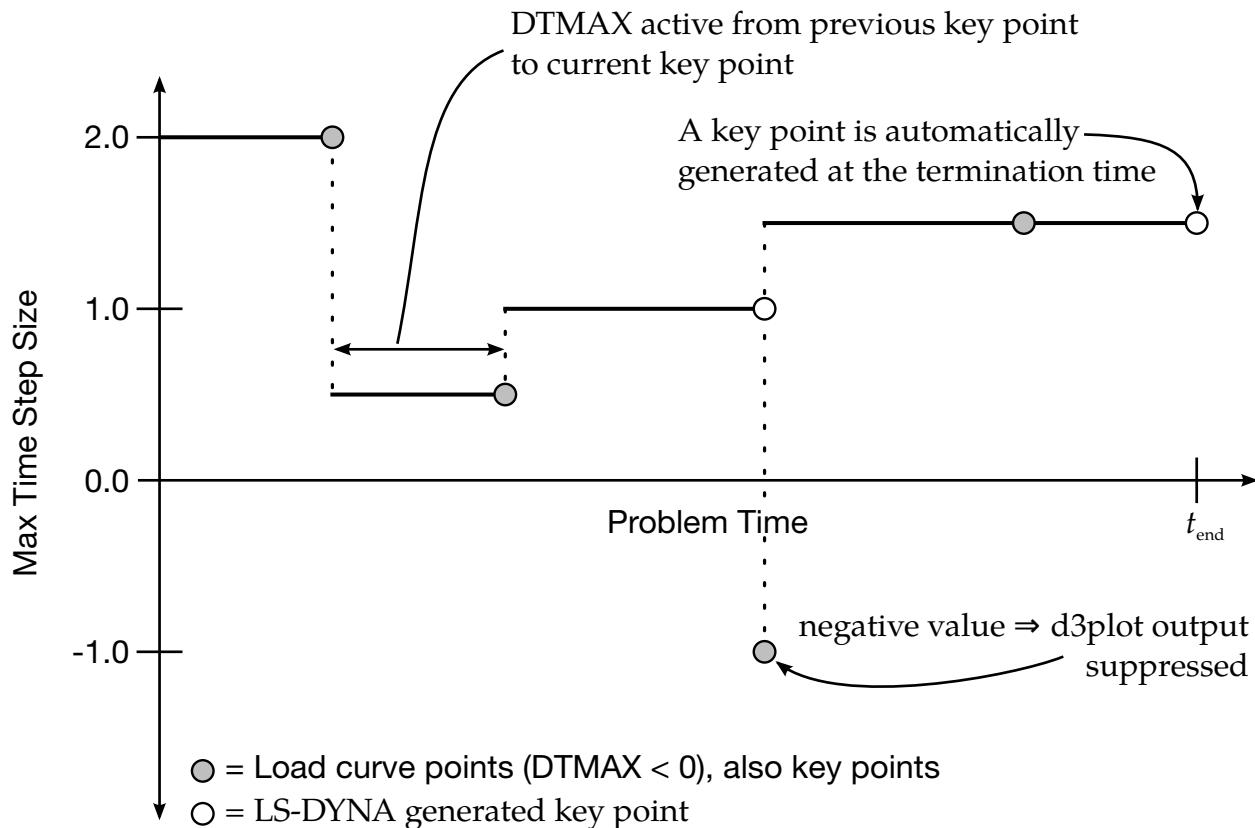


Figure 0-3. $DTMAX < 0$. The maximum time step is set by a load curve of $LCID = -DTMAX$ interpolated using piecewise constants. The abscissa values of the load curve determine the set of *key points*. The *absolute value* of the ordinate values set the maximum time step size. *Key points* are special time values for which the integrator will adjust the time step so as to reach *exactly*. For each key point with a positive function value, LS-DYNA will write the state to the binary database.

Remarks:

VARIABLE	REMARK
IAUTO	The default for IAUTO depends on the analysis type. For “spring-back” analysis, automatic time step control and artificial stabilization are activated by default.
	IAUTO = 3 accounts for the residual norm at the mid-point geometry between converged steps. We have included below a numbered set of conditions. These conditions are used to adjust the time step according to the following algorithm:

VARIABLE**REMARK**

1. The time step is *increased* if either of the following criteria is met simultaneously:
 - 1b, 2b, 3, 4a, or
 - 1a, 2a, 3, 4b.

2. If the time step is not increased by these conditions, then it is decreased if any of the following above criteria is *violated*
 - 1b or
 - 2b or
 - 3 or
 - 4b.

The Criteria Mentioned Above

1. The relative Euclidian residual norm at the mid-point is less than
 - a) HCMIN
 - b) HCMAX.

2. The relative maximum residual norm at the mid-point is less than
 - a) HMMIN
 - b) HMMAX.

3. At least one of the following conditions is satisfied at the midpoint
 - a) The max norm of nodal translation < HNTMAX
 - b) The max norm of nodal rotation < HNRMAX
 - c) The max norm of rigid body translation < HRTMAX
 - d) The max norm of rigid body rotation < HRRMAX

4. The number of iterations for convergence is
 - a) less than ITEOPT – ITEWIN
 - b) or equal to ITEOPT + ITEWIN.

Note

IAUTO = 1 is obtained as a special case for

$$\text{HCMIN} = \text{HMMIN} = 0$$

VARIABLE	REMARK
	and
	$\begin{aligned} \text{HCMAX} &= \text{HMMAX} = \text{HNTMAX} = \text{HNRMAX} \\ &= \text{HRTMAX} = \text{HRRMAX} = \infty \end{aligned}$
	The remaining parameters below works the same way as for IAU-T0 = 1.
ITEOPT	With IAUTO = 1 or 2, the time step size is adjusted if convergence is reached in a number of iterations that falls outside the specified "iteration window", increasing after "easy" steps, and decreasing after "difficult" but successful steps. ITEOPT defines the midpoint of the iteration window. A value of ITEOPT = 30 or more can be more efficient for highly nonlinear simulations by allowing more iterations in each step, hence fewer total steps.
DTMIN	Often specific simulation times that are of special interest, such as when a peak load value occurs, require the simulation to be solved at exactly that time. The automatic key point generation feature attempts to automatically find such times by analyzing the load curves for stationary points in time. The keywords whose load curves are analyzed along with what type of curve value are shown below.
	<ul style="list-style-type: none">• *INITIAL_AXIAL_FORCE_BEAM - global max• *INITIAL_STRESS_SECTION - global max• *LOAD_NODE_POINT - local min/max• *LOAD_NODE_SET - local min/max• *LOAD_SEGMENT - local min/max• *LOAD_SEGMENT_SET - local min/max• *LOAD_SHELL - local min/max• *LOAD_SHELL_SET - local min/max• *BOUNDARY_PRESCRIBED_MOTION_NODE - local min/max• *BOUNDARY_PRESCRIBED_MOTION_SET - local min/max
	For *CONTROL_IMPLICIT_DYNAMICS, the birth, death, and burial time are key points.
	For *CONTACT_AUTOMATIC_SURFACE_TO_SURFACE-MORTAR, MPAR1 is a key point when IGNORE = 3.
	The start time and end time of the simulation are also key points.

VARIABLE	REMARK
	To enable automatic key point generation, enter DTMIN negated.
DTMAX	To strike a particular simulation time exactly, create a key point curve (Figure 0-3) and enter DTMAX = -(curve ID). This is useful to guarantee that important simulation times, such as when peak load values occur, are reached exactly. A user defined key point curve can be used together with automatic key point generation. In that case all key points are merged into a compound list of key points.
DTEXP	When the automatic implicit-explicit switching option is activated (IMFLAG = 4 or 5 on *CONTROL_IMPLICIT_GENERAL), the solution method will begin as implicit, and if convergence of the equilibrium iterations fails, automatically switch to explicit for a time interval of DTEXP. A small value of DTEXP should be chosen so that significant dynamic effects do not develop during the explicit phase, since these can make recovery of nonlinear equilibrium difficult during the next implicit time step. A reasonable starting value of DTEXP may equal several hundred explicit time steps.

*CONTROL

*CONTROL_IMPLICIT_BUCKLE

*CONTROL_IMPLICIT_BUCKLE

Purpose: Activate implicit buckling analysis when termination time is reached (see also *CONTROL_IMPLICIT_GENERAL). Optionally, buckling analyses are performed at intermittent times.

Card 1	1	2	3	4	5	6	7	8
Variable	NMODE	BCKMTH						
Type	I	I						
Default	0	↓						

VARIABLE	DESCRIPTION
NMODE	Number of buckling modes to compute: EQ.0: none (default) GT.0: compute n lowest buckling modes LT.0: curve ID = (-NEIG) used for intermittent buckling analysis
BCKMTH	Method used to extract buckling modes: EQ.1: Use Block Shift and Invert Lanczos. Default of all problems not using *CONTROL_IMPLICIT_INERTIA_RELIEF. EQ.2: Use Power Method. Only valid option for problems using *CONTROL_IMPLICIT_INERTIA_RELIEF. Optional for other problems. See Remarks.

Remarks:

1. **Buckling Eigenproblem.** Buckling analysis is performed at the end of a static implicit simulation or at specified times during the simulation. The simulation may be linear or nonlinear but must be implicit. After loads have been applied to the model, the buckling eigenproblem is solved:

$$[\mathbf{K}_M + \lambda \mathbf{K}_G]\{u\} = 0$$

where \mathbf{K}_M is the material tangent stiffness matrix, and the geometric or initial stress stiffness matrix \mathbf{K}_G is a function of internal stress in the model. The lowest

n eigenvalues and eigenvectors are computed. The eigenvalues, written to text file **eigout**, represent multipliers to the applied loads which give buckling loads. The eigenvectors, written to binary database **d3eigv**, represent buckling mode shapes. These modes can be viewed and animated using LS-PrePost.

2. **NMODE.** When NMODE > 0, eigenvalues will be computed at the termination time and LS-DYNA will terminate.

When NMODE < 0, an intermittent buckling analysis will be performed. This is a transient simulation during which loads are applied, with buckling modes computed periodically during the simulation. Changes in geometry, stress, material, and contact conditions will affect the buckling modes. The transient simulation must be implicit. The curve ID = -NMODE indicates when to extract the buckling modes, and how many to extract. Define one curve point at each desired extraction time, with a function value equal to the number of buckling modes desired at that time. A d3plot database will be produced for the transient solution results. Consecutively numbered **d3eigv** and **eigout** databases will be produced for each intermittent extraction. The extraction time is indicated in each database's analysis title.

3. **Solver.** The buckling modes can be computed using either Block Shift and Invert Lanczos or the Power Method. It is strongly recommended that the Block Shift and Invert Lanczos method is used as it is a more powerful and robust algorithm. For problems using *CONTROL_IMPLICIT_INERTIA_RELIEF the Power Method must be used and any input value for BCKMTH will be overridden with the required value of 2. There may be some problems, which are not using *CONTROL_IMPLICIT_INERTIA_RELIEF, where the Power Method may be more efficient than Block Shift and Invert Lanczos. But the Power Method is not as robust and reliable as Lanczos and results should be verified. Furthermore convergence of the Power Method is better for buckling problems where the expected buckling mode is close to one in magnitude and the dominant mode is separated from the secondary modes. The number of modes extracted via the Power Method should be kept in the range of 1 to 5.
4. **Geometric Stiffness.** The geometric stiffness terms needed for buckling analysis will be automatically computed when the buckling analysis time is reached, regardless of the value of the geometric stiffness flag IGS on *CONTROL_IMPLICIT_GENERAL.
5. **Executable Precision.** A double precision executable should be used for best accuracy in buckling analysis.
6. **Applicable Parameters.** Parameters CENTER, LFLAG, LFTEND, RFLAG, RHTEND and SHFSCL from *CONTROL_IMPLICIT_EIGENVALUE are applicable to buckling analysis. For buckling analysis CENTER, LFTEND, RHTEND and SHFSCL are in units of the eigenvalue spectrum.

*CONTROL

*CONTROL_IMPLICIT_CONSISTENT_MASS

*CONTROL_IMPLICIT_CONSISTENT_MASS

Purpose: Use the consistent mass matrix in implicit dynamics and eigenvalue solutions.

Card 1	1	2	3	4	5	6	7	8
Variable	IFLAG							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
IFLAG	Consistent mass matrix flag: EQ.0: Use the standard lumped mass formulation (default) EQ.1: Use the consistent mass matrix.

Remarks:

The consistent mass matrix formulation is currently available for:

- three-node and four-node shell elements,
- solid elements types 1, 2, 10, 15, 16, and 18 (see [*SECTION_SOLID](#)),
- all thick shell element types, and
- beam types 1, 2, 3, 4, and 5 (see [*SECTION_BEAM](#)).

All other element types continue to use a lumped mass matrix.

CONTROL_IMPLICIT_DYNAMICS**CONTROL*****CONTROL_IMPLICIT_DYNAMICS_{OPTION}**

Available options include:

<BLANK>

DYN

SPR

Purpose: Activate implicit dynamic analysis and define time integration constants (see also *CONTROL_IMPLICIT_GENERAL). The DYN option allows setting controls specifically for the dynamic relaxation phase. The SPR option allows setting control specifically for the springback phase.

Card 1	1	2	3	4	5	6	7	8
Variable	IMASS	GAMMA	BETA	TDYBIR	TDYDTH	TDYBUR	IRATE	ALPHA
Type	I	F	F	F	F	F	I	F
Default	0	.50	.25	0.0	10^{28}	10^{28}	0	0.0

Rotational Dynamics Card. If ALPHA ≤ -1 , specify |ALPHA| cards with this format.

Card 2	1	2	3	4	5	6	7	8
Variable	PSID	ANGLE						
Type	I	F						
Default	none	90.0						

VARIABLE**DESCRIPTION**

IMASS

Implicit analysis type:

LT.0: -IMASS is a curve ID with ordinate values used to control the amount of implicit dynamic effects applied to the analysis. TDYBIR, TDYDTH and TDYBUR are ignored with this option.

EQ.0: Static analysis

*CONTROL

*CONTROL_IMPLICIT_DYNAMICS

VARIABLE	DESCRIPTION
	EQ.1: Dynamic analysis using Newmark time integration. EQ.2: Dynamic analysis by modal superposition following the solution of the eigenvalue problem EQ.3: Dynamic analysis by modal superposition using the eigenvalue solution in the d3eigv files that are in the runtime directory.
GAMMA	Newmark time integration constant (see Remark 2).
BETA	Newmark time integration constant (see Remark 2).
TDYBIR	Birth time for application of dynamic terms. See Figure 12-67 .
TDYDTH	Death time for application of dynamic terms.
TDYBUR	Burial time for application of dynamic terms.
IRATE	Rate effects switch (see Remark 4): EQ.-1: Rate effects are on in constitutive models even in implicit statics EQ.0: Rate effects are on in constitutive models, except implicit statics EQ.1: Rate effects are off in constitutive models EQ.2: Rate effects are off in constitutive models for both explicit and implicit.
ALPHA	Composite time integration constant (see Remark 2): GT.0: Bathe composite scheme is activated. LT.0.AND.GT.-1: HHT scheme is activated, LE.-1: Specify part sets for finite rotational dynamics.
PSID	Part set ID for a body undergoing rotational (spinning) motion
ANGLE	Target angle increment during a single time step in degrees

Remarks:

1. **Equilibrium equations and dynamic terms.** For the dynamic problem, the linearized equilibrium equations may be written in the form

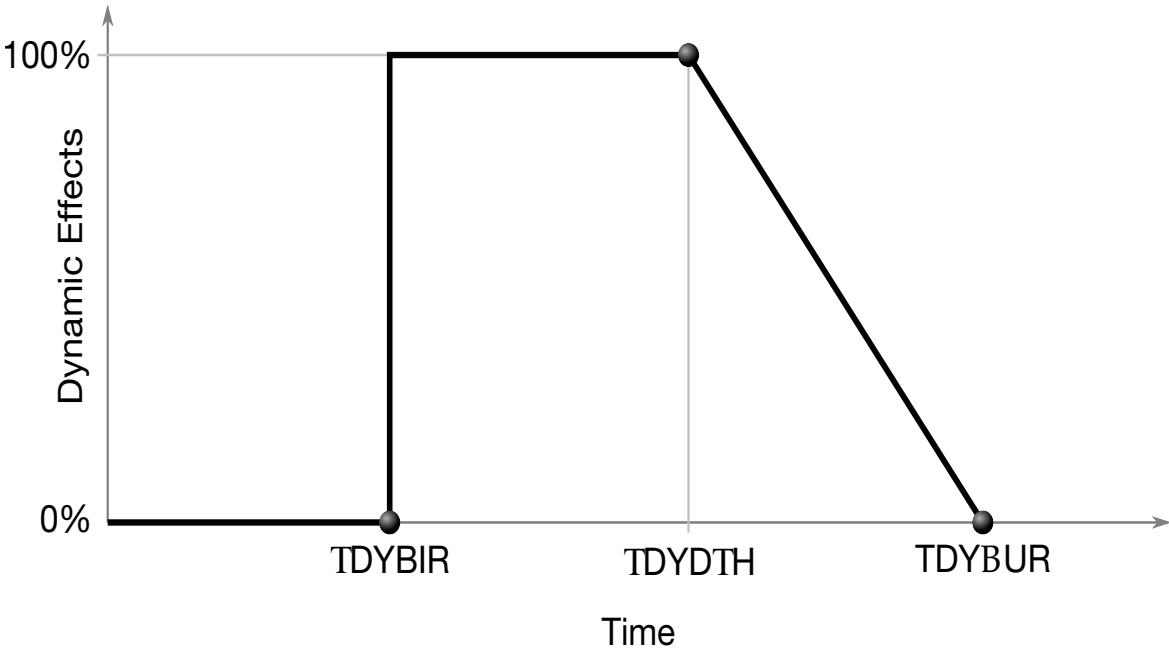


Figure 12-67. Birth, death, and burial time for implicit dynamics. The terms involving **M** and **D** are scaled by a factor ranging between 1 and 0 to include or exclude dynamical effects, respectively.

$$\mathbf{M}\ddot{\mathbf{u}}^{n+1} + \mathbf{D}\dot{\mathbf{u}}^{n+1} + \mathbf{K}_t(\mathbf{x}^n)\Delta\mathbf{u} = \mathbf{P}(\mathbf{x}^n)^{n+1} - \mathbf{F}(\mathbf{x}^n) ,$$

where

M = lumped mass matrix

D = damping matrix

$\mathbf{u}^{n+1} = \mathbf{x}^{n+1} - \mathbf{x}^0$ = nodal displacement vector

$\dot{\mathbf{u}}^{n+1}$ = nodal point velocities at time $n + 1$

$\ddot{\mathbf{u}}^{n+1}$ = nodal point acceleration at time $n + 1$

Between the birth and death times 100% of the dynamic terms, that is, the terms involving **M** and **D**, are applied. Between the death and burial time, the dynamic terms are decreased linearly with respect to time until 0% of the dynamic terms are applied after the burial time. This feature is useful for problems that are initially singular because the parts are not in contact initially such as in metal stamping. For these problems dynamics is required for stable convergence. When contact is established, the problem becomes well conditioned and the dynamic terms are no longer required for stable convergence. For such problems setting the death time to be after contact is established and the burial time for 2 or 3 time steps after the death time is recommended.

For problems with more extensive loading and unloading patterns the amount of dynamic effects added to the model can be controlled by using a load curve; see IMASS < 0. This curve should have ordinate values between 0.0 and 1.0. The user should use caution in ramping the load curve and the associated dynamic

effects from 1.0 to 0.0. Such a ramping down should take place over 2 or 3 implicit time steps.

2. **Time integration.** The time integration is by default the unconditionally stable, one-step, Newmark- β time integration scheme

$$\begin{aligned}\ddot{\mathbf{u}}^{n+1} &= \frac{\Delta \mathbf{u}}{\beta \Delta t^2} - \frac{\dot{\mathbf{u}}^n}{\beta \Delta t} - \frac{1}{\beta} \left(\frac{1}{2} - \beta \right) \ddot{\mathbf{u}}^n \\ \dot{\mathbf{u}}^{n+1} &= \dot{\mathbf{u}}^n + \Delta t (1 - \gamma) \ddot{\mathbf{u}}^n + \gamma \Delta t \ddot{\mathbf{u}}^{n+1} \\ \mathbf{x}^{n+1} &= \mathbf{x}^n + \Delta \mathbf{u}\end{aligned}$$

Here, Δt is the time step size, and β and γ are the free parameters of integration. For $\gamma = 1/2$ and $\beta = 1/4$, the method reduces to the trapezoidal rule and is energy conserving. If

$$\begin{aligned}\gamma &> \frac{1}{2} \\ \beta &> \frac{1}{4} \left(\frac{1}{2} + \gamma \right)^2,\end{aligned}$$

numerical damping is induced into the solution leading to a loss of energy and momentum.

The Newmark method, and the trapezoidal rule in particular, is known to lack the robustness required for simulating long term dynamic implicit problems. Even though numerical damping may improve the situation from this aspect, it is difficult to know how to set γ and β without deviating from desired physical properties of the system. In the literature, a vast number of *composite* time integration algorithms have been proposed to handle this, and a family of such methods is implemented and governed by the value of α (ALPHA, parameter 8 on Card 1).

For $\alpha > 0$, every other implicit time step is a three point backward Euler step given as

$$\begin{aligned}\ddot{\mathbf{u}}^{n+1} &= \frac{(1 + \alpha)}{\Delta t} (\dot{\mathbf{u}}^{n+1} - \dot{\mathbf{u}}^n) - \frac{\alpha}{\Delta t_-} (\dot{\mathbf{u}}^n - \dot{\mathbf{u}}^{n-1}) \\ \dot{\mathbf{u}}^{n+1} &= \frac{(1 + \alpha)}{\Delta t} \Delta \mathbf{u} - \frac{\alpha}{\Delta t_-} \Delta \mathbf{u}_-\end{aligned}$$

where $\Delta t_- = t^n - t^{n-1}$ and $\Delta \mathbf{u}_- = \mathbf{u}^n - \mathbf{u}^{n-1}$ are constants. Because of this three step procedure, the method is particularly suitable for nodes/bodies undergoing curved motion as it better accounts for curvature than the default Newmark step. For $\alpha = 1/2$, and default values of γ and β , the method defaults to the Bathe time integration scheme, Bathe [2007], and is reported to preserve energy and momentum to a reasonable degree. The improvement in stability over the Newmark method is primarily attributed to numerical dissipation, but fortunately this dissipation appears to mainly be due to damping of high frequency content

and the underlying physics is therefore not affected as such; see Bathe and Noh [2012].

For a negative value of ALPHA (strictly between -1 and 0), the HHT, Hilber-Hughes-Taylor [1977], scheme is activated. This scheme is similar to that of the Newmark method, but the equilibrium is sought at time step $n + 1 + \alpha$ instead of at $n + 1$. As a complement to the Newmark scheme above, we introduce

$$\begin{aligned}\dot{\mathbf{u}}^\alpha &= -\alpha \dot{\mathbf{u}}^n + (1 + \alpha) \dot{\mathbf{u}}^{n+1} \\ \mathbf{x}^\alpha &= -\alpha \mathbf{x}^n + (1 + \alpha) \mathbf{x}^{n+1}\end{aligned}$$

and solve a modified system of equilibrium equations

$$\mathbf{M}\ddot{\mathbf{u}}^{n+1} + \mathbf{D}\dot{\mathbf{u}}^\alpha + \mathbf{F}(\mathbf{x}^\alpha) = \mathbf{P}(\mathbf{x}^\alpha).$$

This method is stable for $-1/3 \leq \alpha \leq 0$ and $\gamma = (1 - 2\alpha)/2$ and $\beta = (1 - \alpha)^2/4$, which becomes the default values of γ and β if not explicitly set. Parameter α controls the amount of dissipation in the problem; for $\alpha = 0$ an undamped Newmark scheme is obtained, whereas $\alpha = -1/3$ introduces significant damping. From the literature, a value of $\alpha = -0.05$ appears to be a good choice.

Finite rotational dynamics can be activated for a part set by specifying a negative integer for ALPHA, the absolute value of ALPHA then indicates for how many rotating bodies this feature is activated. This option assumes that each body, meaning the parts within each part set PSID, is “almost” a rigid body in the sense that

- the assembly as a whole undergoes relatively large rigid body motion and small deformation in each step;
- the part set includes all parts that make up the assembly;
- and the assembly is not merged to some other elements in the model by sharing nodes, that is, interaction with other parts should be through joints, contacts, and similar constraints.

The target angle, ANGLE, is meant to be used as a reasonable (in terms of accuracy and convergence) rotation angle for each time step and may vary depending on application. For relatively simple inputs a value of 90 may be ok while for advanced applications it may have to be reduced. The time step will be adjusted to not exceed this target angle. The intention with this option is to use Newmark time integration with large time steps while maintaining high accuracy and robustness and hence serves as an undamped alternative to the damped Bathe and HHT schemes.

3. **Modal superposition.** When modal superposition is invoked, NEIGV on *CONTROL_IMPLICIT_EIGENVALUE indicates the number of modes to be used. With modal superposition, stresses are computed only for linear shell formulation 18.

4. **Rate effects in constitutive models.** The option IRATE is currently supported by the following material models: 3, 15, 24, 26, 57, 58, 63, 73, 75, 77, 81, 83, 89, 98, 103, 123, 124, 133, 142, 153, 157, 179, 181, 183, 187, 199, 215, 224, 225, 240, 251, 252, 264, 269, 273, 280, and GISSMO (see *MAT_ADD_DAMAGE). IRATE = 2 is also supported for material model 1 when used with truss elements.

***CONTROL_IMPLICIT_EIGENVALUE**

Purpose: Activates implicit eigenvalue analysis and defines associated input parameters (see also [*CONTROL_IMPLICIT_GENERAL](#)). The available methods are Block Shift and Invert Lanczos, ARPACK for the nonsymmetric eigenvalue problem (see [Remark 3](#)), MCMS (SMP only) for problems requiring the computation of thousands of eigenmodes, Fast Lanczos for computing thousands of approximate eigenmodes in MPP, LOBPCG for problems requiring a small number of eigenmodes, and Sectoral Symmetry for problems with sectoral (rotational) symmetry.

Card Summary:

Card 1. This card is required.

NEIG	CENTER	LFLAG	LFTEND	RFLAG	RHTEND	EIGMTH	SHFSCL
------	--------	-------	--------	-------	--------	--------	--------

Card 2. Card 2 and beyond are optional. If Card 3a or 3b are included, then Card 2 must be included; Card 2 can be a blank line in this case, so default values are used. Similarly, Card 4 requires Cards 2 and 3 (if applicable) to be included.

ISOLID	IBEAM	ISHELL	ITSHELL	MSTRES	EVDUMP	MSTRSCL	
--------	-------	--------	---------	--------	--------	---------	--

Card 3a. Card 3a is read only when EIGMTH = 2, 3, 5, or 6. It is optional.

		IPARM3					RPARM4
--	--	--------	--	--	--	--	--------

Card 3b. Card 3b is read only when EIGMTH = 101. It is optional.

IPARM1	IPARM2	IPARM3	IPARM4	RPARM1			RPARM4
--------	--------	--------	--------	--------	--	--	--------

Card 3c. Card 3c is read only when EIGMTH = 102. It is optional.

IPARM1	IPARM2	IPARM3		RPARM1	RPARM2		RPARM4
--------	--------	--------	--	--------	--------	--	--------

Card 3d. Card 3d is read only when EIGMTH = 103. It is optional.

IPARM1	IPARM2	IPARM3		RPARM1	RPARM2		RPARM4
--------	--------	--------	--	--------	--------	--	--------

Card 3e. Card 3e is read only when EIGMTH = 111. It is optional.

IPARM1	IPARM2	IPARM3	IPARM4	IPARM5	IPARM6		
--------	--------	--------	--------	--------	--------	--	--

Card 4. This card is optional.

ROTSCL	EIGMSCL						
--------	---------	--	--	--	--	--	--

*CONTROL

*CONTROL_IMPLICIT_EIGENVALUE

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	NEIG	CENTER	LFLAG	LFTEND	RFLAG	RHTEND	EIGMTH	SHFSCL
Type	I	F	I	F	I	F	I	F
Default	none	0.0	0	-infinity	0	+infinity	2	0.0

VARIABLE	DESCRIPTION
NEIG	Number of eigenvalues to extract. This must be specified. The other parameters below are optional. See Remark 1 . LT.0: Curve ID = (-NEIG) used for intermittent eigenvalue analysis
CENTER	Center frequency. This option finds the nearest NEIG eigenvalues located about this value. See Remarks 2, 3 , and 10 .
LFLAG	Left endpoint finite flag (see Remarks 2 and 3): EQ.0: Left endpoint is -infinity. EQ.1: Left endpoint is LFTEND.
LFTEND	Left endpoint of the interval. Only used when LFLAG = 1. See Remarks 2, 3 , and 10 .
RFLAG	Right endpoint finite flag (see Remarks 2 and 3): EQ.0: Right endpoint is +infinity. EQ.1: Right endpoint is RHTEND.
RHTEND	Right endpoint of the interval. Only used when RFLAG = 1. See Remarks 2, 3 , and 10 .
EIGMTH	Eigenvalue extraction method (see Remark 3): EQ.2: Block Shift and Invert Lanczos (default). See Remark 2 . EQ.3: Lanczos with $[M] = [I]$ (for debugging only) EQ.5: Same as 3 but include Dynamic Terms EQ.6: Same as 2 but include Dynamic Terms

VARIABLE	DESCRIPTION							
	EQ.101: MCMS. See Remark 4 .							
	EQ.102: LOBPCG. See Remark 7 .							
	EQ.103: Fast Lanczos. See Remark 5 .							
	EQ.111: Sectoral Symmetry. See Remark 8 .							
SHFSCL	Shift scale. Generally, not used. See Remarks 3, 9 , and 10 .							

Card 2	1	2	3	4	5	6	7	8
Variable	ISOLID	IBEAM	ISHELL	ITSHELL	MSTRES	EVDUMP	MSTRSCL	
Type	I	I	I	I	I	I	F	
Default	0	0	0	0	0	0	0.001	

VARIABLE	DESCRIPTION
ISOLID	If nonzero, reset all solid element formulations to ISOLID for the implicit computations. It can be used for all implicit analyses, not just eigenvalue computations.
IBEAM	If nonzero, reset all beam element formulations to IBEAM for the implicit computations. It can be used for all implicit computations, not just eigenvalue computations.
ISHELL	If nonzero, reset all shell element formulations to ISHELL for the implicit computations. It can be used for all implicit computations, not just eigenvalue computations.
ITSHELL	If nonzero, reset all thick shell element formulations to ITSHELL for the implicit computations. It can be used for all implicit computations, not just eigenvalue computations.
MSTRES	Flag for computing the stresses for the eigenmodes (see Remark 11): EQ.0: Do not compute the stresses. EQ.1: Compute the stresses.

*CONTROL

*CONTROL_IMPLICIT_EIGENVALUE

VARIABLE	DESCRIPTION
EVDUMP	Flag for writing eigenvalues and eigenvectors to file Eigen_Vectors in SMP or files Eigen_Vectors.xxxx in MPP where xxxx is the process ID (see Remark 11): EQ.0: Do not write eigenvalues and eigenvectors. GT.0: Write eigenvalues and eigenvectors using an ASCII format. LT.0: Write eigenvalues and eigenvectors using a binary format.
MSTRSCL	Scaling for computing the velocity based on the mode shape for the stress computation. See Remark 11 .

Block Shift and Invert Lanczos Card. Omitted unless EIGMTH = 2, 3, 5, or 6. Additional card for eigenvalue extraction method Block Shift and Invert Lanczos and its variations.

Card 3a	1	2	3	4	5	6	7	8
Variable			IPARM3					RPARM4
Type			I					F
Default			{Ø}					0.0

VARIABLE	DESCRIPTION
IPARM3	Node set to reduce output of entries from each eigenvector. See Remark 6 .
RPARM4	Control output of eigenvectors to the d3eigv database (see Remark 6): LT.0.0: No output EQ.0.0: Write all eigenmodes to d3eigv. GT.0.0: Write only the first RPARM4 eigenmodes to d3eigv.

CONTROL_IMPLICIT_EIGENVALUE**CONTROL**

MCMS Card. Omitted unless EIGTH = 101. Additional card for eigenvalue extraction method MCMS.

Card 3b	1	2	3	4	5	6	7	8
Variable	IPARM1	IPARM2	IPARM3	IPARM4	RPARM1			RPARM4
Type	I	I	I	I	F			F
Default	100	↓	{Ø}	1500	4.0			0.0

VARIABLE	DESCRIPTION
IPARM1	Minimum block size for the Cholesky factorization
IPARM2	Maximum block size for the Cholesky factorization. The default is the model size.
IPARM3	Node set ID specifying particular nodes in the model where increased accuracy is desired. See Remark 4 .
IPARM4	MCMS minimum group/substructure size. See Remark 4 .
RPARM1	Eigenvalue expansion factor τ . See Remark 4 .
RPARM4	Control output of eigenvectors to the d3eigv database: LT.0.0: No output EQ.0.0: Write all eigenmodes to d3eigv. GT.0.0: Write only the first RPARM4 eigenmodes to d3eigv.

LOBPCG Card. Omitted unless EIGMTH = 102. Additional card for eigenvalue extraction method LOBPCG.

Card 3c	1	2	3	4	5	6	7	8
Variable	IPARM1	IPARM2	IPARM3		RPARM1	RPARM2		RPARM4
Type	I	I	I		F	F		F
Default	100	100	{Ø}		10 ⁻¹²	10 ⁻⁵		0.0

*CONTROL

*CONTROL_IMPLICIT_EIGENVALUE

VARIABLE	DESCRIPTION
IPARM1	Maximum number of iterations
IPARM2	Block size
IPARM3	Node set to reduce output of entries from each eigenvector. See Remark 6 .
RPARM1	Convergence tolerance
RPARM2	Flag for using a Block Low-Rank (BLR) factorization for the preconditioner: GT.0.0: Use BLR preconditioner with compression threshold RPARM2. LT.0.0: Use exact factorization instead of a BLR preconditioner
RPARM4	Control output of eigenvectors to the d3eigv database (see Remark 6): LT.0.0: No output EQ.0.0: Write all eigenmodes to d3eigv. GT.0.0: Write only the first RPARM4 eigenmodes to d3eigv.

Fast Lanczos Card. Omitted unless EIGTH = 103. Additional card for eigenvalue extraction method Fast Lanczos.

Card 3d	1	2	3	4	5	6	7	8
Variable	IPARM1	IPARM2	IPARM3		RPARM1	RPARM2		RPARM4
Type	I	I	I		F	F		F
Default	40	40	0		10^6	0.80		0.0

VARIABLE	DESCRIPTION
IPARM1	Maximum number of Block Lanczos iterations
IPARM2	Block size for the Block Lanczos recurrence
IPARM3	Node set to reduce output of entries from each eigenvector. See Remark 6 .

VARIABLE	DESCRIPTION
RPARM1	First shift. Supply the approximate value of the 100 th eigenvalue.
RPARM2	Shift factor. This field reduces the aggressiveness of the shift logic.
RPARM4	Control output of eigenvectors to the d3eigv database (see Remark 6): LT.0.0: No output EQ.0.0: Write all eigenmodes to d3eigv. GT.0.0: Write only the first RPARM4 eigenmodes to d3eigv.

Sectoral Symmetry Card. Omitted unless EIGHTH = 111. Additional card for eigenvalue extraction method MCMS.

Card 3e	1	2	3	4	5	6	7	8
Variable	IPARM1	IPARM2	IPARM3	IPARM4	IPARM5	IPARM6		
Type	I	I	I	I	I	I		
Default	none	0	none	none	none	0		

VARIABLE	DESCRIPTION
IPARM1	Node set ID for nodes on the left surface of the sector
IPARM2	Node set ID for nodes on the axis of rotation. EQ.0: No nodes on the axis of rotation (default)
IPARM3	Node set ID for nodes on the left surface of the sector
IPARM4	Number of sectors
IPARM5	Harmonic index
IPARM6	Vector ID for the axis of rotation. EQ.0: Axis of rotation is the global z-axis (default)

*CONTROL

*CONTROL_IMPLICIT_EIGENVALUE

Optional Control Card.

Card 4	1	2	3	4	5	6	7	8
Variable	ROTSCL	EIGMSCL						
Type	F	I						
Default	0.001	0						

VARIABLE	DESCRIPTION
ROTSCL	Scale factor for the inertia of rotational degrees of freedom. See Remark 13 . EQ.0.0: Default value of 0.001 EQ.1.0: Inertia not scaled.
EIGMSCL	For intermittent eigenvalue extractions, control whether to use the original (unscaled) or the scaled mass (when mass scaling is active through parameter DT2MS on *CONTROL_TIMESTEP). Prior to version R15 (unless noted otherwise in the release notes), the scaled mass was incorrectly used. We added this option for backwards compatibility. EQ.0: Use original mass (default). EQ.1: Use scaled mass.

Remarks:

1. **Performing eigenvalue analysis.** To perform an eigenvalue analysis, activate the implicit method by selecting IMFLAG = 1 on [*CONTROL_IMPLICIT_GENERAL](#), and indicate a nonzero value for NEIG above. By default, the lowest NEIG eigenvalues will be found. Specifying a nonzero center frequency causes LS-DYNA to find the NEIG eigenvalues nearest to CENTER.

When NEIG > 0, LS-DYNA computes eigenvalues at time = 0 and then terminates.

When NEIG < 0, LS-DYNA performs an intermittent eigenvalue analysis, which can be in both transient and dynamic relaxation (DR) phases. This is a transient/DR simulation during which loads are applied, with eigenvalues computed periodically during the simulation. Changes in geometry, stress, material, and

contact conditions will affect the eigenvalues. The transient simulation can be either implicit or explicit according to IMFLAG = 1 or IMFLAG = 6, respectively, on [*CONTROL_IMPLICIT_GENERAL](#). The curve ID = -NEIG indicates when to extract eigenvalues and how many to extract. Define one curve point at each desired extraction time, with a function value equal to the number of eigenvalues needed at that time. If requiring eigenvalues during the dynamic relaxation phase, set SIDR = 2 in [*DEFINE_CURVE](#), and then set the extraction time to a negative value, corresponding to the pseudo time during the dynamic phase. LS-DYNA produces a d3plot database for the transient solution results. It produces consecutively numbered d3eigv and eigout databases for each intermittent extraction. Each database's analysis title indicates the extraction time.

2. **Block Shift and Invert Lanczos method.** The Block Shift and Invert Lanczos code is from BCSLIB-EXT, Boeing's Extreme Mathematical Library.

When using Block Shift and Invert Lanczos, you can specify a semifinite or finite interval region in which to compute eigenvalues. Setting LFLAG = 1 changes the left endpoint from -infinity to the value specified by LFTEND. Setting RFLAG = 1 changes the right endpoint from +infinity to the values given by RHTEND. If the interval includes CENTER (default value of 0.0), LS-DYNA computes the NEIG eigenvalues nearest to CENTER. If the interval does not include CENTER, LS-DYNA computes NEIG eigenvalues with the smallest magnitude.

If all of the eigenvalues are needed in an interval where both endpoints are finite, input a large number for NEIG. The software automatically computes the number of eigenvalues in the interval and decreases NEIG to that value. The most general problem specification is to compute NEIG eigenvalues nearest CENTER in the interval [LFTEND,RHTEND]. Computing the lowest NEIG eigenvalues is equivalent to computing the NEIG eigenvalues nearest 0.0.

3. **Nonsymmetric eigenvalue problems.** When the field LCPACK in [*CONTROL_IMPLICIT_SOLVER](#) equals 3, LS-DYNA assumes the eigenvalue problem is nonsymmetric. This feature allows you to compute the eigensolutions for problems with both nonsymmetric terms in the stiffness matrix and with damping terms. *Note that the fields CENTER, LFLAG, LFTEND, RFLAG, RHTEND, EIGMTH, and SHFSCL are ignored for the nonsymmetric eigenvalue problem.*

By setting LCPACK to 3, ARPACK will automatically be chosen to solve the eigenvalue problem. All damping terms in the model will also be added to the first order terms of the eigenvalue problem. Implicit Rotational Dynamics will also cause the use of ARPACK due to the added first order terms.

4. **MCMS method.** MCMS is an SMP only implementation of the AMLS algorithm available as of LS-DYNA R11. See the discussion of Fast Lanczos in [Remark 5](#) for an MPP capability. The target application is automotive NVH models, where

thousands of approximate eigenmodes are computed for Frequency Response Analysis. MCMS calculates all the eigenmodes with eigenvalues in the interval $[-\infty, \text{RHTEND}]$. NEIG must be positive to turn on the eigencomputation. If the number of eigenvalues in the interval is less than NEIG, then the next few eigenmodes will be included to reach NEIG. NEIG should not be too large compared to the expected number of eigenvalues in the interval, as the accuracy of the eigenmodes past RHTEND will be less than that of the eigenmodes before RHTEND.

Increasing the group/substructure size (IPARM4) and the eigenvalue expansion factor (RPARM1) improves the accuracy with the cost of increasing the time of the MCS eigen computation. To improve accuracy at a few nodes, such as the nodes of the mounting bracket for an automotive chassis, specify these nodes in a node set using IPARM3.

5. **Fast Lanczos method.** Fast Lanczos is designed to compute thousands of approximate eigenmodes for Noise-Vibration-Harshness (NVH) applications. It is an implementation of the Block Shift and Invert Lanczos algorithm used by EIGMTH = 2 with many adjustments to make it faster and less accurate. The goal of this method is to compute the lowest thousands of eigenmodes with an accuracy of 1% for the eigenvalues. It is assumed that the factorization computed by the MF solver (LSOLVR = 2 on [*CONTROL_IMPLICIT_SOLVER](#)) ends up with the factorization in memory. At each shift, Block Lanczos runs MXITER iterations using a blocksize of BLKSIZ. These parameters may be adjusted to allow the computation to fit entirely in memory.

Fast Lanczos requires an approximate value of the 100th eigenvalue (RPARM1). The default of 10^6 works well for most problems, but the choice of units and the actual model can impact this value. For example, for models with milliseconds as the time unit, we recommend using a value with an order of magnitude of 10^2 . The most robust way to find this value is to compute 100 modes with Standard Lanczos (EIGMTH = 2) and use the 100th eigenvalue from that computation.

RPARM2 controls the aggressiveness of the shift logic.

The interval controls of RFLAG and RHTEND are honored for the case where you desire all of the eigenmodes less than RHTEND.

6. **Reducing the cost of outputting eigenvectors for EIGMTH = 2, 3, 5, 6, 102, and 103.** The cost of outputting the entirety of eigenvectors to the d3mode database can be substantial. Some applications using the eigenmodes only need a subset of the entries for each eigenvector. For EIGMTH = 2, 3, 5, 6, 102, and 103, set IPARM3 to a node set to define this subset. Then, the output for that subset for each eigenvector is written to an LSDA database called **LanczosEigenVectors** for EIGMTH = 2, 3, 5, and 6, **LOBPCGEigenVectors** for EIGMTH = 102, and

FastLnzEigenVectors for EIGMTH = 103. Using IPARM3 can vastly reduce the I/O requirements.

Some post-processing applications require only a subset of the eigenvectors. RPARM4 for the previously mentioned eigenvalue extraction methods controls the number of eigenvectors written to the d3eigv database. The choice is no eigenvectors, some eigenvectors, or all of the eigenvectors. A node set specified with IPARM3 and RPARM4 > 0.0 can be used at the same time. If IPARM3 is nonzero and RPARM4 is zero or blank, no eigenmodes are written to d3eigv. The parameter MSTRES specifies whether to perform stress computations. Computing stresses is expensive for large finite element models and when outputting thousands of eigenvectors. Care should be used in setting MSTRES in combination with RPARM4. Visualizing the first few eigenmodes for model verification can be accomplished by setting RPARM4 to a small value, such as 10.

7. **LOBPCG method.** LOBPCG [1] is a preconditioned eigensolver that can be faster and use less memory than Lanczos when computing a few eigenmodes. The preconditioner is based on a Block Low-Rank (BLR) factorization. The user can adjust the BLR compression threshold as well as various parameters. It is available in SMP as of LS-DYNA R11 and in MPP as of R14.
8. **Sectoral symmetry method.** Sectoral symmetry uses the rotational symmetry of a model to reduce the model size to that of a sector but at the cost of performing many eigenvalue computations based on the harmonic index. This method requires specifying the nodes on the left surface of the sector, the nodes on the right surface of the sector, and any nodes in the sector on the axis of rotation. The default for the axis of rotation is the global Z-axis. The axis is otherwise given by a vector ID from [*DEFINE_VECTOR](#) or [*DEFINE_VECTOR_NODES](#). The model is checked to verify that the right nodes align with the left nodes when properly rotated given the axis and the number of sectors.

The eigenvalue problem is partitioned based on the left, center, right, and interior node sets. The boundary condition given by the harmonic index is solved.

Note that the solver only computes a subset of the eigenmodes, those associated with a given harmonic index. Obtaining the complete spectrum requires multiple runs with harmonic indices varying from 0 to the number of sectors – 1.

The sectoral symmetry method is available for both SMP and MPP, but in MPP it only works for jobs running on one processor.

9. **Initial shift.** For some problems, it is helpful to override the internal heuristic for picking a starting point for Lanczos shift strategy, that is, the initial shift. In these rare cases, you may specify the initial shift via the parameter SHFSCL. SHFSCL should be in the range of the first few nonzero frequencies.

10. **Units.** Parameters CENTER, LFTEND, RHTEND, and SHFSCL are in units of Hertz for eigenvalue problems. These four parameters, along with LFLAG and RFLAG, apply to buckling problems. For buckling problems, CENTER, LFTEND, RHTEND, and SHFSCL are in units of the eigenvalue spectrum.
11. **Output files.** Eigenvectors are written to an auxiliary binary plot database named d3eigv, which is automatically created. These can be viewed using a postprocessor in the same way as a standard d3plot database. The time value associated with each eigenvector plot is the corresponding frequency in units of cycles per unit time. A summary table of eigenvalue results is printed to the eigout file. In addition to the eigenvalue results, modal participation factors and modal effective mass tables are written to the eigout" file. The user can export individual eigenvectors using LS-PrePost.

Set MSTRES = 1 to request computing the stresses and write them to d3eigv. A velocity is computed by dividing the displacements from the eigenmode by MSTRSCL. The element routine then computes the stresses based on this velocity, but LS-DYNA inversely scales those stresses by MSTRSCL before writing them to d3eigv. Thus, MSTRSCL does not affect the results of linear element formulations. The strains associated with the stresses output using the MSTRES option can be obtained by setting the STRFLG on *DATABASE_EXTENT_BINARY.

Setting EVDUMP to a nonzero value causes writing the eigenvalues and eigenvectors to file Eigen_Vectors in SMP or files Eigen_Vectors.xxxx in MPP. In MPP, xxxx is the process ID. For example, for a run with 128 processes, LS-DYNA outputs files Eigen_Vectors.0001, Eigen_Vectors.0002, ..., Eigen_Vectors.0127. The sign of EVDUMP determines the file type for both SMP and MPP. For EVDUMP > 0, the files are output in ASCII format, while for EVDUMP < 0, they are output in a simple binary format. The binary format reduces file space. The written eigenvectors are orthonormal with respect to the mass matrix.

12. **Solver controls.** The print control parameter, LPRINT, and ordering method parameter, ORDER, from the [*CONTROL_IMPLICIT_SOLVER](#) keyword card also apply to the various eigensolvers.
13. **Inertia of rotational DOF.** By default, the inertia of the rotational degrees of freedom is scaled by a thousand. This scaling causes an increase in the rotational eigenfrequencies and exclusion of the associated eigenmodes (which often are of little interest) from the lowest NEIG eigenmodes. This approach is acceptable when no coupling exists between the translational and rotational degrees of freedom. Coupling occurs when the neutral and reference planes do not coincide, such as in nonsymmetric laminates or when NLOC ≠ 0. When coupling exists, use correct rotational inertia. Setting ROTSCL = 1.0 tells LS-DYNA to use the correct rotational inertia (the same as in explicit analyses).

References:

- [1] A. V. Knyazev, Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. *SIAM journal on scientific computing*, 23(2), 517-541 (2001).

*CONTROL

*CONTROL_IMPLICIT_FORMING

*CONTROL_IMPLICIT_FORMING_{OPTION}

Available options include:

<BLANK>

DYN

SPR

Purpose: This keyword is used to perform implicit static analysis, especially for metal forming processes, such as gravity loading, binder closing, flanging, and stamping sub-assembly simulation. A systematic study had been conducted to identify the key factors affecting implicit convergence, and the preferred values are automatically set with this keyword. In addition to forming application, this keyword can also be used in other applications, such as dummy loading and roof crush, etc. The DYN option allows setting controls specifically for the dynamic relaxation phase. The SPR option allows setting controls specifically for the springback phase.

A related keyword is *CONTACT_AUTO_MOVE, where an empty distance between the tool and blank can be automatically eliminated in a combined gravity and closing simulation for implicit static simulations.

Card 1	1	2	3	4	5	6	7	8
Variable	IOPION	NSMIN	NSMAX	BIRTH	DEATH	PENCHK	DTO	
Type	I	I	I	F	F	F	F	
Default	1	↓	2	0.0	10 ²⁰	0.0	↓	

VARIABLE	DESCRIPTION
IOPION	Solution type: EQ.1: Gravity loading simulation; see Remark 2 and examples: Gravity Loading, Flanging Simulation Using IOPION = 1 , and Switching between Implicit Dynamic and Implicit Static for Gravity Loading . EQ.2: Binder closing and flanging simulation; see Remark 2 and examples: Binder Closing , Binder Closing with Real Beads , and Flanging .
NSMIN	Minimum number of implicit steps for IOPION = 2.

VARIABLE	DESCRIPTION
NSMAX	Maximum number of implicit steps for IOPTION = 2.
BIRTH	Birth time to activate this feature.
DEATH	Death time.
PENCHK	Relative allowed penetration with respect to the part thickness in contact for IOPTION = 2.
DT0	Initial time step size that overrides the DT0 field defined in *CONTROL_IMPLICIT_GENERAL

Remarks:

1. **Implicit Settings.** This keyword provides a simplified interface for implicit static analysis. If no other implicit cards are used, the stiffness matrix is reformed every iteration. Convergence tolerances (DCTOL, ECTOL, etc.) are automatically set and recommended not to be changed. In almost all cases, only two additional implicit control cards (*CONTROL_IMPLICIT_GENERAL and *CONTROL_IMPLICIT_AUTO) may be needed to control the time step size with fields DT0, DTMIN and DTMAX.

If multiple steps are required for IOPTION = 1, *CONTROL_IMPLICIT_GENERAL must be placed *after* *CONTROL_IMPLICIT_FORMING with DT0 specified as a certain fraction of the ENDTIM (see *CONTROL_TERMINATION). Otherwise, even with DT0 specified as a fraction of the ENDTIM, only one step (with step size of ENDTIM) will be performed.

2. **Contact.** As always, the field IGAP should be set to "2" in *CONTACT_FORMING... cards for a more realistic contact simulation in forming. The contact type *CONTACT_FORMING_SURFACE_TO_SURFACE is recommended to be used with implicit analysis.

Smaller penalty stiffness scale factor SLSFAC on *CONTROL_CONTACT produces a certain amount of contact penetration but yields faster simulation time, and therefore is recommended for gravity and closing (in case of no physical beads) simulation. Subsequent forming process is likely to follow and contact conditions will be reestablished there, where a tighter, default SLSFAC with a value of 0.1 should be used.

3. **Element Type.** It is recommended that the fully integrated element type 16 is to be used for all implicit calculations. For solids, type "-2" is recommended.

4. **Double Precision.** Executable with double precision is to be used for all implicit calculations.
5. **MPP.** MPP is more efficient for models with over 100,000 deformable elements.

Gravity Loading Example:

An example of the implicit gravity is provided below, where a blank is loaded with gravity into a toggle die. A total of five steps are used, controlled by the field DT0. The results are shown in [Figure 12-68](#). If this binder closing is simulated with explicit dynamics, the inertia effects on the blank need to be reduced since contact with the upper binder only happens along the periphery and a large middle portion of the blank is not driven or supported by anything. With implicit static method, there is no inertia effect at all on the blank during the closing, and no tool speed, time step size, etc. to be concerned about.

The implicit gravity application for both air and toggle draw processes is available through LS-PrePost in Metal Forming Application → eZ Setup (<http://ftp.lstc.com/-anonymous/outgoing/lsprepost/>).

```
*KEYWORD
*PARAMETER
:
*CONTROL_TERMINATION
1.0
$-----1-----2-----3-----4-----5-----6-----7-----8
*CONTROL_IMPLICIT_FORMING
$ IOPTION
    1
*CONTROL_IMPLICIT_GENERAL
$ IMFLAG      DT0
    1      0.2
*CONTROL_CONTACT
$ SLSFAC     RWPNAL   ISLCHK   SHLTHK   PENOPT   THKCHG   ORIEN
    0.03      0.0       2          1          4          0          4
$ USRSTR     USRFAC    NSBCS    INTERM    XPENE    SSTHK    ECDT    TIEDPRJ
    0          0        10         0          1.0        0
*PART
Blank
&blkpid &blksec &blkmid
*SECTION_SHELL
$ SID      EIFORM    SHRF      NIP      PROPT    QR/IRID    ICOMP    SETYP
&blksec      16      0.833     7      1.0
$ T1       T2       T3       T4      NLOC
&bthick,&bthick,&bthick,&bthick
*CONTACT_FORMING_SURFACE_TO_SURFACE
$ SURFA     SURFB     SURFATYP  SURFBTYP  SABOXID  SBBOXID   SAPR     SBPR
&blkuid &lpuuid      2          2
$ FS        FD        DC        VC        VDC      PENCHK    BT       DT
    0.12      0.0      0.0      0.0      20.0      0      0.0      1E+20
$ SFSA     SFSB     SAST      SBST      SFSAT    SFSBT     FSF      VSF
    1.0      1.0      0.0      &mstp
$ SOFT     SOFSCL   LCIDAB    MAXPAR    PENTOL   DEPTH     BSORT   FRCFRQ
    0
$ PENMAX   THKOPT   SHLTHK   SNLOG     1
$ IGAP     IGNORE    DPRFAC   DTSTIF
    2
```

```
:  
*LOAD_BODY_Z  
90994  
*DEFINE_CURVE_TITLE  
Body Force on blank  
90994  
0.0,9810.0  
10.0,9810.0  
*LOAD_BODY_PARTS  
&blkSID  
*END
```

Binder Closing Example:

An example of binder closing and its progression is shown in [Figures 12-69, 12-70, 12-71, and 12-72](#), using the NUMISHEET'05 deck lid inner, where a blank is being closed in a toggle die (modified). An adaptive level of three was used in the closing process. Gravity is and should be always applied at the same time, regardless of if a prior gravity loading simulation is performed or not, as listed at the end of the input deck. The presence of the gravity helps the blank establish an initial contact with the tool, thus improving the convergence rate. The upper binder is moved down by a closing distance (defined by a parameter &bindmv) using a displacement boundary condition (VAD = 2), with a simple linearly increased triangle-shaped load curve. The variable DT0 is set at 0.01, determined by the expected total deformation. The solver will automatically adjust based on the initial contact condition. The maximum step size is controlled by the variable DTMAX, and this value needs to be sufficiently small (<0.02) to avoid missing contact, but yet not too small causing a long running time. In some cases, this variable can be set to a larger value, but the current value works for most cases.

```
*KEYWORD  
*PARAMETER  
:  
*CONTROL_TERMINATION  
1.0  
*CONTROL_IMPLICIT_FORMING  
$ IOPTION      NSMIN      NSMAX  
    2          2         100  
*CONTROL_IMPLICIT_GENERAL  
$ IMFLAG       DT0  
    1          0.01  
*CONTROL_IMPLICIT_AUTO  
$ IAUTO        ITEOPT     ITEWIN      DTMIN      DTMAX  
    0          0          0         0.01      0.03  
*CONTROL_ADAPTIVE  
:  
*CONTROL_CONTACT  
$ SLSFAC       RWPNAL     ISLCHK      SHLTHK      PENOPT      THKCHG      ORIEN  
    0.03        0.0          2           1           4             0             4  
$ USRSTR       USRFAC     NSBCS       INTERM      XPENE       SSTHK       ECDT      TIEDPRJ  
    0          0          10          0          1.0            0  
$-----+---1-----+---2-----+---3-----+---4-----+---5-----+---6-----+---7-----+---8  
:  
*PART  
Blank  
$ PID        SECID      MID        EOSID      HGID       GRAV      ADPOPT      TMID  
  &blkpid   &blksec   &blkmid  
*SECTION_SHELL
```

*CONTROL

*CONTROL_IMPLICIT_FORMING

```
$      SID    ELMFORM      SHRF      NIP      PROPT      QR/IRID      ICOMP      SETYP
&blksec          16     0.833      7       1.0
$      T1      T2      T3      T4      NLOC
&bthick,&bthick,&bthick,&bthick
:
*CONTACT_FORMING_SURFACE_TO_SURFACE
$      SURFA      SURFB      SURFATYP      SURFBTYP      SABOXID      SBBOXID      SAPR      SBPR
&blkSID      &lpunSID      2      2
$      FS      FD      DC      VC      VDC      PENCHK      BT      DT
0.12      0.0      0.0      0.0      20.0      0      0.0      1E+20
$      SFSA      SFSB      SAST      SBST      SFSAT      SFSBT      FSF      VSF
1.0      1.0      0.0      &mstp
$      SOFT      SOFSCL      LCIDAB      MAXPAR      PENTOL      DEPTH      BSORT      FRCFRQ
0
$      PENMAX      THKOPT      SHLTHK      SNLOG      ISYM      I2D3D      SLDTHK      SLDSTF
$      IGAP      IGNORE      DPRFAC      DTSTIF
2
*CONTACT...
$-----1-----2-----3-----4-----5-----6-----7-----8
*BOUNDARY_PRESCRIBED_MOTION_RIGID
$      typeID      DOF      VAD      LCID      SF      VID      DEATH      BIRTH
&bindpid      3      2      3      -1.0      0
*DEFINE_CURVE
3
0.0,0.0
1.0,&bindmv
$-----1-----2-----3-----4-----5-----6-----7-----8
$ Activate gravity on blank:
*LOAD_BODY_Z
90994
*DEFINE_CURVE_TITLE
Body Force on blank
90994
0.0,9810.0
10.0,9810.0
*LOAD_BODY_PARTS
&blkSID
*END
```

Binder Closing with Real Beads Example:

Binder closing with real beads can also be done with implicit static, and with adaptive mesh. An example is shown in [Figure 12-73](#), where a hood outer is being closed implicitly. It is noted a small buckle can be seen near the draw bead region along the fender line. These kind of small forming effects can be more accurately detected with implicit static method.

The implicit static closing can now be set up in LS-PrePost Metal Forming Application → eZ Setup (<http://ftp.lstc.com/anonymous/outgoing/lsprepost/>).

Flanging Example:

An example flanging simulation using this feature is shown in [Figures 12-74, 12-75](#) and [12-76](#), with NUMISHEET'02 fender outer, where flanging is conducted along the hood line. A partial input is provided below, where DTMAX is controlled by a load curve for

contact and speed; a load curve input for DTMAX is usually not necessary. Gravity, pad closing and flanging were set to 10%, 10% and 80% of the total step size, respectively. The pad travels a distance of &padtrav starting at 0.1, when it is to be automatically moved to close the gap with the blank due to gravity loading (*CONTACT_AUTO_MOVE), and finishing at 0.2 and held in that position until the end. Flanging steel travels a distance of '&flgtrav' starting at 0.2 and completing at 1.0. A detailed section view of the simulation follows in [Figure 12-77](#).

```
*KEYWORD
*PARAMETER ...
*CONTROL_TERMINATION
1.0
*CONTROL_IMPLICIT_FORMING
$ IOPTION      NSMIN      NSMAX
      2          2          200
*CONTROL_IMPLICIT_GENERAL
      1          0.100
*CONTROL_IMPLICIT_AUTO
$ IAUTO        ITEOPT     ITEWIN      DTMIN      DTMAX
      0          0          0          0.005      -9980
*DEFINE_CURVE
9980
0.0,0.1
0.1,0.1
0.2,0.1
0.7,0.005
1.0,0.005
*CONTROL_ADAPTIVE...
*CONTROL_CONTACT...
*PART...
*SECTION_SHELL...
*CONTACT...
*CONTACT_FORMING_SURFACE_TO_SURFACE_ID_MPP
2
0,200,,3,2,1.005
$ SURFA        SURFB      SURFATYP    SURFBTYP    SABOXID    SBBOXID    SAPR      SBPR
  &blkSID      &padsID    2           2           VDC        PENCHK     BT        DT
  FS          FD          DC          VC          20.0       0          0.0       1E+20
  0.12        0.0         0.0         0.0         20.0       0          0.0       1E+20
$ SFSA        SFSB      SAST        SBST        SFSAT      SFSBT      FSF       VSF
  1.0         1.0         0.0         &mstp
$ SOFT        SOFSCL    LCIDAB     MAXPAR     PENTOL     DEPTH      BSORT     FRCFRQ
  0
$ PENMAX      THKOPT     SHLTHK     SNLOG      ISYM       I2D3D     SLDTHK     SLDSTF
$ IGAP        IGNORE     DPRFAC     DTSTIF
  2
*BOUNDARY_PRESCRIBED_MOTION_RIGID
$ typeID      DOF        VAD        LCID        SF        VID        DEATH     BIRTH
  &padpid     3           2           3           -1.0      0
  &flgpid     3           2           4           -1.0      0
*DEFINE_CURVE
3
0.0,0.0
0.1,0.0
0.2,&padtrav
1.0,&padtrav
*DEFINE_CURVE
4
0.0,0.0
0.2,0.0
1.0,&flgtrav
$ Activate gravity on blank:
```

*CONTROL

*CONTROL_IMPLICIT_FORMING

```
*LOAD_BODY_PARTS
&blkstd
*LOAD_BODY_Z
90994
*DEFINE_CURVE_TITLE
Body Force on blank
90994
0.0,9810.0
10.0,9810.0
*CONTACT_AUTO_MOVE
$      ID      ContID      VID      LCID      ATIME
      -1          2        89          3        0.1
*END
```

Flanging Simulation Using IOPTION = 1:

IOPTION = 1 can also be used for closing and flanging simulation, or other applications in which there are large plastic strains or deformation. This is used when an equal step size throughout the simulation is desired and is done by specifying the equal step size in the variable DT0 in *CONTROL_IMPLICIT_GENERAL, as shown in the following partial keyword deck where DT0 of 0.014 is chosen. An application of this is shown in [Figures 12-78 and 12-79](#).

```
*CONTROL_TERMINATION
1.0
*CONTROL_IMPLICIT_FORMING
$  IOPTION
    1
*CONTROL_IMPLICIT_GENERAL
$  IMFLAG      DT0
    1      0.014
```

Switching between Implicit Dynamic and Implicit Static for Gravity Loading:

For sheet blank gravity loading, it is now possible to start the simulation using implicit dynamic method and then switch to the implicit static method at a user defined time until completion. This feature is activated by setting the variable TDYDTH in *CONTROL_IMPLICIT_DYNAMICS and was recently (Rev. 81400) linked together with *CONTROL_IMPLICIT_FORMING. In a partial keyword example below, death time for the implicit dynamic is set at 0.55 second. The test model shown in [Figure 12-80](#) (left) results in a gravity loaded blank shape in [Figure 12-80](#) (right). Without the switch from dynamic to static, the blank at the end of the simulation will be as shown in [Figure 12-81](#). The result with switching is more reasonable. The energy history in [Figure 12-82](#) reveals that the kinetic energy dissipates completely at 0.60 second.

```
*CONTROL_TERMINATION
1.0
*CONTROL_IMPLICIT_FORMING
$  IOPTION      NSMIN      NSMAX      BIRTH      DEATH      PENCHK
    1
*CONTROL_IMPLICIT_DYNAMICS
$  IMASS      GAMMA      BETA      TDYBIR      TDYDTH      TDYBUR      IRATE
    1      0.600      0.380                  0.55
```

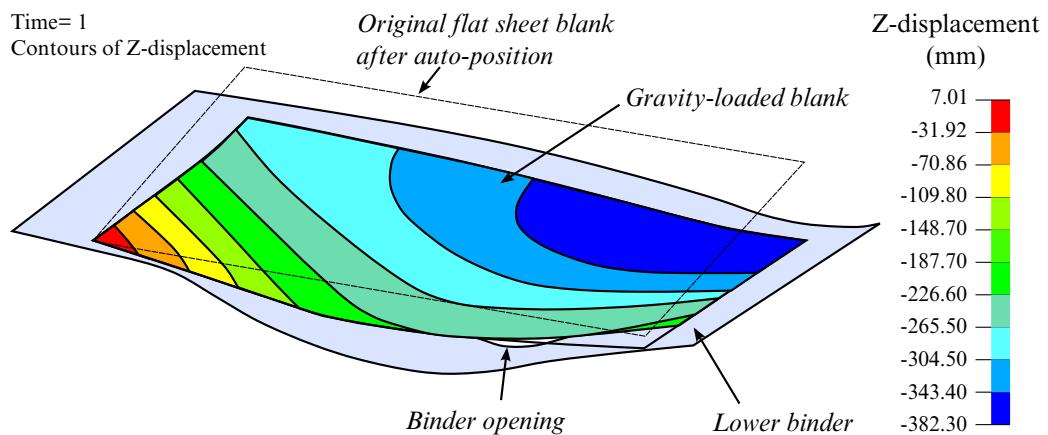


Figure 12-68. Gravity loading on a box side outer toggle die (courtesy of Auto-die, LLC).

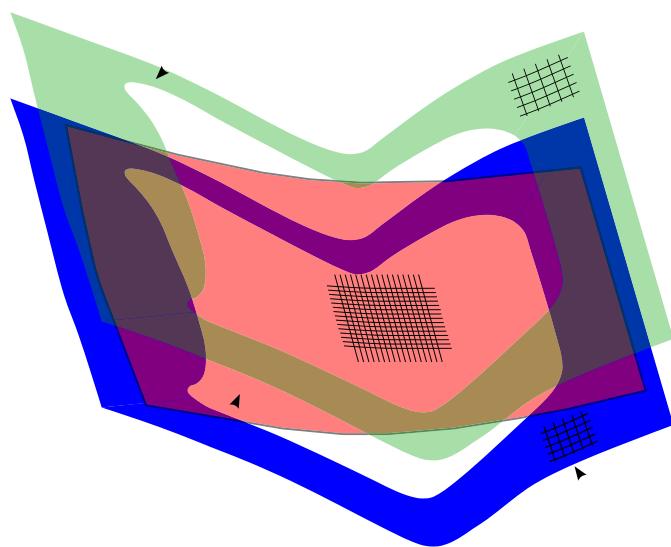


Figure 12-69. Initial auto-positioning (NUMISHEET2005 decklid inner).

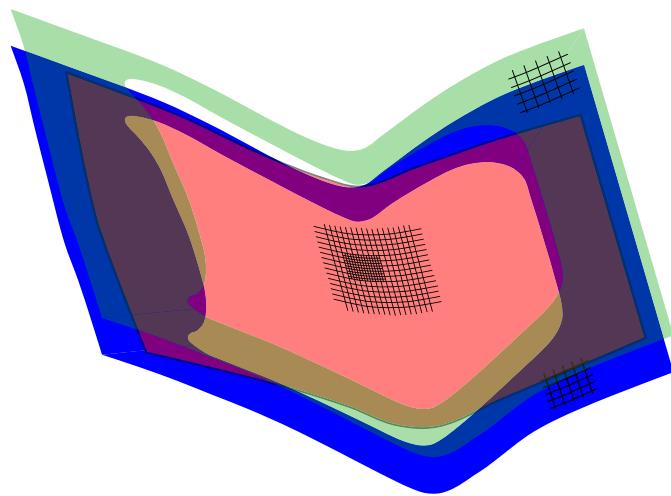


Figure 12-70. At 50% upper travel.

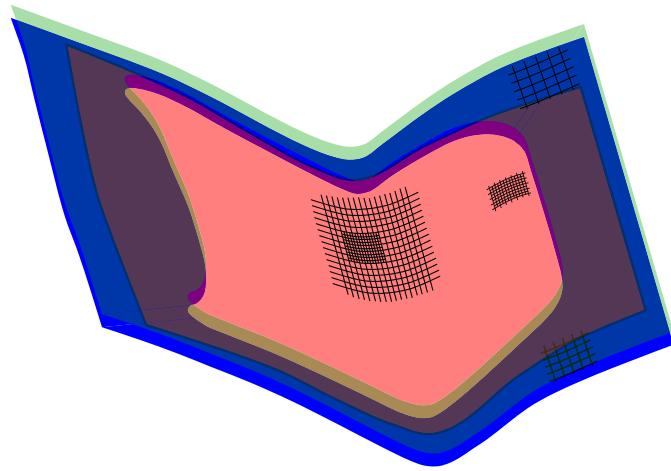


Figure 12-71. At 80% upper travel.

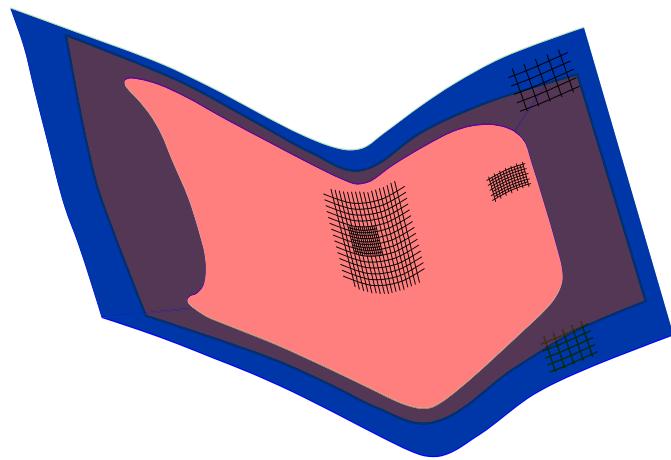


Figure 12-72. Upper travels to home.

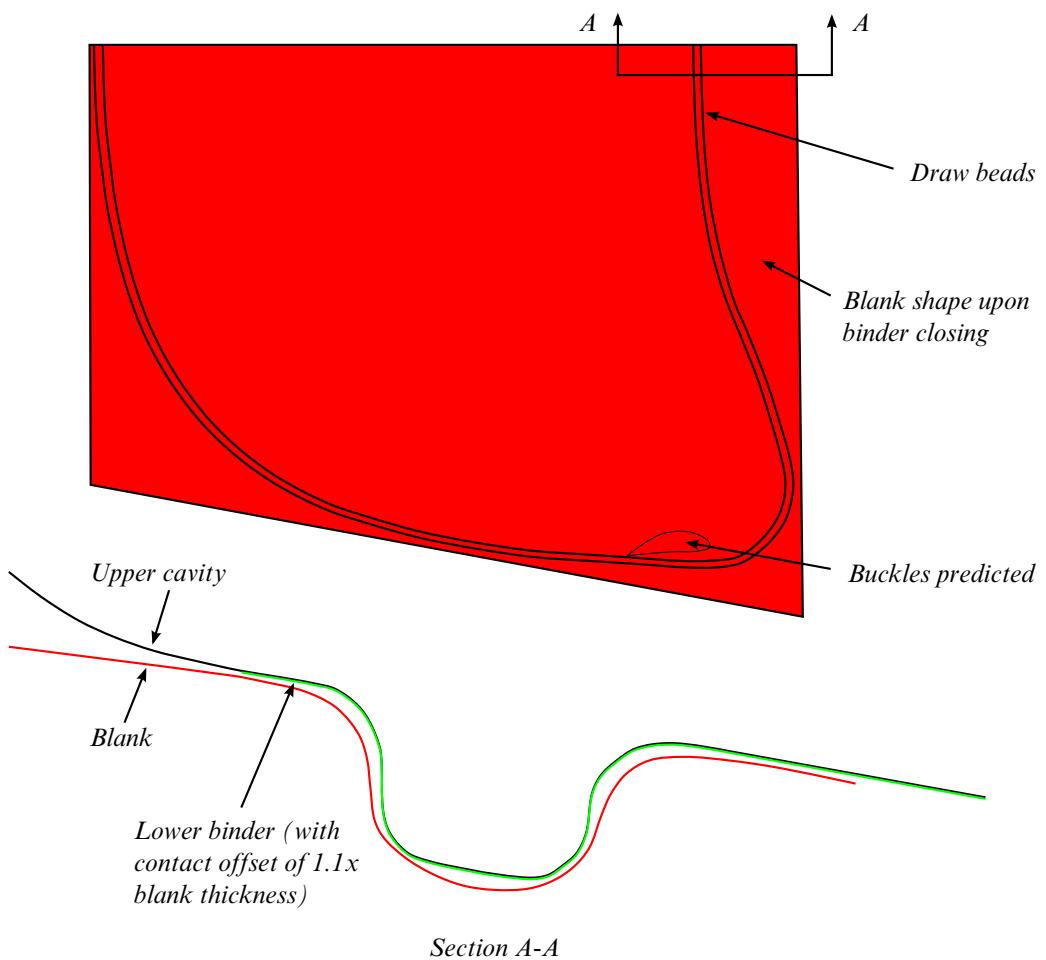


Figure 12-73. Binder closing with beads on a hood outer.

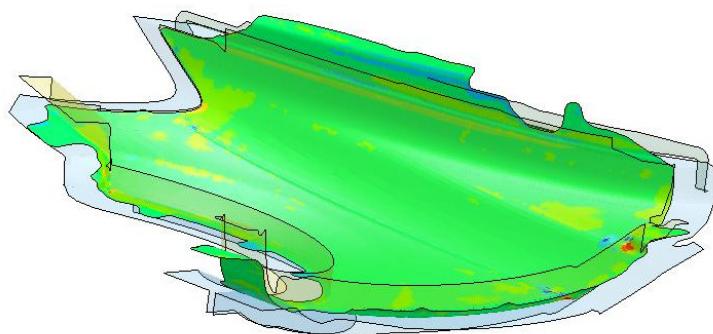


Figure 12-74. Mean stress at pad closing.

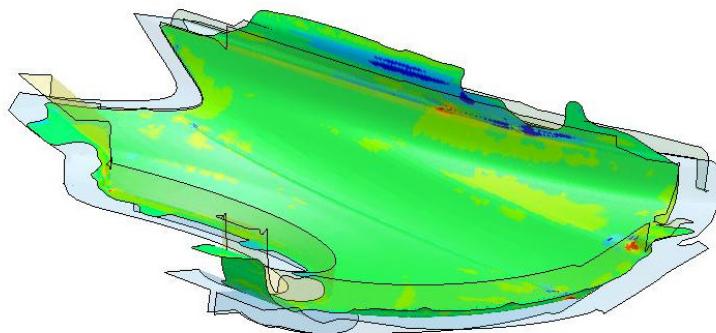


Figure 12-75. Mean stress at 40% Travel.

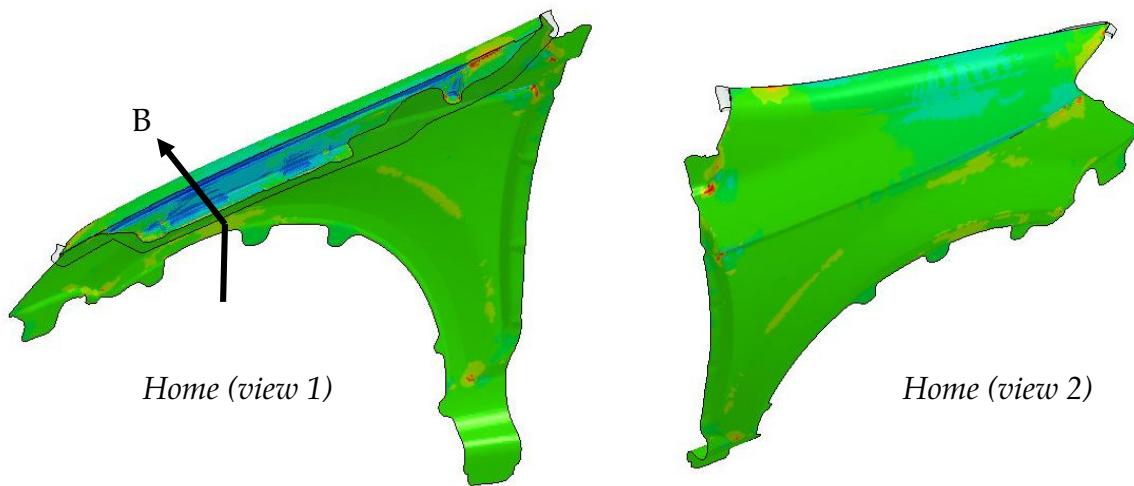


Figure 12-76. Mean stress at flanging home (compression/surface lows in red).

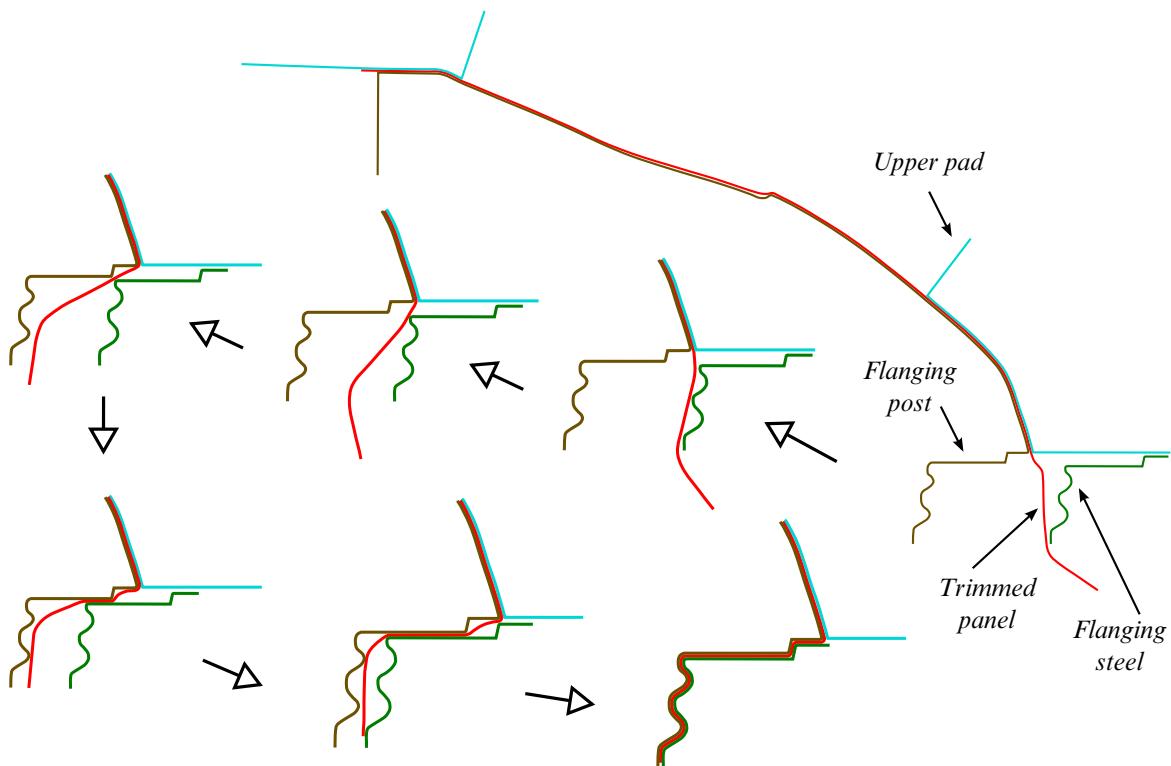


Figure 12-77. Flanging progression along section B (flanging post stationary).

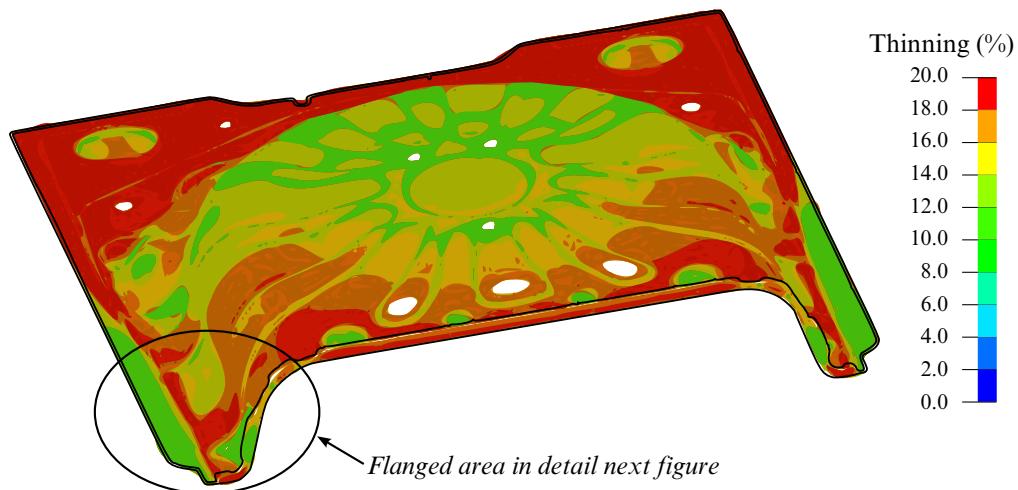


Figure 12-78. Flanging simulation of a rear floor pan using IOPTION 1 (Courtesy of Chrysler, LLC).

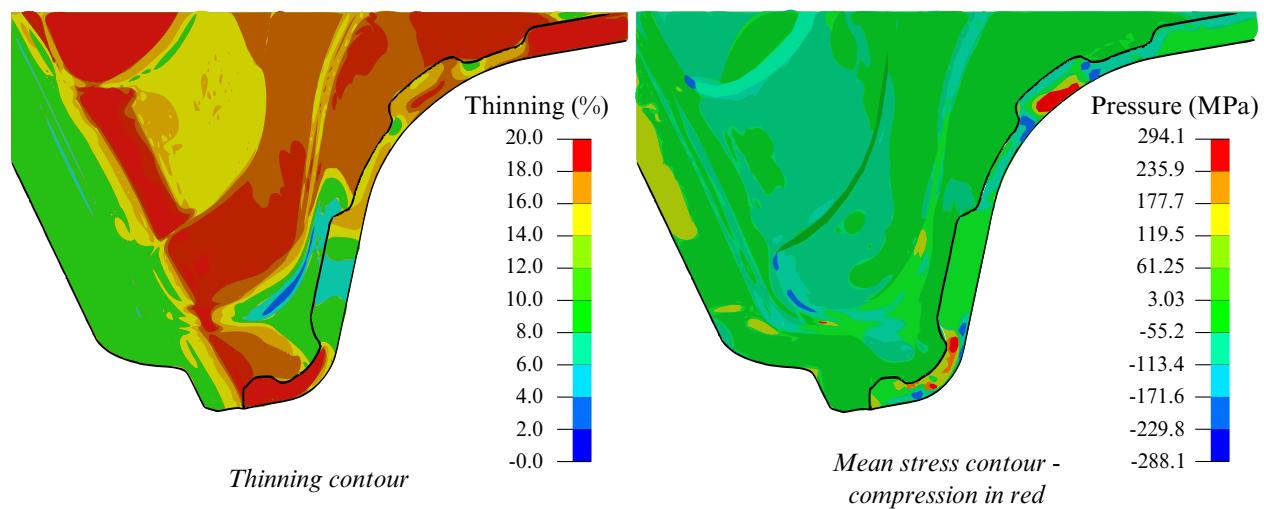


Figure 12-79. Localized view of the last figure.

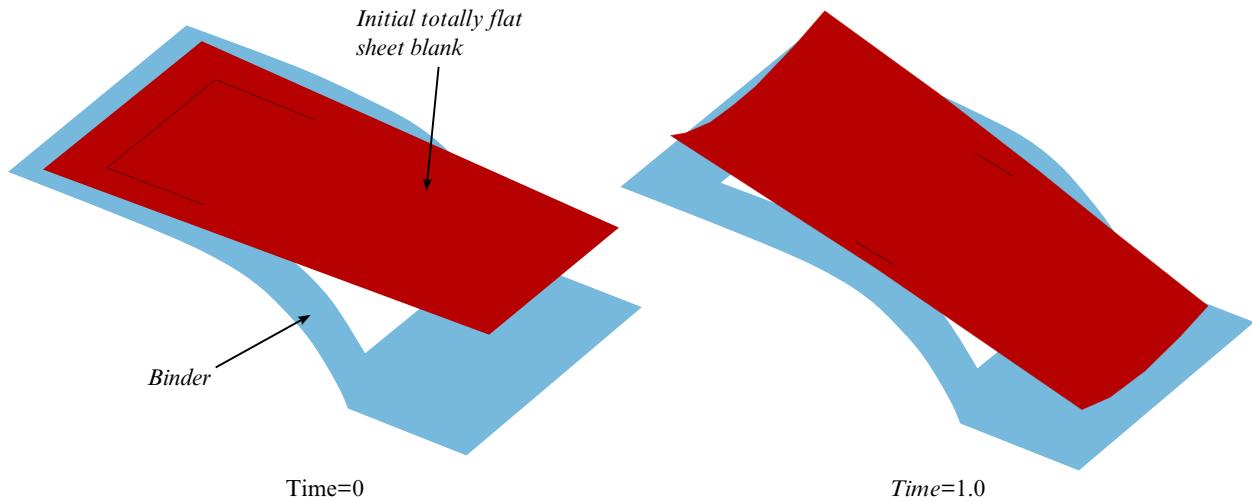


Figure 12-80. Test model (left) and gravity loaded blank (right) with switching from implicit dynamic to implicit static.

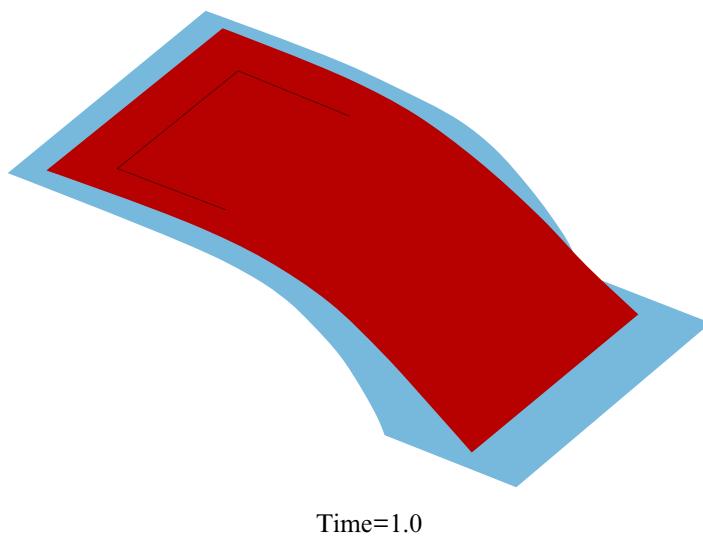


Figure 12-81. Gravity loaded blank without the “switching”.

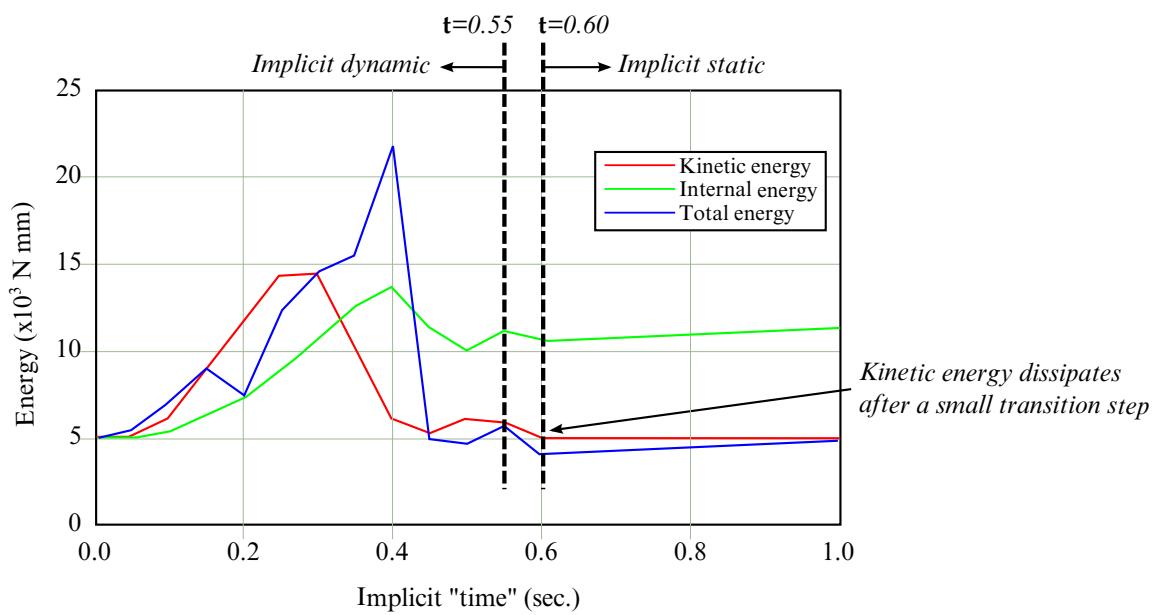


Figure 12-82. Switching between implicit dynamic and implicit static.

***CONTROL_IMPLICIT_GENERAL_{OPTION}**

Available options include:

<BLANK>

DYN

SPR

Purpose: Activate implicit analysis and define associated control parameters. This keyword is required for all implicit analyses. The DYN option enables setting controls specifically for the dynamic relaxation phase. The SPR option enables setting controls specifically for the springback phase.

Card 1	1	2	3	4	5	6	7	8
Variable	IMFLAG	DTO	IMFORM	NSBS	IGS	CNSTN	FORM	ZERO_V
Type	I	F	I	I	I	I	I	I
Default	0	none	2	1	2	0	0	0

VARIABLE	DESCRIPTION
IMFLAG	<p>Implicit/Explicit analysis type flag (see Remark 1):</p> <ul style="list-style-type: none"> EQ.0: explicit analysis EQ.1: implicit analysis EQ.2: explicit followed by implicit, (seamless springback). *INTERFACE_SPRINGBACK_SEAMLESS is required to activate seamless springback. EQ.4: implicit with automatic implicit-explicit switching EQ.5: implicit with automatic switching and mandatory implicit finish EQ.6: explicit with intermittent eigenvalue extraction LT.0: curve ID = -IMFLAG specifies IMFLAG as a function of time.

*CONTROL

*CONTROL_IMPLICIT_GENERAL

VARIABLE	DESCRIPTION
DT0	Initial time step size for implicit analysis. See Remarks 2 and 5 . LT.0: eliminate negative principal stresses in geometric (initial stress) stiffness. Initial time step is DT0 .
IMFORM	Element formulation flag for seamless springback; see *INTERFACE_SPRINGBACK_SEAMLESS. See Remark 3 . EQ.1: switch to fully integrated shell formulation for springback EQ.2: retain original element formulation (default)
NSBS	Number of implicit steps in seamless springback; see *INTERFACE_SPRINGBACK_SEAMLESS. See Remark 4 .
IGS	Geometric (initial stress) stiffness flag (see Remark 5): EQ.1: include EQ.2: ignore LT.0: include on part set IGS
CNSTN	Indicator for consistent tangent stiffness (solid materials 3 & 115 only): EQ.0: do not use (default) EQ.1: use
FORM	Fully integrated element formulation (IMFLAG = 2 and IMFORM = 1 only). See Remark 3 . EQ.0: type 16 EQ.1: type 6
ZERO_V	Zero out the velocity before switching from explicit to implicit. EQ.0: the velocities are not zeroed out. EQ.1: the velocities are set to zero.

Remarks:

1. **Analysis Types.** The default value 0 for IMFLAG indicates a standard explicit analysis will be performed. Using value 1 causes an entirely implicit analysis to be performed. Value 2 is automatically activated when the keyword *INTERFACE_SPRINGBACK_SEAMLESS is present, causing the analysis type to

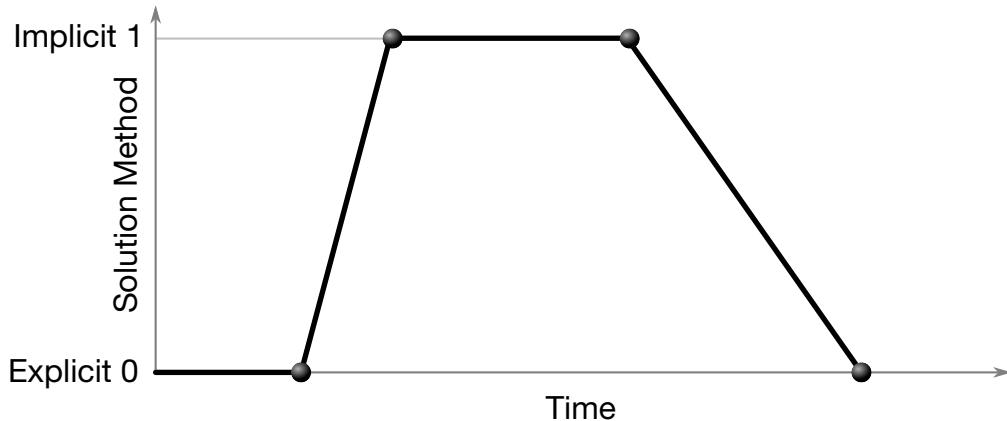


Figure 12-83. Solution method, implicit or explicit, controlled by a load curve.

switch from explicit to implicit when the termination time is reached. Other nonzero values for IMFLAG can also be used with *INTERFACE_SPRING-BACK_SEAMLESS. After this switch, the termination time is extended by NS-BS \times DT0 or reset to twice its original value if DT0 = 0.0. The implicit simulation then proceeds until the new termination time is reached. Contact interfaces are automatically disabled during the implicit phase of seamless springback analysis. Furthermore, implicit stabilization (*CONTROL_IMPLICIT_STABILIZATION) and automatic step size adjustment (*CONTROL_IMPLICIT_AUTO) are on by default for seamless springback.

When the automatic implicit-explicit switching option is activated (IMFLAG = 4 or 5), the solution method will begin as implicit. If convergence of the equilibrium iterations fails, the solution will automatically switch to explicit for a time interval of DTEXP (see *CONTROL_IMPLICIT_AUTO). After this time interval, the solution method will switch back to implicit and attempt to proceed. The implicit simulation may be either static or dynamic. When this feature is used in a static implicit job, simulation time is no longer arbitrary and must be chosen along with DTEXP in a realistic way to cause efficient execution of any explicit phases. Mass scaling may also be activated (see *CONTROL_TIMESTEP) and will apply only during the explicit phases of the calculation. In cases where much switching occurs, users must exercise caution to ensure that negligible dynamic effects are introduced by the explicit phases.

When IMFLAG = 5, the final step of the simulation must be implicit. The termination time will be extended automatically as necessary, until a successfully converged implicit step can be obtained. This is useful for example in difficult metal forming springback simulations.

When IMFLAG = 6, an explicit simulation will be performed. Eigenvalues will be extracted intermittently according to a curve indicated by curve -NEIG on *CONTROL_IMPLICIT_EIGENVALUE. Beware that dynamic stress oscillations which may occur in the explicit simulation will influence the geometric

(initial stress) stiffness terms used in the eigen solution, potentially producing misleading results and/or spurious modes. As an alternative, eigenvalues can also be extracted intermittently during an implicit analysis, using IMFLAG = 1 and curve -NEIG.

IMFLAG < 0 indicates a curve ID that gives the solution method as a function of time. Define a curve value of zero during explicit phases, and a value of one during implicit phases. Use steeply sloping sections between phases. An arbitrary number of formulation switches may be activated with this method. See [Figure 12-83](#).

2. **Implicit Phase Initial Time Step Size.** DT0 selects the initial time step size for the implicit phase of a simulation. The step size may change during a multiple step simulation if the automatic time step size control feature is active (see [*CONTROL_IMPLICIT_AUTO](#)).
3. **Element Formulation Switching.** An adaptive mesh must be activated when using element formulation switching. For best springback accuracy, use of shell type 16 is recommended during the entire stamping and springback analysis, in spite of the increased cost of using this element during the explicit stamping phase.
4. **Seamless Springback.** The NSBS option allows a seamless springback analysis, invoked with [*INTERFACE_SPRINGBACK_SEAMLESS](#), to use multiple unloading steps. Implicit seamless springback begins at time, $t = \text{ENDTIM}$ and finishes at $t = \text{ENDTIM} + \text{NSBS} \times \text{DT0}$ where ENDTIM is specified in [*CONTROL_TERMINATION](#) and DT0 is specified in [*CONTROL_IMPLICIT_GENERAL](#).
5. **Geometric Stiffness.** The geometric stiffness (IGS) adds the effect of initial stress to the global stiffness matrix. This effect is seen in a piano string whose natural frequency changes with tension. Geometric stiffness does not always improve nonlinear convergence, especially when compressive stresses are present, so its inclusion is optional. Furthermore, the geometric stiffness may lead to convergence problems with incompressible, or nearly incompressible, materials. To eliminate compressive stresses in the geometric stiffness, set DT0 to a negative value.

CONTROL_IMPLICIT_INERTIA_RELIEF**CONTROL*****CONTROL_IMPLICIT_INERTIA_RELIEF**

Purpose: Allows analysis of models that have rigid body modes by suppressing the rigid body motion. It can be used for implicit problems as well as implicit/explicit switching.

Card 1	1	2	3	4	5	6	7	8
Variable	IRFLAG	THRESH	IRCNT	MXMODES				
Type	I	F	I	I				
Default	0	0.1	0	100				

Additional Mode List Cards. This card should be included only when the user wants to specify the modes to use. Include as many cards as needed to provide all values. This input ends at the next keyword ("*") card. The mode numbers do not have to be consecutive.

Card 2	1	2	3	4	5	6	7	8
Variable	MODE1	MODE2	MODE3	MODE4	MODE5	MODE6	MODE7	MODE8
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
IRFLAG	Inertia relief flag EQ.0: do not perform inertia relief EQ.1: do perform inertia relief and use for both implicit and explicit EQ.2: do perform inertia relief but only use for implicit time steps
THRESH	Threshold for what is a rigid body mode. The default is set to 0.1. Thus, the default threshold is 0.1 Hz when using seconds for the unit of time. If the unit of time is not seconds, you may need to adjust this parameter appropriately. For instance, a unit of time of ms means that the default is 0.1 kHz which leads to too many modes.

***CONTROL**

***CONTROL_IMPLICIT_INERTIA_RELIEF**

VARIABLE	DESCRIPTION
IRCNT	The user can specify to use the lowest IRCNT modes instead of using THRESH to determine the number of modes.
MXMODES	Maximum number of allowed modes that can be stored in memory
MODE <i>i</i>	Ignore THRESH and IRCNT and use a specific list of modes, skipping those that should not be used.

***CONTROL_IMPLICIT_JOINTS**

Purpose: Specify penalty or constraint treatment of joints for implicit analysis.

Card 1	1	2	3	4	5	6	7	8
Variable	ISPHER	IREVOL	ICYLIN					
Type	I	I	I					
Default	1	1	1					

VARIABLE	DESCRIPTION
ISPHER	Treatment of spherical joints EQ.1: use constraint method for all spherical joints (default) EQ.2: use penalty method for all spherical joints
IREVOL	Treatment of revolute joints EQ.1: use constraint method for all revolute joints (default) EQ.2: use penalty method for all revolute joints
ICYLIN	Treatment of cylindrical joints EQ.1: use constraint method for all cylindrical joints (default) EQ.2: use penalty method for all cylindrical joints

Remarks:

For most implicit applications one should use the constraint (default) method for the treatment of joints. When explicit-implicit switching is used the joint treatment should be consistent. This keyword allows the user to choose the appropriate treatment for their application.

*CONTROL_IMPLICIT

*CONTROL_IMPLICIT_MODAL_DYNAMIC

*CONTROL_IMPLICIT_MODAL_DYNAMIC

Purpose: Activate implicit modal dynamic analysis. Eigenmodes are used to linearize the model by projecting the model onto the space defined by the eigenmodes. The eigenmodes can be computed or read from a file. All or some of the modes can be used in the linearization. Modal damping can be applied.

Card 1	1	2	3	4	5	6	7	8
Variable	MDFLAG	ZETA	MD_STRS	DTOUT	INTEG	NSID		
Type	I	F	I	F	I	I		
Default	0	0.0	0	0	1	{all}		

Eigenmodes Card. This card is optional if MDFLAG = 1, but it is required, even if blank, if MDFLAG > 1.

Card 2	1	2	3	4	5	6	7	8
Variable				FILENAME				
Type				A80				

Constraints Card. Include this card if MDFLAG = 2 or 4.

Card 3	1	2	3	4	5	6	7	8
Variable				FILENAME2				
Type				A80				

Residual Vector Card. Include this card if MDFLAG = 3 or 4.

Card 4	1	2	3	4	5	6	7	8
Variable				FILENAME3				
Type				A80				

VARIABLE	DESCRIPTION
MDFLAG	<p>Modal Dynamic flag:</p> <ul style="list-style-type: none"> EQ.0: No modal dynamic analysis EQ.1: Perform modal dynamic analysis. EQ.2: Perform modal dynamic analysis with prescribed motion constraints on the constraint modes input with Card 3. See Remark 7. EQ.3: Perform modal dynamic analysis with the addition of residual vectors to the eigenvectors. EQ.4: Combine analysis types 2 and 3.
ZETA	Modal Dynamic damping constant. See Remark 4 .
MD_STRS	<p>Modal dynamic stress flag:</p> <ul style="list-style-type: none"> EQ.0: No modal stress calculated NE.0: Compute modal stresses for output to d3plot using the modal stresses from the d3eigv database. See Remark 3.
DTOUT	<p>Modal dynamics output interval.</p> <ul style="list-style-type: none"> LE.0: No modal variable output GT.0: Output modal displacement, velocity, and acceleration to file moddynout every DTOUT of simulation time.
INTEG	<p>Integration method</p> <ul style="list-style-type: none"> EQ.1: perform modal dynamic analysis with explicit time integration (default_) EQ.2: perform modal dynamic analysis with implicit time integration. See Remark 6.
NSID	The node set ID of the nodes subjected to loads in the modal model. If the set is not specified, then the forces are summed over all the nodes, and that is usually much more expensive than summing over only those subjected to a load.
FILENAME	If specified, the eigenmodes are read from the specified file. Otherwise, the eigenmodes are computed as specified on *CONTROL_IMPLICIT_EIGENVALUE . See Remark 2 .

*CONTROL_IMPLICIT

*CONTROL_IMPLICIT_MODAL_DYNAMIC

VARIABLE	DESCRIPTION
FILENAME2	If specified, constraint modes are read from the specified file. Prescribed motion constraints can then be applied to the constraint modes. See Remark 7 .
FILENAME3	If specified, the residual vectors are read from the specified file. Residual vectors can be used in conjunction with eigenmodes for the analysis. See Remark 8 .

Remarks:

1. **Modal dynamics.** Modal Dynamic uses the space spanned by the eigenmodes of the generalized eigenvalue problem

$$\mathbf{K}\Phi_i = \lambda_i \mathbf{M}\Phi_i.$$

The matrix of eigenmodes, Φ , diagonalizes \mathbf{K} and \mathbf{M}

$$\Phi^T \mathbf{K} \Phi = \Lambda$$

and

$$\Phi^T \mathbf{M} \Phi = \mathbf{I}.$$

Multiplication by Φ changes coordinates from amplitude space to displacement space as

$$\mathbf{u} = \Phi \mathbf{a},$$

where \mathbf{a} is a vector of modal amplitudes. The equations of motion,

$$\mathbf{M}\ddot{\mathbf{u}}^{n+1} + \mathbf{K}\Delta\mathbf{u} = \mathbf{F}(\mathbf{x}^n),$$

when multiplied on the left by Φ^T and substituting $\mathbf{u} = \Phi \mathbf{a}$ become the linearized equations of motion in its spectral form as

$$\mathbf{I}\ddot{\mathbf{a}}^{n+1} + \Lambda(\Delta\mathbf{a}) = \Phi^T \mathbf{F}(\mathbf{x}^n).$$

The modal damping feature adds a velocity-dependent damping term,

$$\mathbf{I}\ddot{\mathbf{a}}^{n+1} + 2Z\dot{\mathbf{a}}^n + \Lambda(\Delta\mathbf{a}) = \Phi^T \mathbf{F}(\mathbf{x}^n),$$

where $Z_{ii} = \zeta_i \omega_i$, $\omega_i = \sqrt{\lambda_i}$, and each ζ_i is a user-specified damping coefficient.

The matrices in the reduced equations are diagonal and constant. Thus, Modal Dynamics can quickly compute the acceleration of the amplitudes and, hence, the motion of the model. The space spanned by the eigenmodes, however, restricts the motion.

2. **Eigenmodes.** Eigenmodes are either computed based on [*CONTROL_IMPLICIT_EIGENVALUE](#) or read from file FILENAME. By default, all modes are used

in the projection. Selected modes can be specified via [*CONTROL_IMPLICIT_MODAL_DYNAMIC_MODE](#) to reduce the size of the projection.

3. **Stress output.** Stresses are computed only for linear shell formulation 18, linear solid formulation 18, and linear beam formulation 13.
4. **Modal damping.** Modal damping on all modes can be specified using ZETA. More options for specifying modal damping can be found on [*CONTROL_IMPLICIT_MODAL_DYNAMIC_DAMPING](#).
5. **Similar keywords.** Using MDFLAG = 1, ZETA = 0.0, and FILENAME = " " is the same as using IMASS = 2 with [*CONTROL_IMPLICIT_DYNAMICS](#). Using MDFLAG = 1, ZETA = 0.0 and FILENAME = d3eigv is the same as IMASS = 3. The new keywords [*CONTROL_IMPLICIT_MODAL_DYNAMIC_MODE](#) and [*CONTROL_IMPLICIT_MODAL_DYNAMIC_DAMPING](#) provide additional options for mode selection and modal damping.
6. **Time integration method.** When INTEG is set to 2, the implicit Newmark time integration method is used instead of the explicit central difference method, and the time step size is taken from the [*CONTROL_IMPLICIT_GENERAL](#) input. The integration parameters are taken from the [*CONTROL_IMPLICIT_DYNAMICS](#) input. If they are not specified, the default values of GAMMA and BETA are used. This option permits a time step size limited only by the accuracy requirements of the user.
7. **Prescribed Motion Constraints.** To apply prescribed motion constraints in global coordinates at specified nodes, first use [*CONTROL_IMPLICIT_MODES](#) to compute constraint modes for these nodes for this model. The resulting d3mode file is then input as FILENAME2. The computation of the constraint modes must be performed separately from the computation of the eigenmodes specified in file FILENAME. The prescribed motion constraint ([*BOUNDARY_PRESCRIBED_MOTION](#)) should only be present in the execution with [*CONTROL_IMPLICIT_MODAL_DYNAMIC](#) and should not be present during the computation of the eigenmodes or the constraint modes. Any prescribed motion constraints on any node *not* in the constraint mode set will be ignored.
8. **Residual vectors.** Residual vectors can be used in conjunction with eigenmodes in the analysis to better represent specific loading conditions.

***CONTROL_IMPLICIT**

***CONTROL_IMPLICIT_MODAL_DYNAMIC_DAMPING**

***CONTROL_IMPLICIT_MODAL_DYNAMIC_DAMPING_{OPTION}**

Available options include:

<BLANK>

SPECIFIC

FREQUENCY_RANGE

Purpose: Define damping terms to be used in implicit modal dynamics.

Card Summary:

Card 1a. Include this card if and only if the keyword option is unused, that is, <BLANK>.

ZETA1								
-------	--	--	--	--	--	--	--	--

Card 1b. Include this card if and only if the SPECIFIC keyword option is used. This input ends at the next keyword ("*") card.

MID1	ZETA1	MID2	ZETA2	MID3	ZETA3	MID4	ZETA4
------	-------	------	-------	------	-------	------	-------

Card 1c. Include this card if and only if the FREQUENCY_RANGE keyword option is used. This input ends at the next keyword ("*") card.

FREQ1	ZETA1	FREQ2	ZETA2	FREQ3	ZETA3	FREQ4	ZETA4
-------	-------	-------	-------	-------	-------	-------	-------

Data Card Definitions:

Damping Card. Card for option set to <BLANK>.

Card 1a	1	2	3	4	5	6	7	8
Variable	ZETA1							
Type	F				I			

Specific Damping Cards. Card for the SPECIFIC option. This input ends at the next keyword ("**") card.

Card 1b	1	2	3	4	5	6	7	8
Variable	MID1	ZETA1	MID2	ZETA2	MID3	ZETA3	MID4	ZETA4
Type	I	F	I	F	I	F	I	F

Frequency Range Damping Cards. Card for FREQUENCY_RANGE option. This input ends at the next keyword ("**") card.

Card 1c	1	2	3	4	5	6	7	8
Variable	FREQ1	ZETA1	FREQ2	ZETA2	FREQ3	ZETA3	FREQ4	ZETA4
Type	F	F	F	F	F	F	F	F

VARIABLE	DESCRIPTION
ZETAn	Modal Dynamic damping coefficient n .
MIDn	Mode ID n .
FREQn	Frequency value n .

Remarks:

- No Keyword Option.** If no option is specified, the value of ZETA1 becomes the damping coefficient for all modes involved in implicit modal dynamic analysis. This value overrides the value on *CONTROL_IMPLICIT_MODAL_DYNAMIC.
- SPECIFIC Keyword Option.** If option SPECIFIC is specified, the integers MIDn indicate which modes involved in *CONTROL_IMPLICIT_MODAL_DYNAMIC will have modal damping applied to them. The associated value ZETAn will be the modal damping coefficient for that mode.
- FREQUENCY_RANGE Keyword Option.** If option FREQUENCY_RANGE is specified, all modes involved will have modal damping applied. The damping coefficient will be computed by linear interpolation of the pairs (FREQi,ZETAi). If the modal frequency is less than FREQ1, then the modal damping coefficient will be ZETA1. If the modal frequency is greater than FREQn, then the modal

*CONTROL_IMPLICIT

*CONTROL_IMPLICIT_MODAL_DYNAMIC_DAMPING

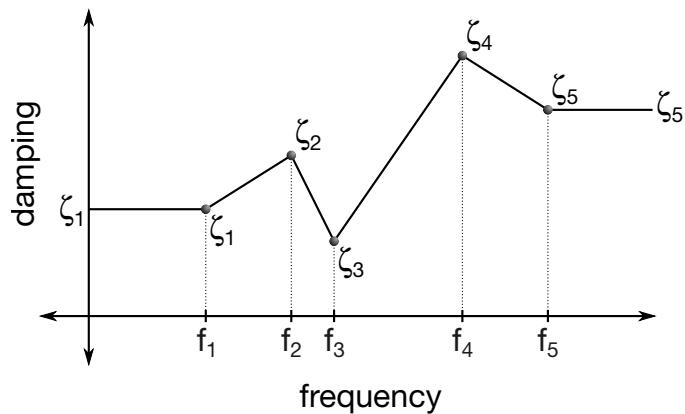


Figure 12-84. Schematic illustration of frequency range damping.

damping coefficient will be ZETAn. The values of FREQi must be specified in ascending order.

***CONTROL_IMPLICIT_MODAL_DYNAMIC_MODE_OPTION**

Available options include:

LIST

GENERATE

Purpose: Define vibration modes to be used in implicit modal dynamic.

Mode ID Cards. Card 1 for the LIST keyword option. Set one value per mode. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1a	1	2	3	4	5	6	7	8
Variable	MID1	MID2	MID3	MID4	MID5	MID6	MID7	MID8
Type								

Mode Range Cards. Card 1 for the GENERATE keyword option. Set one pair of MnBEG and MnEND values per block of modes. Include as many cards as needed. This input ends at the next keyword ("*") card.

Card 1b	1	2	3	4	5	6	7	8
Variable	M1BEG	M1END	M2BEG	M2END	M3BEG	M3END	M4BEG	M4END
Type								

VARIABLE	DESCRIPTION
MID n	Mode ID n .
MnBEG	First mode ID in block n .
MnEND	Last mode ID in block n . All mode IDs between and including MnBEG and MnEND are added to the list.

Remarks:

- Related Keywords.** This keyword may be used with *CONTROL_IMPLICIT_MODAL_DYNAMIC if some of the vibration modes have less contribution to

***CONTROL_IMPLICIT**

***CONTROL_IMPLICIT_MODAL_DYNAMIC_MODE**

the total structural response and can be removed from the implicit modal dynamic analysis.

CONTROL_IMPLICIT_MODES**CONTROL*****CONTROL_IMPLICIT_MODES {OPTION}**

Available options include:

<BLANK>

BINARY

Purpose: Request calculation of constraint, attachment, and /or eigenmodes for later use in modal analysis using *PART_MODES (see also *CONTROL_IMPLICIT_GENERAL) or *ELEMENT_DIRECT_MATRIX_INPUT.

Card Summary:

Card 1. This card is required.

NSIDC	NSIDA	NEIG	IBASE	SE_MASS	SE_DAMP	SE_STIFF	SE_INERT
-------	-------	------	-------	---------	---------	----------	----------

Card 1.1. Include this card if at least one of SE_MASS, SE_DAMP, SE_STIFF, or SE_INERT is not blank.

SE_FILENAME

Card 2. This card is optional.

IRESVEC	ISTRESS	ID3MODE					
---------	---------	---------	--	--	--	--	--

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	NSIDC	NSIDA	NEIG	IBASE	SE_MASS	SE_DAMP	SE_STIFF	SE_INERT
Type	I	I	I	I	A	A	A	A
Default	{Ø}	{Ø}	0	0	blank	blank	blank	blank

CONTROL**CONTROL_IMPLICIT_MODES**

Card 1.1	1	2	3	4	5	6	7	8
Variable	SE_FILENAME							
Type	A							

Card 2	1	2	3	4	5	6	7	8
Variable	IRESVEC	ISTRESS	ID3MODE					
Type	I	I	I					
Default	0	0	0					

VARIABLE	DESCRIPTION
NSIDC	Node set ID for constraint modes. See Remark 2 . EQ.0: No constraint modes will be generated
NSIDA	Node set ID for attachment modes. See Remark 2 . EQ.0: No attachment modes will be generated
NEIG	Number of eigenmodes (normal modes). See Remark 2 . EQ.0: No eigenmodes will be generated.
IBASE	Offset for numbering of the generalized internal degrees of freedom for the superelement
SE_MASS	Name of the superelement mass matrix. If left blank, it is not generated.
SE_DAMP	Name of the superelement damping matrix. If left blank, it is not generated.
SE_STIFF	Name of the superelement stiffness matrix. If left blank, it is not generated.
SE_INERT	Name of the superelement inertia matrix, required for gravity loading applications of the superelement. If left blank, it is not generated.

VARIABLE	DESCRIPTION
SE_FILENAME	If any of SE_MASS, SE_DAMP, SE_STIFF, or SE_INERT are not blank, then the second line is required and contains the filename for the superelement. See Remark 3 .
IRESVEC	Converting the attachment modes to residual vectors flag (see Remark 4): EQ.0: No conversion EQ.1: Conversion by orthogonalizing them to the eigenvectors.
ISTRESS	Flag to compute stresses: EQ.0: No stresses are computed. EQ.1: Stresses are computed for the mode shapes and added to the d3mode database.
ID3MODE	Write d3mode file flag: EQ.0: Write the d3mode file. EQ.1: Do not write the d3mode file. This is useful when *CONTROL_IMPLICIT_MODES is being used to create superelement matrices and you do not care about the contents of the d3mode file.

Remarks:

- Input and output files.** To use this feature, an implicit analysis must be requested using IMFLAG = 1 on *CONTROL_IMPLICIT_GENERAL, and a non-zero termination time must be specified on *CONTROL_TERMINATION. A double precision version of LS-DYNA should be used for best accuracy. Care must be taken to apply a sufficient number of constraints to the model to eliminate static rigid body motion. Computed modes are written to binary output file d3mode, with the order of output being constraint modes, followed by attachment modes, and then eigenmodes. The modes can be viewed by reading d3mode into LS-PrePost. Eigenmodes are also written to binary output file d3eigv.
- Modes.** Constraint and attachment modes are generated by applying unit displacements and unit forces, respectively, to each specified degree of freedom. By default, modes are computed for all degrees of freedom for each node in sets NSIDC and NSIDA. The first and second node set attribute parameters can be

optionally used to restrict the translational and rotational degrees of freedom for which modes are requested, respectively, according to the following syntax:

- a) Node set attribute parameters DA1 and A1: translational degree of freedom codes
- b) Node set attribute parameters DA2 and A2: rotational degree of freedom codes

<u>code</u>	<u>modes computed</u>
0	(See note below.)
1	X degree of freedom only
2	Y degree of freedom only
3	Z degree of freedom only
4	X, Y degrees of freedom only
5	Y, Z degrees of freedom only
6	X, Z degrees of freedom only
7	X, Y, Z degrees of freedom

Setting both node set attributes to zero is equivalent to setting both node set attributes to 7 (X, Y, and Z for translational and rotational degrees of freedom).

If one node set attribute is nonzero (codes 1 to 7) and the other node set attribute is zero, then the zero attribute means NO degrees of freedom are considered. For example, if DA1 = 2 and DA2 = 0, then only the Y-translational degree of freedom modes are calculated.

Eigenmodes are generated for the model with single point constraints applied on the constraint modes. The number of eigenmodes is specified here. If the user wants to compute eigenmodes other than the lowest ones, the controls on *CONTROL_IMPLICIT_EIGENVALUE can be used.

The combination of constraint modes and eigenmodes form the Hurty-Craig-Bampton linearization for a model. Using only constraint modes is the same as static condensation.

Some broad guidelines for appropriate selection of constraint modes, attachment modes, and eigenmodes include:

- c) Use constraint modes for the nodal degrees-of-freedom that are to be "constrained" with SPCs or prescribed motion.
- d) Use attachment modes for nodal degrees-of-freedom that are under the influence of point loads.

- e) Use eigenmodes in the construction of the superelement to capture the reaction of the part being modeled by the superelement and the associated feedback to the rest of the model.

Note: This node set should only contain deformable nodes, using a node that is part of a rigid body will result in errors.

3. **Superelements.** When the superelement is created, an internal numbering must be applied to the attachment and eigenmodes. This numbering starts at IBASE+1.

The user can create the superelement representation of the reduced model by specifying the SE_MASS, SE_DAMP, SE_STIFF, SE_INERT and SE_FILENAME fields. The inertia matrix is necessary if body forces, such as gravity loads, are applied to the superelement. The file, by default, is written in the Nastran DMIG file format and can be used as input to *ELEMENT_DIRECT_MATRIX_INPUT. The BINARY keyword option can be used to create a binary representation for the superelement which can be used with *ELEMENT_DIRECT_MATRIX_INPUT_BINARY to reduce the file size.

4. **Residual vectors.** For some applications residual vectors may be desired. These are attachment modes (response of the model to a unit force) orthogonalized to the eigenmodes. To do this, set IRESVEC to 1.

*CONTROL_IMPLICIT

*CONTROL_IMPLICIT_ORDERING

*CONTROL_IMPLICIT_ORDERING

Purpose: Provide user control for ordering algorithms used by implicit linear algebra.

Card 1	1	2	3	4	5	6	7	8
Variable	ORDER	NMETIS	REUSE					
Type	I	I	I					
Default	0	0	0					

VARIABLE	DESCRIPTION
ORDER	Ordering option (see Remarks 1 and 2): EQ.0: Method set automatically by LS-DYNA EQ.1: MMD (Multiple Minimum Degree) EQ.2: METIS EQ.3: ParMETIS EQ.4: LS-GPart
NMETIS	Number of times to use METIS on each compute node for MPP. See Remark 2 .
REUSE	Control the ordering reuse feature (MPP only). See Remark 3 .

Remarks:

1. **Ordering option.** The system of linear equations is reordered to optimize the sparsity of the factorization when using direct methods. Reordering is a hard problem, and there is no one best algorithm for all problems. LS-DYNA offers four different methods:
 - a) *MMD (Multiple Minimum Degree)*. This reordering method is inexpensive but should only be used for small problems.
 - b) *METIS*. METIS is from the University of Minnesota. It is the default method for most problems. It is very effective for large problems. METIS, however, is a serial algorithm. For large problems in MPP, however, a bottleneck in both time and memory will occur. Thus, for a large number of

MPP processes, you might want to use one of the two parallel algorithms mentioned next.

- c) *ParMETIS*. ParMETIS is also from the University of Minnesota. It is the MPP version of METIS. For a large number of processes, it should run much faster than METIS, but there might be some variation in the factorization cost (storage and number of operations). It is available as of R14.
- d) *LS-GPart*. LS-GPart is an MPP algorithm developed by Ansys and available as of R11. For a large number of processes, it should run faster than METIS. Quality (in terms of factorization cost) compared to METIS or Par-METIS will depend on the problem.

Reordering cost is included in the symbolic factorization phase of the linear solver. For large models, if this cost exceeds 20% of the numeric factorization cost, it may be more efficient to select the MMD method.

Note that the value of ORDER also affects the eigensolution software. That is ORDER from this keyword card is applicable to eigensolution.

2. **METIS in MPP.** In MPP LS-DYNA will run METIS on each compute node based on the estimate of required storage for METIS and the amount of free storage. The estimate is usually an overestimate. If insufficient storage is available, then MMD is used instead of METIS, usually leading to a more expensive solution. You can force the use of METIS with NMETIS > 0. However, a memory error and fatal termination of LS-DYNA may result in this case. If you do choose to try this option, NMETIS = 1 is recommended.
3. **Ordering reuse.** In MPP the cost of reordering for the direct factorization can be a substantial proportion of the overall cost of a linear algebra solution. LS-DYNA attempts to reuse the ordering from the previous factorization as the default. If the matrix structure changes due to contact or another dynamic feature, a reordering is performed. Otherwise, the previous ordering is reused. REUSE < 0 causes no attempt to reuse the ordering to be made. REUSE > 0 invokes attempting to predict matrix changes due to contact so that an ordering might be able to be reused for future factorization. Except for special problems, it is best to use REUSE = 0, the default. This is an MPP-only feature as the cost of ordering for SMP execution is not a significant portion of the overall cost.

*CONTROL

*CONTROL_IMPLICIT_RESIDUAL_VECTOR

*CONTROL_IMPLICIT_RESIDUAL_VECTOR

Purpose: Activate and control the computations of residual vectors. Residual vectors are the linear response of a model to a specified load which is then orthogonalized to a set of eigenmodes and any previously computed residual vectors. The eigenmodes can be computed during the execution or read from an input file. The computation of residual vectors is the same as multi-step linear (see NSOLVR = -1 on *CONTROL_IMPLICIT_SOLUTION) but has the additional step of orthogonalization.

Card Summary:

Card 1a. Include this card if NEIG > 0.

IRESVEC	NEIG	IFORMAT					
---------	------	---------	--	--	--	--	--

Card 1a.1. Include this card if IFORMAT ≠ 0.

RV_FILENAME

Card 1b. Include this card if NEIG = 0.

IRESVEC	NEIG	IFORMAT					
---------	------	---------	--	--	--	--	--

Card 1b.1. Include this card if IRESVEC ≠ 0.

RV_FILENAME

Data Card Definitions:

Compute Eigenmodes Card. This card is included if and only if NEIG > 0.

Card 1a	1	2	3	4	5	6	7	8
Variable	IRESVEC	NEIG	IFORMAT					
Type	I	I	I					
Default	0	0	0					

VARIABLE

DESCRIPTION

IRESVEC

Residual vector control flag:

EQ.0: Do not compute residual vectors.

CONTROL_IMPLICIT_RESIDUAL_VECTOR**CONTROL**

VARIABLE	DESCRIPTION
	GT.0: Compute residual vectors.
NEIG	Number of eigenmodes to compute to orthogonalize the computed load (NEIG > 0)
IFORMAT	Format for processing eigenmodes (NEIG > 0): EQ.0: Do not dump the computed eigenmodes. LT.0: Dump the computed eigenmodes in binary format. GT.0: Dump the computed eigenmodes in ASCII format.

Eigenmodes File Card. This card is included if and only if IFORMAT ≠ 0.

Card 1a.1	1	2	3	4	5	6	7	8
Variable	RV_FILENAME							
Type	A80							

VARIABLE	DESCRIPTION
RV_FILENAME	Name of the file for dumping the eigenvector

Read Eigenmodes Card. This card is included if and only if NEIG = 0.

Card 1b	1	2	3	4	5	6	7	8
Variable	IRESVEC	NEIG	IFORMAT					
Type	I	I	I					
Default	0	0	0					

VARIABLE	DESCRIPTION
IRESVEC	Residual vector control flag: EQ.0: Do not compute residual vectors. GT.0: Compute residual vectors.

*CONTROL

*CONTROL_IMPLICIT_RESIDUAL_VECTOR

VARIABLE	DESCRIPTION
NEIG	Read the eigenmodes from the file RV_FILENAME (if IRESVEC ≠ 0), which is the file used for dumping eigenvectors; see EVDUMP on *CONTROL_IMPLICIT_EIGENVALUE.
IFORMAT	Format for processing eigenmodes read from RV_FILENAME when NEIG = 0 (if IRESVEC > 0): LT.0: File is in binary format. GT.0: File is in ASCII format. If IRSEVEC > 0 and NEIG = 0, IFORMAT = 0 is not allowed.

Eigenmodes File Card. This card is included if and only if IRESVEC ≠ 0.

Card 1b.1	1	2	3	4	5	6	7	8
Variable	RV_FILENAME							
Type	A80							

VARIABLE	DESCRIPTION
RV_FILENAME	Name of the file read to obtain the eigenvectors. See EVDUMP on *CONTROL_IMPLICIT_EIGENVALUE.

Remarks:

For each load step, LS-DYNA resets the geometry to the original geometry and computes the response for the given load. This feature is similar to multi-step linear (NSOLVR = -1 for *CONTROL_IMPLICIT_SOLUTION). Then, the algorithm orthogonalizes the response with respect to the eigenmodes using the mass matrix norm and with respect to any previously computed residual vectors. Finally, LS-DYNA normalizes the residual vector with respect to the mass matrix.

LS-DYNA reads the eigenmodes from an eigenvector dump file or computes them depending on the values of NEIG. RV_FILENAME gives the file name.

LS-DYNA creates an LSDA database named residual_vectors to hold the results of these computations. For each load step, the database includes the following:

- the specified load
- the residual vector, v

- the residual frequency ($v^T K v / v^T M v$)

Please contact Technical Support for instructions for reading this database.

In addition, LS-DYNA writes a binary database d3resvec, which LS-PrePost can read to animate residual vectors.

*CONTROL

*CONTROL_IMPLICIT_ROTATIONAL_DYNAMICS

*CONTROL_IMPLICIT_ROTATIONAL_DYNAMICS

Purpose: This keyword is used to model rotational dynamics using the implicit time integrator. Applications for this feature include the transient and vibration analysis of rotating parts, such as turbine blades, propellers in aircraft, and rotating disks in hard disk drives. This feature requires the use of a double precision executable. For transient simulations ($NOMEG = 0$) both SMP and MPP are supported. The MPP implementation of the eigenvalue computations ($NOMEG > 0$) is not yet available.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	STYPE	OMEGA	VID	NOMEG	IREF	OMEGADR	ISTFNS
Type	I	I	F	I	I	I	F	I
Default	none	0	none	none	0	0	0.0	3

Additional Rotational Speed Cards. This card should be included only when $NOMEG > 0$. Include as many cards as needed to provide all NOMEG values. This input ends at the next keyword ("**") card.

Card 2	1	2	3	4	5	6	7	8
Variable	OMEG1	OMEG2	OMEG3	OMEG4	OMEG5	OMEG6	OMEG7	OMEG8
Type	F	F	F	F	F	F	F	F

VARIABLE	DESCRIPTION
SID	Set ID of the rotational components
STYPE	Set type: EQ.0: Part EQ.1: Part set
OMEGA	Rotating speed in units of radians/(time unit). GT.0: Rotating speed LT.0: Curve ID = OMEGA gives rotating speed as a function of time.

VARIABLE	DESCRIPTION
VID	Vector ID to define the rotating axis. It can be defined by *DEFINE_VECTOR and *DEFINE_VECTOR_NODES. The tail of the vector should be set as the rotating center.
NOMEG	Number of rotating speeds. This feature is intended to automatically perform parameter studies with respect to the rotation speed. The keyword *CONTROL_IMPLICIT_EIGENVALUE must be included if NOMEG > 0.
IREF	Reference frame (see Remark 7): <ul style="list-style-type: none"> EQ.0: Rotating coordinate system. Rotating parts will not rotate in the visualization. Solid element and thick shell element will use IREF = 0. EQ.1: Fixed coordinate system. Using rotational dynamics will add Coriolis and spin softening terms to the matrices but will not rotate the structures. EQ.2: Rotating coordinate system, only rotate rotating parts for visualization purpose.
OMEGADR	Rotating speed defined in dynamic relaxation. <ul style="list-style-type: none"> GT.0: Rotating speed defined in dynamic relaxation. LT.0: Curve ID = OMEGADR gives rotating speed as a function of time.
ISTFNS	Flag for adding spin softening and gyroscopic stiffness terms: <ul style="list-style-type: none"> EQ.1: Add no stiffness for rotational dynamics effects. EQ.2: Add only the spin softening stiffness terms which keep the stiffness matrix symmetric. EQ.3: Add both the spin softening and gyroscopic stiffness terms which make the stiffness matrix nonsymmetric.
OMEG n	The n th rotating speed

Remarks:

1. **Equilibrium Equation.** The equilibrium equations depend on the reference frame. For the rotating coordinate system, the linearized equilibrium equation is given by

$$\mathbf{M}\ddot{\mathbf{u}} + (\mathbf{D} + 2\Omega\mathbf{C})\dot{\mathbf{u}} + (\mathbf{K} - \Omega^2\mathbf{K}_G)\mathbf{u} = \mathbf{F}.$$

Whereas, in a fixed coordinate system, the linearized equilibrium equation is

$$\mathbf{M}\ddot{\mathbf{u}} + (\mathbf{D} + \Omega\mathbf{C})\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{F}$$

with

- \mathbf{M} = lumped mass matrix
- \mathbf{D} = damping matrix
- \mathbf{K} = stiffness matrix
- \mathbf{C} = gyroscopic matrix
- \mathbf{K}_G = centrifugal stiffness matrix
- \mathbf{u} = nodal displacement vector
- $\dot{\mathbf{u}}$ = nodal point velocities at time
- $\ddot{\mathbf{u}}$ = nodal point acceleration at time
- Ω = rotating speed

The chief difference between the equations for the rotating and fixed frames is the inclusion of the centrifugal stiffness matrices \mathbf{K}_G . Additionally, the coefficient on the gyroscopic matrix, \mathbf{C} , as well as its content are modified in the rotating-frame case. Specifically, the rotating system includes an additional Coriolis contribution to \mathbf{C} .

2. **Spin Softening and Gyroscopic Effects.** Rotational Dynamics generates additional terms of spin softening and gyroscopic effects. For transient simulations adding both terms makes the stiffness matrix nonsymmetric which doubles the storage and cost for the numerical factorization. For large problems the user might consider using ISTFNS = 2 which adds only the spin softening terms. This feature maintains the symmetry of the stiffness matrix, hence, avoiding the doubling of the cost of the numerical factorization. No matter the value of ISTFNS, all the effects of rotational dynamics are included in the force computation.
3. **Eigen-Frequencies.** In many applications of rotational dynamics, the critical speed – the theoretical angular velocity that excites the natural frequency of a rotating object – is of particular concern. Therefore, the study of mode frequency response with the change of the rotating speed is very important. The Campbell diagram, which is defined to represent a system's eigen-frequencies as a function of rotating speeds, is introduced for this purpose. In order to do this, the user needs to define a set of rotating speeds on Card 2, and LS-DYNA will do modal analysis for each of these speeds. NOMEQ should be defined as the number of rotating speeds specified in Card 2. A keyword file example in this application can be set as follows:

```
*KEYWORD
*CONTROL_TERMINATION...
*CONTROL_IMPLICIT_EIGENVALUE
      5

*CONTROL_IMPLICIT_GENERAL
      1      0.05
*CONTROL_IMPLICIT_ROTATIONAL_DYNAMICS
$#     SID      STYPE      OMEGA      VID      NOMEQA      IREF
      1          0      0.0          1          4          1
$#     OMEG1    OMEG2    OMEG3    OMEG4
```

```

      50.0    100.0    150.0    200.0
*DEFINE_VECTOR
$#   VID       XT       YT       ZT       XH       YH       ZH     CID
    1       0.0      0.0      0.0      1.0      0.0      0.0
*DATABASE_...
*PART...
*SECTION...
*MAT...
*ELEMENT...
*NODE...
*END

```

4. **Transient Analysis.** Along with modal analysis, transient analysis can also be done using this keyword. A keyword file example can be set as follows:

```

*KEYWORD
*CONTROL_TERMINATION...
*CONTROL_IMPLICIT_GENERAL
    1      0.05
*CONTROL_IMPLICIT_ROTATIONAL_DYNAMICS
$#   SID      STYPE      OMEGA      VID      NOMEWA      IREF
    1          0        0.0          1            0            0
*DEFINE_VECTOR
$#   VID       XT       YT       ZT       XH       YH       ZH     CID
    1       0.0      0.0      0.0      1.0      0.0      0.0
*DATABASE_...
*PART...
*SECTION...
*MAT...
*ELEMENT...
*NODE...
*END

```

5. **Limitations.** The rotational dynamics effects are currently not available for solid and thick shell elements in a fixed coordinate system ($IREF = 1$), so the Campbell diagram in fixed coordinate only shows horizontal lines for these two element types. You may use $IREF = 0$ to see the rotational dynamics effects. The rotational effects are available for other element types, such as shells and beams, in both fixed ($IREF = 1$) and rotating ($IREF = 0$) coordinate systems.
6. **Plotting Campbell Diagrams in LS-PrePost.** To plot a Campbell diagram (frequency as a function of rotating speed) using LS-PrePost, select Post → ASCII → eigout → Load → Frequency → (select modes of interest) → Plot.
7. **Choosing the Reference Frame.** The following table indicates how to choose the reference frame:

Fixed Coordinate System	Rotating Coordinate System
Structure must be axisymmetric about the spin axis.	Structure need not be axisymmetric about the spin axis.
Rotating structure can be part of a stationary structure.	Rotating structure must be the only part of an analysis model.

*CONTROL

*CONTROL_IMPLICIT_ROTATIONAL_DYNAMICS

8. **Comparison to Similar Keywords.** The following table compares the capabilities for keywords *LOAD_BODY, *LOAD_BODY_GENERALIZED and *CONTROL_IMPLICIT_ROTATIONAL_DYNAMICS:

Feature	*LOAD_BODY	*LOAD_BODY_GENERALIZED	*CONTROL_IMPLICIT_ROTATIONAL_DYNAMICS
Explicit	✓	✓	✗
Implicit	✓	✓	✓
Centrifugal force	✓	✓	✓
Coriolis force	✓	✓	✓
Angular acceleration force	✗	✓	✗
Rotating reference frame	✓	✓	✓
Fixed reference frame	✗	✗	✓
Mode frequency depends on spin softening	✓	✓	✓
Mode frequency depends on gyroscopic effects	✗	✗	✓
Complex eigensolver	✗	✗	✓
Moving rotational axis	✗	✗	✓
Campbell diagram	✗	✗	✓

***CONTROL_IMPLICIT SOLUTION_{OPTION}**

Available options include:

<BLANK>

DYN

SPR

Purpose: These optional cards apply to implicit calculations. Use these cards to specify whether a linear or nonlinear solution is desired. Parameters are also available to control the implicit nonlinear and arc length solution methods (see also [*CONTROL_IMPLICIT_GENERAL](#)). The DYN option allows setting controls specifically for the dynamic relaxation phase. The SPR option allows setting controls specifically for the springback phase.

Card Summary:

Card 1. This card is required.

NSOLVR	ILIMIT	MAXREF	DCTOL	ECTOL	RCTOL	LSTOL	ABSTOL
--------	--------	--------	-------	-------	-------	-------	--------

Card 2. Card 2 and beyond are optional.

DNORM	DIVERG	ISTIF	NLPRINT	NLNORM	D3ITCTL	CPCHK	
-------	--------	-------	---------	--------	---------	-------	--

Card 2.1. Card 2.1 is read if DNORM is less than 0.

DMTOL	EMTOL	RMTOL		NTTOL	NRTOL	RTTOL	RRTOL
-------	-------	-------	--	-------	-------	-------	-------

Card 3. The contents of this card are ignored unless an arc-length method is activated ($6 \leq \text{NSOLVR} \leq 9$, or $\text{NSOLVR} = 12$ and $\text{ARCMTH} = 3$).

ARCCTL	ARCDIR	ARCLEN	ARCMTH	ARCDMP	ARCPSI	ARCALF	ARCTIM
--------	--------	--------	--------	--------	--------	--------	--------

Card 4. This card is optional.

LSMTD	LSDIR	IRAD	SRAD	AWGT	SRED	KSSIZE	
-------	-------	------	------	------	------	--------	--

*CONTROL

*CONTROL_IMPLICIT SOLUTION

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	NSOLVR	ILIMIT	MAXREF	DCTOL	ECTOL	RCTOL	LSTOL	ABSTOL
Type	I	I	I	F/I	F/I	F/I	F	F
Default	12	11	15	0.001	0.01	10^{10}	0.90	10^{-10}

VARIABLE	DESCRIPTION
NSOLVR	Solution method for implicit analysis: EQ.-12: Linear analysis with nonlinear contacts (see Remark 1) EQ.-1: Multistep linear (see Remark 1) EQ.1: Linear (see Remark 1) EQ.6: Nonlinear with BFGS updates + arc length (see Remark 2) EQ.7: Nonlinear with Broyden updates + arc length (see Remark 2) EQ.8: Nonlinear with DFP updates + arc length (see Remark 2) EQ.9: Nonlinear with Davidon updates + arc length (see Remark 2) EQ.12: Nonlinear with BFGS updates (default). This solver incorporates different line search and integration schemes as compared to obsolete NSOLVR = 2. This solver optionally allows the inclusion of an arc length method. Setting ARCMTH = 3 invokes an arc length method. EQ.13: Nonlinear with JFNK updates. See parameter KSSIZE on Card 4 and Remark 8 .
ILIMIT	Iteration limit between automatic stiffness reformations (see Remark 3). LT.0: -ILIMIT references a curve that defines reformation limit as a function of time.
MAXREF	Stiffness reformation limit per time step.

VARIABLE	DESCRIPTION							
	LT.0: If $ \text{MAXREF} $ matrix reformations occur, convergence for that time step is forced; see Remark 4 .							
DCTOL	Displacement relative convergence tolerance (see Remark 5). LT.0: -DCTOL references a curve that defines tolerance as a function of time.							
ECTOL	Energy relative convergence tolerance (see Remark 5). LT.0: -ECTOL references a curve that defines tolerance as a function of time.							
RCTOL	Residual (force) relative convergence tolerance (see Remark 5). LT.0: -RCTOL references a curve that defines tolerance as a function of time.							
LSTOL	Line search convergence tolerance; see Remark 6 . LT.0: -LSTOL is the line search tolerance, but this option activates an alternate strategy where line search acts only on the independent degrees of freedom. This is opposed to the default strategy, where prescribed motions on nodes and rigid bodies are also incorporated, sometimes leading to unnecessarily small time steps because of the requirement of fulfilling these boundary conditions.							
ABSTOL	Absolute convergence tolerance. LT.0: Convergence detected when the residual norm is less than $ ABSTOL $. Note: To drive convergence based on $ ABSTOL $, set DCTOL and ECTOL to 10^{-20} .							

Card 2	1	2	3	4	5	6	7	8
Variable	DNORM	DIVERG	ISTIF	NLPRINT	NLNORM	D3ITCTL	CPCHK	
Type	I	I	I	I	F/I	I	I	
Default	2	1	1	0	2	0	0	

*CONTROL

*CONTROL_IMPLICIT SOLUTION

Strict Tolerances Optional Card. Define this card if DNORM is less than 0.

Card 2.1	1	2	3	4	5	6	7	8
Variable	DMTOL	EMTOL	RMTOL		NTTOL	NRTOL	RTTOL	RRTOL
Type	F	F	F		F	F	F	F
Default	0.0	0.0	0.0		0.0	0.0	0.0	0.0

VARIABLE	DESCRIPTION
DNORM	Displacement norm for convergence test (see Remark 5): EQ.1: Increment as a function of displacement over current step EQ.2: Increment as a function of total displacement (default) LT.0: DNORM ; also activates reading of optional Card 2.1.
DIVERG	Divergence flag (force imbalance increases during equilibrium iterations): EQ.1: Reform stiffness if divergence detected (default) EQ.2: Ignore divergence (see Remark 3)
ISTIF	Initial stiffness formation flag (see Remark 3): EQ.1: Reform stiffness at start of each step (default) EQ.n: Reform stiffness at start of every n th step LT.0: Curve ID = (-ISTIF) gives the number of steps to wait before reforming the stiffness as a function of time.
NLPRINT	Nonlinear solver print flag (see Remark 5): EQ.1: Print iteration information to screen, message, d3hsp files (default) EQ.2: Print extra norm information (NLNORM = 1) EQ.3: Same as 2, but also print information from line search NOTE: during execution, interactive commands can be used: <u>interactive command</u> <u>response</u> <ctrl-c> diagnostic toggle NLPRINT between 1 and 2 <ctrl-c> information set NLPRINT = 2 for one iteration

VARIABLE	DESCRIPTION
NLNORM	Nonlinear convergence norm type (see Remark 5 ; input is an integer if ≥ 0 and a float if < 0): LT.0: Same as 4, but rotational degrees of freedom are scaled with characteristic length $ NLNORM $ to account for units. EQ.1: Consider translational and rotational degrees of freedom EQ.2: Consider translational degrees of freedom only (default) EQ.4: Consider sum of translational and rotational degrees of freedom, that is, no separate treatment.
D3ITCTL	Control d3iter database. If nonzero, the search directions for the nonlinear implicit solution are written to the d3iter database. To reduce the size of the d3iter database, the database is reset every D3ITCTL time steps.
CPCHK	Contact penetration check flag. This flag does not apply to mortar contacts. EQ.0: No contact penetration check is performed (default). EQ.1: Check for contact penetration during the nonlinear solution procedure. If such penetration is found, modify the line search to prevent unnecessary penetration.
DMTOL	Maximum displacement convergence tolerance; convergence is detected when the <i>relative</i> maximum nodal or rigid body displacement is less than this value. See Remark 5 .
EMTOL	Maximum energy convergence tolerance; convergence is detected when the <i>relative</i> maximum nodal or rigid body energy increment is less than this value. See Remark 5 .
RMTOL	Maximum residual convergence tolerance; convergence is detected when the <i>relative</i> maximum nodal or rigid body residual is less than this value. See Remark 5 .
NTTOL	Nodal translational convergence tolerance; convergence is detected when the <i>absolute</i> maximum nodal translational residual is less than this value. See Remark 5 .
NRTOL	Nodal rotational convergence tolerance; convergence is detected when the <i>absolute</i> maximum nodal rotational residual is less than this value. See Remark 5 .

*CONTROL

*CONTROL_IMPLICIT SOLUTION

VARIABLE	DESCRIPTION
RTTOL	Rigid body translational convergence tolerance; convergence is detected when the <i>absolute</i> maximum rigid body translational residual is less than this value. See Remark 5 .
RRTOL	Rigid body rotational convergence tolerance; convergence is detected when the <i>absolute</i> maximum rigid body rotational residual is less than this value. See Remark 5 .

Arc Length Optional Card. The contents of this card are ignored unless an arc-length method is activated ($6 \leq \text{NSOLVR} \leq 9$, or $\text{NSOLVR} = 12$ and $\text{ARCMTH} = 3$).

Card 3	1	2	3	4	5	6	7	8
Variable	ARCCTL	ARCDIR	ARCLEN	ARCMTH	ARCDMP	ARCPSI	ARCALF	ARCTIM
Type	I	I	F	I	I	F	F	F
Default	0	↓	0.	1	2	0.	0.	0.

VARIABLE	DESCRIPTION
ARCCTL	Arc length controlling node ID (see Remark 7). EQ.0: Generalized arc length method
ARCDIR	Arc length controlling node direction (ignored if ARCCTL = 0 above; see Remark 7): EQ.1: Global <i>x</i> -translation EQ.2: Global <i>y</i> -translation EQ.3: Global <i>z</i> -translation
ARCLEN	Relative arc length size. See Remark 7 . LE.0.0: Use automatic size. GT.0.0: Use ARCLEN \times (automatic step size).
ARCMTH	Arc length method (see Remark 7): EQ.1: Crisfield (default) EQ.2: Ramm

VARIABLE	DESCRIPTION
	EQ.3: Modified Crisfield (used with NSOLVR = 12 only) Note: When using the arc-length method, termination should be based on some criteria using *TERMINATION_OPTION keyword.
ARCDMP	Arc length damping option (see Remark 7): EQ.2: Off (default) EQ.1: On. Oscillations in the static solution are suppressed.
ARCPSI	Relative influence of load/time parameter in spherical arclength constraint; default value is 0 which corresponds to a cylindrical arclength constraint. Applies to ARCMTH = 3.
ARCALF	Relative influence of predictor step direction for positioning of the arc center; default is 0 which means that the center is at the origin. Applies to ARCMTH = 3.
ARCTIM	Optional time when arclength method is initiated. Applies to ARCMTH = 3.

Line Search Parameter Optional Card.

Card 4	1	2	3	4	5	6	7	8
Variable	LSMTD	LSDIR	IRAD	SRAD	AWGT	SRED	KSSIZE	
Type	I	I	F	F	F	F	I	
Default	4	2	0.0	0.0	0.0	0.0	10	

VARIABLE	DESCRIPTION
LSMTD	Line search convergence method (see Remark 6): EQ.1: Energy method using only translational variables EQ.2: Residual method EQ.3: Energy method using both translational and rotational variables EQ.4: Energy method using sum of translational and rotational degrees of freedom (default), that is, no separate

VARIABLE	DESCRIPTION
	treatment
	EQ.5: Same as 4, but account for residual norm growth to be extra conservative in step length
	EQ.6: Same as 5, but minimizes the residual norm whenever convenient
LSDIR	Line search direction method (see Remark 6): EQ.1: Search on all variables (traditional approach used in versions prior to 971) EQ.2: Search only on the independent (unconstrained) variables EQ.3: Use adaptive line search (see AWGT, SRED) EQ.4: Use curved line search (see IRAD, SRAD)
IRAD	Normalized curvature factor for curved line search, where 0 indicates a straight line search and 1 indicates full curved line search. See Remark 6 .
SRAD	Radius of influence for determining curve in curved line search. For each independent node, all nodes within this radius are used for determining the curve. If 0, then all nodes connected to the same element as the independent node are used. See Remark 6 .
AWGT	Adaptive line search weight factor between 0 and 1. A high value tends to restrict the motion of oscillating nodes during the implicit process. See Remark 6 .
SRED	Initial step reduction between 0 and 1 for adaptive line search; use large number for conservative start in implicit procedure. See Remark 6 .
KSSIZE	Size of Krylov space in JFNK iterative method. The default is 10. Decreasing the default value tends to make the problem more robust but potentially more expensive to solve. Increasing it could save simulation time but may also result in a less robust scheme. See Remark 8 .

Remarks:

1. **Linear solution methods.** NSOLVR = 1 invokes a linear analysis for a user specified number of time steps. The results are cumulative for those time steps. NSOLVR = -1 resets the geometry back to the initial state for each time step.

With this method, a linear analysis for several different loading cases with only one stiffness matrix formation and factorization can be done. If a linear analysis is selected, equilibrium checking and iterations are not performed. NSOLVR = -12 causes a hybrid approach, where all features are linearized except for contacts. This option is suitable for situations with small displacement for which the state of the contact is important.

2. **Unmaintained nonlinear solution methods.** The nonlinear solution methods 6-9 are not being maintained. NSOLVR = 12 with ARCMTH = 3 is the suggested method for arc length solvers.
3. **Stiffness matrix reformation.** In the default BFGS method, the global stiffness matrix is only reformed every ILIMIT iterations. Otherwise, an inexpensive stiffness update is applied. By setting ILIMIT = 1, a stiffness reformation is performed every iteration which is equivalent to the Full Newton method (with line search). A higher value of ILIMIT (20-25) can reduce the number of stiffness matrix reformations and factorizations which may lead to a significant reduction in cost. However, increasing the value of ILIMIT may increase the number of iterations. Note that the storage requirements for implicit include storing two vectors per iteration. Therefore, large values of ILIMIT will cause a substantial increase in storage requirements. By using a negative ILIMIT, it is possible to switch between Full Newton and various degrees of BFGS methods, if for instance nonlinearity varies significantly during the simulation.

By default, a new stiffness matrix is formed whenever divergence (growing out-of-balance force) is detected. The DIVERG flag can be used to suppress this stiffness reformation. Also, by default, a new stiffness matrix is formed at the start of every time step. Suppressing this stiffness reformation by changing the value of ISTIF can decrease the cost of simulations which have many tiny steps that are mostly linear, such as transient dynamics.

4. **MAXREF.** The nonlinear equilibrium search will continue until the stiffness matrix has been reformed $|MAXREF|$ times, with ILIMIT iterations between each reformation. If equilibrium has not been found and $MAXREF > 0$, control will be passed to the automatic time step controller if it is activated. If the automatic time step controller is not active, error termination will result. When the auto time step controller is active, it is often efficient to choose $MAXREF = 5$ and try another stepsize quickly, rather than wasting too many iterations on a difficult step.

When $MAXREF < 0$ and $|MAXREF|$ matrix reformations have occurred, convergence for the current time step is declared, with a warning, and the simulation moves to the next time step. This option should be used with caution as the results for that particular time step may be wrong.

5. **Convergence criteria.** For convergence criteria DCTOL, ECTOL, and RCTOL, the conditions are satisfied when the displacement norm ratio, energy norm ratio, and residual norm ratio are reduced below DCTOL, ECTOL, and RCTOL, respectively. Smaller tolerances lead to a stricter determination of equilibrium, but also result in more iterations and higher costs. By default, residual norm ratio (RCTOL) criterion is effectively disabled (RCTOL = 10¹⁰).

When computing the displacement ratio, the norm of the incremental displacement vector is divided by the norm of “total” displacement. This “total” displacement may be either the total over the current step, or the total over the entire simulation. The latter tends to be more lax and can be poor at the end of simulations where large motions develop. For these problems, an effective combination is DNORM = 1 and DCTOL = 0.01 or larger.

For all nonzero values of the strict tolerance fields in optional Card 2.1, the associated criterion must be satisfied *in addition* to the ones defined through DCTOL, ECTOL, and RCTOL. These criteria are based on the maximum norm, which is regarded as stronger than the Euclidian norm used for the other parameters; using these criteria will likely result in higher accuracy at the price of more iterations. For NLPRINT greater than or equal to 2, a table is listed in the messag and d3hsp files for each iteration, providing the values associated with all the criteria activated. The first three (DMTOL, EMTOL and RMTOL) of these extra fields are unitless and honor the meaning of both DNORM and NLNORM. The last four (NTTOL, NRTOL, RTTOL and RRTOL) are to be given in units of force, torque, force, and torque, respectively, and the values used should account for the representative loads in the problem as well as the discretization size.

By default, only translational degrees of freedom are used when evaluating convergence norms. NLNORM can be set to include rotational degrees of freedom or to make additional data available for diagnosing convergence problems. This additional data includes the worst offending node and degree of freedom contributing to each norm.

By setting NLNORM to 1, rotational degrees of freedom can be considered independently from the translational degrees of freedom, meaning that two separate scalar products are used for evaluating norms, $\langle \mathbf{u}, \mathbf{v} \rangle_t = \mathbf{u}^T \mathbf{J}_t \mathbf{v}$ and $\langle \mathbf{u}, \mathbf{v} \rangle_r = \mathbf{u}^T \mathbf{J}_r \mathbf{v}$. Here \mathbf{J}_t and \mathbf{J}_r are diagonal matrices with ones on the diagonal to extract the translational and rotational degrees of freedom, respectively. In this case, the convergence criteria must be satisfied for both translational and rotational degrees of freedom simultaneously.

Alternatively they can be included by defining the single scalar product $\langle \mathbf{u}, \mathbf{v} \rangle = \langle \mathbf{u}, \mathbf{v} \rangle_t + \lambda_u \lambda_v \langle \mathbf{u}, \mathbf{v} \rangle_r$, where λ_u and λ_v are scale factors to account for different units of the rotational degrees of freedom. For NLNORM = 4 these scale factors are equal to a characteristic element size that is calculated internally. But for

$\text{NLNORM} < 0$ λ_u is equal to $|\text{NLNORM}|$ if \mathbf{u} is a displacement vector and $|\text{NLNORM}|^{-1}$ if it is a force vector; the same goes for the pair λ_v and \mathbf{v} . So $|\text{NLNORM}|$ is a characteristic length that appropriately weighs translational and rotational degrees of freedom together. For $\text{NLNORM} = 4$ the internally calculated size is echoed to the screen, d3hsp and messag files for observation and based on this $\text{NLNORM} < 0$ can be used if another scale factor is more suitable.

6. **Line search.** A line search is performed on stiffening systems to guard against divergence of Newton-based nonlinear solvers. With the Full Newton method, it is sometimes helpful to define a large value of LSTOL ($\text{LSTOL} = 9999.0$) to effectively disable line search.

The default method for determining convergence of the nonlinear line search is to find the minimum of the energy. LMSTD allows choosing the energy on only the translational variables, energy of both the translational and rotational variables, or for minimizing the residual (forces). The effect of using a residual based line search is not always positive; sometimes it is too restrictive and stops convergence. But it is a more conservative approach than using the energy based method since it explicitly controls the norm of the residual. It should not be seen as a better strategy than the energy method but as an alternative to try in cases when the default method seems to be working poorly. Line search methods 5 and 6 are conservative line search methods to be used for highly nonlinear problems; these should not be used as default but as final resorts to potentially resolve convergence issues. The rule of thumb is that the $\text{LSMTD} = 5$ is slow but robust and $\text{LSMTD} = 6$ is even slower but more robust.

In Version 971 of LS-DYNA, new line search options were added. The traditional approach ($\text{LSDIR} = 1$) computes the line search direction using all variables. The new (default) approach of $\text{LSDIR} = 2$ computes the line search direction only on the unconstrained variables. It has proven to be both robust and more efficient. We have also included two new approaches to try for problems where the default and traditional approach fail, and the user is using Full Newton ($\text{ILIMIT} = 1$). These methods are described below:

- a) *Curve Line Search.* The parameters IRAD and SRAD are for the curved line search ($\text{LSDIR} = 4$). The first parameter is a switch (0 or 1) to invoke this line search; an intermediate value is interpreted as weighted combination of a straight and curved line search (the curvature radius is decreased with increasing IRAD). A value of unit is recommended in situations with rather smooth responses, such as springback. Also, $\text{IRAD} = 1$ seems to work best with full Newton iterations. The SRAD parameter should be equal to 0 for most cases; this means that the search curve for a node is determined from the search direction of nodes connected to the same elements as that node. $\text{SRAD} > 0$ is interpreted as a radius of influence, meaning that the search curve for a node is determined from the search direction of nodes within a

distance SRAD of this node. This option was introduced as an experiment to see if this had a smoothing and stabilizing effect. A value of 0.0 is currently recommended.

- b) *Adaptive Line Search.* The parameters AWGT and SRED are for the adaptive line search. The intention is to improve robustness for problems that have tendencies to oscillate or diverge, indicated by the dnorm and enorm parameter outputs in the iterations (stdout). A value of 0.5 is recommended for AWGT as a starting point. With a nonzero value the motions of individual nodes are tracked. For nodes that are oscillating (going back and forth in space), the maximum step size for the next iteration is reduced in proportion to the parameter AWGT, and for nodes that are not oscillating but going nicely along a straight path, the maximum step size for the next iteration is increased in proportion to 1-AWTG.

In test problems, the introduction of the adaptive line search has stabilized the implicit procedure in the sense that the dnorm and enorm values are more monotonically decreasing until convergence with virtually no oscillations. If a problem is still oscillating or diverging, the user should try to increase the AWGT parameter since this is a more restrictive approach but probably gives a slower convergence rate. An option for nasty problems is also to use SRED > 0 which is the initial step reduction factor (less than 1). This means that the initial step size is reduced by this value but the maximum step size will increase by an amount that is determined by the success in the iterative procedure; eventually it will reach unity. It can never decrease. Also here, it is intended to be used with full Newton method.

7. **Arc length methods.** In the neighborhood of limit points the Newton based iteration schemes often fail. The arc length method of Riks and Wempner (combined here with the BFGS method) adds a constraint equation to limit the load step to a constant "arc length" in load-displacement space. This method is frequently used to solve snap through buckling problems. *When applying the arc-length method, the curves that define the loading should contain only two points, and the first point should be at the origin (0,0).* LS-DYNA will extrapolate, if necessary, to determine the load. In this way, time and load magnitude are related by a constant. It is possible that time can become negative in case of load reversal. The arc length method cannot be used in a dynamic analysis.

In many cases the arc length method has difficulty tracking the load-displacement curve through critical regions. Using $0 < \text{ARCLEN} < 1$ will reduce the step size to assist tracking the load-displacement curve with more accuracy. Use of $\text{ARCLEN} < 1$ will cause more steps to be taken. Suggested values are 1.0 (the default), 0.5, 0.25, and 0.10.

The arc length method can be controlled based on the displacement of a single node in the model using ARCCTL. For example, in dome reversal problems the node at the center of the dome can be used. By default, the generalized arc length method is used, where the norm of the global displacement vector controls the solution. This includes all nodes.

Some static problems exhibit oscillatory response near instability points. ARCDMP numerically suppresses these oscillations and may improve the convergence behavior of the post-buckling solution.

8. **JFNK.** The Jacobian-Free Newton-Krylov (JFNK) method is a multidisciplinary variant of the linear GMRES method that may potentially bridge the gap between full Newton and BFGS. In each iteration, the method constructs a Krylov subspace of dimension KSSIZE based on recursively computing force vectors from ϵ -perturbations of the current geometry. The method then obtains a search direction within this subspace by means of a minimization principle. Utilizing the factorization of the stiffness matrix from the last stiffness reformation as a preconditioner accelerates the method. Since the method exploits the quasi-linear characteristics of the force in the vicinity of a geometry iterate, this method potentially increases the robustness when compared to BFGS. The method seems particularly promising for very nonlinear problems.
9. **Deprecated keyword name.** *CONTROL_IMPLICIT_NONLINEAR is a deprecated name for this keyword.

*CONTROL

*CONTROL_IMPLICIT_SOLVER

*CONTROL_IMPLICIT_SOLVER_{OPTION}

Available options include:

<BLANK>

DYN

SPR

Purpose: These optional cards apply to implicit calculations. The linear equation solver performs the CPU-intensive stiffness matrix solution (see also *CONTROL_IMPLICIT_GENERAL). The DYN option allows setting controls specifically for the dynamic relaxation phase. The SPR option enables setting controls specifically for the springback phase.

Card 1	1	2	3	4	5	6	7	8
Variable	LSOLVR	LPRINT	NEGEV	ORDER	DRCM	DRCPRM	AUTOSPC	AUTOTOL
Type	I	I	I	I	I	F	I	F
Default	7	0	2	0	4	↓	1	↓

Card 2 is optional.

Card 2	1	2	3	4	5	6	7	8
Variable	LCPACK	MTXDMMP	IPARM1	RPARM1	RPARM2		RPARM4	RPARM5
Type	I	I	I	F	F		F	F
Default	2	0	↓	10 ⁻¹²	10 ⁻⁸		240.0	0.0

Card 3 is optional.

Card 3	1	2	3	4	5	6	7	8
Variable	EMXDMPI	RDCMEM	ABSMEM	ISINGLE	IBLROPT		ISPD	MEMLVL
Type	I	F	F	I	I		I	F
Default	0	0.85	optional	0	0		0	0

VARIABLE	DESCRIPTION
LSOLVR	<p>Linear equation solver method (see Remark 1).</p> <p>EQ.2: Parallel multi-frontal sparse solver (deprecated)</p> <p>EQ.7: Parallel multi-frontal sparse solver (default) Iterative if ISINGLE > 0 or IBLROPT > 0.</p> <p>EQ.22: Iterative, CG with diagonal preconditioner when LCPACK = 2 (default) or GMRES with diagonal preconditioner when LCPACK = 3 (nonsymmetric matrix assembly)</p> <p>EQ.23: Iterative, CG with local Symmetric Gauss-Seidel (SGS) preconditioner</p> <p>EQ.24: Iterative, CG with local Symmetric Successive Over-Relaxation (SSOR) preconditioner</p> <p>EQ.25: Iterative, CG with local 0-fill incomplete factorization (ILDLT0) preconditioner</p> <p>EQ.26: Iterative, CG with local 0-fill incomplete factorization (ILDLT0) preconditioner that requires extra storage</p> <p>EQ.30: Parallel direct/hybrid solver MUMPS (see Remark 2)</p> <p>EQ.90: User-Supplied Linear Equation Solver (see Remark 3)</p> <p>SMP only:</p> <p>EQ.6: BCSLIB-EXT, direct, sparse, double precision</p>
LPRINT	<p>Linear solver print flag controls screen and message file output (see Remarks 4 and 6).</p> <p>EQ.0: No printing</p> <p>EQ.1: Output summary statistics on memory, CPU requirements</p>

*CONTROL

*CONTROL_IMPLICIT_SOLVER

VARIABLE	DESCRIPTION
	EQ.2: More statistics EQ.3: Even more statistics and debug checking During execution, use the interactive command "<ctrl-c> lprint" to toggle this print flag between 0 and 1.
NEGEV	Negative eigenvalue flag. Selects procedure when negative eigenvalues are detected during stiffness matrix inversion (see Remark 5). EQ.1: Stop or retry step if auto step control is active EQ.2: Print warning message, try to continue (default)
ORDER	Ordering option (see Remark 6): EQ.0: Method set automatically by LS-DYNA EQ.1: MMD, Multiple Minimum Degree EQ.2: METIS EQ.3: ParMETIS EQ.4: LS-GPart
DRCM	Drilling rotation constraint method for shells (see Remark 7): EQ.1: Add drilling stiffness (old Version 970 method) EQ.2: Same as 4 below EQ.3: Add no drilling stiffness EQ.4: Add drilling stiffness (improved method) (default)
DRCPRM	Drilling rotation constraint parameter for shells. This parameter scales the drilling stiffness. For the old method ($DRCM = 1$), the default value of DRCPRM is 1.0 for linear analysis, 100.0 for nonlinear implicit analysis, and either 10^{-12} or 10^{-8} for eigenvalue analysis, depending on the shell element type. For eigenvalue analysis, the input value for DRCPRM is ignored. For the improved method (default, $DRCM = 4$), the default value of DRCPRM is as described above for the old method, except the default value of DRCPRM is 1.0 for nonlinear implicit analysis.
AUTOSPC	Automatic Constraint Scan flag EQ.1: Scan the assembled stiffness matrix looking for unconstrained, unattached degrees of freedom. Generate additional constraints as necessary to avoid negative eigenvalues.

VARIABLE	DESCRIPTION
AUTOTOL	<p>EQ.2: Do not add constraints.</p> <p>AUTOSPC tolerance. The test for singularity is the ratio of the smallest singular value and the largest singular value. If this ratio is less than AUTOTOL, then the triple of columns is declared singular, and a constraint is generated. Default values in single and double precision are 10^{-4} and 10^{-8}, respectively.</p>
LCPACK	<p>Matrix assembly package:</p> <p>EQ.2: Default.</p> <p>EQ.3: Same as 2, but incorporates a nonsymmetric linear solver; see Remark 8.</p>
MTXDMP	<p>Matrix and right-hand-side dumping. LS-DYNA has the option of dumping the globally assembled stiffness matrix and right-hand-side vector files in Harwell-Boeing sparse matrix format. Such output may help with comparing to other linear equation solution packages.</p> <p>EQ.0: No dumping</p> <p>GT.0: Dump all matrices and right-hand-side vectors every MTXDMP time steps. The first dump occurs at the first time step, then at time step MTXDUMP+1, 2*MTXDUMP+1, and so on. Output is written as ASCII text, and the involved filenames are of the following form:</p> <p style="text-align: center;"><u>K xxxx yyy mtx.rb</u></p> <p>This file contains the stiffness matrix at step xxxx, iteration yyy.</p> <p style="text-align: center;"><u>M xxxx yyy mtx.rb</u></p> <p>This file contains the mass matrix at step xxxx, iteration yyy. Only for eigenvalue analysis.</p> <p style="text-align: center;"><u>W xxxx yyy mtx.rb</u></p> <p>This file contains the damping matrix at step xxxx, iteration yyy. Only for simulations with damping.</p> <p style="text-align: center;"><u>K xxxx yyy zzz rhs.rb</u></p> <p>This file contains the right-hand side at step xxxx, iteration yyy, where yyy is the iteration at which a stiffness matrix is formed, and zzz is the cumulative iteration number for the step. The values of yyy and zzz don't always coincide because the stiffness matrix</p>

VARIABLE	DESCRIPTION
	is not necessarily reformed every iteration.
	<u>Node Data xxxx yyyy</u>
	This file maps the stiffness matrix to nodes and provides nodal coordinates.
LT.0:	Like positive values of MTXDMP, but dumped data is binary.
EQ. 9999 :	Simulation is terminated after dumping matrices and right-hand side before factorization.
IPARM1	For iterative solvers, maximum number of iterations. The default is 1000 for LSOLVR = 7. The default is 10000 for $22 \leq LSOLVR \leq 26$.
RPARM1	For iterative solvers, absolute tolerance for convergence. The default is 10^{-12} .
RPARM2	For iterative solvers, relative tolerance for convergence. The default is 10^{-8} .
RPARM4	For LSOLVR = 7 with LCPACK = 2 (symmetric) and IBLROPT > 0, nominal block size. The default is 240.
RPARM5	For LSOLVR = 7 with LCPACK = 2 (symmetric) and IBLROPT > 0, compression tolerance for the low-rank approximation. For LSOLVR = 30, RPARM5 is the compression tolerance used to compute a low-rank factorization with the MUMPS solver (see Remark 2).
EMXDMP	Flag for dumping elemental stiffness and mass matrices: EQ.0: No dumping GT.0: Dump all elemental matrices every EMXDMP time steps. Output is written as ASCII text, and the involved filenames are of the following form: <u>ElmStfMtx xxxx yyyy</u> This file contains the elemental stiffness matrix at step xxxx, iteration yyyy. <u>ElmMssMtx xxxx yyyy</u> This file contains the elemental mass matrix at step xxxx, iteration yyyy. LT.0: Like positive values of MTXDMP, but dumped data is binary. EQ. 9999 : Simulation is terminated after dumping matrices and

VARIABLE	DESCRIPTION
	right-hand side before factorization.
RDCMEM	Starting with LS-DYNA R11, the memory for linear algebra has been moved from static to dynamic memory allocation. For implicit applications, we found that some operating systems are not “robust” when queried about how much dynamic memory is free. This factor caps the amount of dynamic memory requested for linear algebra applications to RDCMEM times the amount the operating system declares available. 0.85 seems to work well for most systems. If you are using a workstation and starting up other applications while running LS-DYNA, you may need to use a number like 0.50.
ABSMEM	Absolute upper bound for the dynamic memory allocated for factorization. The allocated memory will be bounded above by the $\min(RDCMEM \times NWORDS, ABSMEM)$ where NWORDS is the number of available words determined by the operating system. If the predicted amount of required memory is less than this value, then less memory than this bound may be allocated.
	Note that the memory can be specified in this input in the same way as memory is specified on the command line. For instance, an allocation of 3.5 Gwords can be input as 3.5G or 3.5e+9 for ABSMEM.
ISINGLE	Only active if LSOLVR = 7 and LCPACK = 2. This field controls tasks performed in single precision (see Remark 10): EQ.0: No single precision used. EQ.1: The factors of the matrix, which are computed in double precision, are stored in single precision, reducing the storage required to solve the linear system. EQ.2: Same as 1, but additionally some of the factorization operations are performed in single precision, reducing the runtime.
IBLROPT	Flag providing options for the block low-rank (BLR) computations. It is only active if LSOLVR = 7 and LCPACK = 2. As the value of IBLROPT increases, the treatment is increasingly aggressive. However, increasingly aggressive compression can introduce additional error into the factors of the linear system, which can slow down the convergence of the subsequent numerical solve or even prevent convergence. See Remark 10 . EQ.0: BLR is not used.

*CONTROL

*CONTROL_IMPLICIT_SOLVER

VARIABLE	DESCRIPTION
	EQ.1: Factor, Solve, Update, and Compress algorithm (FSUC). A full-rank factorization is performed, and off-diagonal blocks of the factors are then subject to compression with BLR. This method should reduce the storage required to factor and solve the linear system. However, the additional BLR computations increase the runtime.
	EQ.2: Factor, Solve, Compress, and Update algorithm (FSCU). FSUC is applied with the fully assembled equations of the frontal matrix. Where BLR compression is applied, low-rank updates are performed to compute the Schur complement of the frontal matrix. This method should reduce both storage and runtime.
	EQ.3: Update, Factor, Solve, and Compress algorithm (UFSC). Low-rank approximations are used to update blocks within the fully assembled equations in a frontal matrix. Where BLR compression is applied, low-rank computations are also performed when computing the Schur complement of the frontal matrix. This option should reduce storage and further reduce runtime.
	EQ.4: UFSC is performed. In addition, BLR is applied to the contribution blocks of low-rank frontal matrices before they are placed on the real stack. This step further reduces the working storage needed for the low-rank factorization. The additional BLR computations increase the runtime relative to UFSC.
ISPD	Symmetric positive definite flag. Only active when LSOLVR = 7 and LCPACK = 2. EQ.0: The solver only assumes that the linear system is symmetric positive definite if the diagonals of the factorization are all positive. EQ.1: The solver assumes that the linear system is symmetric positive definite and ignores spurious negative diagonal entries.
MEMLVL	Control the load-balancing behavior. Only active when LSOLVR = 7, LCPACK = 2, and ISINGLE > 0. The solver's default behavior is to load balance to reduce runtime. That behavior can cause a subset of processors to have a disproportionate share of the factor entries and drive such processors out-of-core, even when enough storage should be available. MEMLVL enables instructing the solver to consider storage as well.

VARIABLE	DESCRIPTION
	EQ.0.0: Load balancing is entirely based on minimizing in-core runtime.
	GT.1.0: Perform load balancing to reduce storage disparity among the processors.
	GT.1.0: The first load balance is performed to reduce runtime. Then, it is adjusted to reduce any outlier cores, such that they have no more than MEMLVL times an equal share of the factor entries. In this case, MEMLVL is an aspirational target that the solver cannot always achieve.

Remarks:

1. **Direct versus iterative solvers.** The linear solver computes the inverse of the global stiffness matrix, which is costly in both memory and CPU time. Direct solvers apply Gaussian elimination, while iterative solvers successively improve “guesses” at the correct solution. Iterative solvers require far less memory than direct solvers but may suffer from convergence problems. Generally, iterative solvers are poor for automotive applications but can be superior for large brick element models, such as solid models in civil engineering.
2. **MUMPS solver.** Starting with R11.1, we have coupled the external sparse linear solver MUMPS with LS-DYNA. MUMPS can be used as a direct solver (similar to LSOLVR = 7) with RPARM5 = 0 or as a hybrid direct-iterative solver when RPARM5 > 0. When RPARM5 > 0, an approximate factorization is computed using RPARM5 as a compression threshold for Block Low-Rank approximations. The approximate factorization is then used as a preconditioner inside a Conjugate Gradient solver; IPARM1, RPARM1, and RPARM2 are used to drive the Conjugate Gradient solver as described above.

Using MUMPS as the linear solver in LS-DYNA should only be used in conjunction with LS-DYNA Implicit Staff.

MUMPS is the property of CERFACS, CNRS, ENS Lyon, INP Toulouse, Inria, Mumps Technologies, and the University of Bordeaux. See [1] and [2] for references about this solver.

3. **User-supplied solver.** Starting with R11, we have added a capability for a user-supplied solver. Users can follow the template in the file UserLE.F90 to supply their own linear equation solver. Some examples are available in the usermat delivery and from Technical Support.

4. **Solver print flag.** Select printing of the timing and storage information (LPRINT = 1) if you are comparing linear equation solvers' performance or running out of memory for large models. Minimum memory requirements for in-core and out-of-core solutions are printed. This flag can also be toggled using sense switch "<ctrl-c> lprint." *For best performance, increase available memory using "memory= " on the command line until an IN-CORE solution is indicated.*

When using solver option 6, LPRINT = 2 and 3 will increase the printed output of statistics and performance information.

5. **Negative eigenvalues.** Negative eigenvalues result from underconstrained models (rigid body modes), severely deformed elements, or non-physical material properties. When negative eigenvalues are detected, the NEGEV flag allows control to be passed directly to the automatic time step controller. Otherwise, significant numerical roundoff error will likely occur during factorization, and equilibrium iterations may fail (see *CONTROL_IMPLICIT_AUTO).
6. **Ordering algorithms and cost.** The system of linear equations is reordered to optimize the sparsity of the factorization when using direct methods. Reordering is a challenging problem, and no best algorithm exists for all problems. LS-DYNA offers four different methods:
 - a) *MMD (Multiple Minimum Degree)*. This reordering method is inexpensive but should only be used for small problems.
 - b) *METIS*. METIS is from the University of Minnesota. It is the default method for most problems. It is very effective for large problems. METIS, however, is a serial algorithm. Thus, for a large number of MPP processes, you might want to use one of the two parallel algorithms mentioned next.
 - c) *ParMETIS*. ParMETIS is also from the University of Minnesota. It is the MPP version of METIS. For a large number of processes, it should run much faster than METIS, but there might be some variation in the factorization cost (storage and number of operations). It is available as of R14.
 - d) *LS-GPart*. LS-GPart is an MPP algorithm developed by Ansys and available as of R11. For a large number of processes, it should run faster than METIS. Quality (in terms of factorization cost) compared to METIS or Par-METIS will depend on the problem.

The symbolic factorization phase of the linear solver (LPRINT \geq 1) includes the reordering cost. For large models, if this cost exceeds 20% of the numeric factorization cost, it may be more efficient to select the MMD method.

Note that the values of LPRINT and ORDER also affect the eigensolution software. That is, LPRINT and ORDER from this keyword card apply to the eigen-solution.

Please refer to *CONTROL_IMPLICIT_ORDERING for additional parameters for ordering methods.

7. **Drilling degree of freedom.** To avoid a singular stiffness matrix for the implicit analysis of flat shell topologies, some constraint on the drilling degree of freedom is needed. The default method of applying this constraint, DRCM = 4, adds the consistent force vector for consistency and improved convergence compared to the old method, DRCM = 1.

In explicit analysis, an unconstrained drilling degree of freedom is usually not a concern since a stiffness matrix is not used. However, special situations may arise where the user wishes to include additional resisting rotational force in the drilling degree of freedom for improved robustness and/or accuracy. To activate the consistent drilling constraint in explicit analysis, use the input variables DRCPSID and DRCPRM for *CONTROL_SHELL.

8. **Nonsymmetric matrix solver.** Certain features may break the symmetry of the stiffness matrix. Unless LCPACK is set to 3, the default linear solver suppresses or symmetrizes these contributions. However, LCPACK set to 3 activates a more general linear solver, lifting the symmetry requirement. The solver for nonsymmetric matrices is more computationally expensive than for symmetric matrices. LSOLVR = 7 (Parallel multi-frontal sparse solver), 6 (BCSLIB-EXT, direct, sparse, double precision), 22 (CG or GMRES), and 30 (MUMPS) can be used in combination with LCPACK = 3. Other values will default to a direct solver.

For eigenvalue analysis, LCPACK = 3 is only implemented in SMP. LCPACK = 3 is implemented for both SMP and MPP for static and dynamic implicit analyses.

Keywords for which the nonsymmetric contribution is included when LCPACK = 3 are listed below. If none of these features are included in the model, setting LCPACK = 3 offers no benefit but will incur additional computational cost.

***CONTACT_..._MORTAR:**

The mortar contact accounts for frictional nonsymmetry in the resulting tangent stiffness matrix; the effects on convergence characteristics have not yet been shown to be significant.

***LOAD_SEGMENT_NONUNIFORM:**

The nonsymmetric contribution may be significant for the follower load option, LCID < 0.

***LOAD_SEGMENT_SET_NONUNIFORM:**

The nonsymmetric contribution may be significant for the follower load option, LCID < 0.

***MAT_FABRIC_MAP:**

This stress map fabric model accounts for nonsymmetry in the material tangent modulus, representing the non-linear Poisson effect due to the complex interaction of yarns.

***SECTION_SHELL, *SECTION_SOLID:**

Solid element formulation 2, -1, -2 and user-defined resultant elements (ELFORM = 101, 102, 103, 104, 105 with NIP = 0) support the assembly and solution of nonsymmetric element matrices.

***SECTION_BEAM:**

Belytschko-Schwer beam (ELFORM = 2) nonsymmetric geometric stiffness contribution is supported.

9. **Constraints.** Explicit treats all the constraints in sequence, which introduces the limitation of applying several constraints on the same DOF. Implicit (with LCPACK), on the other hand, treats all constraints simultaneously, which precludes (circumvents) the limitation on applying multiple constraints on the same nodes or DOFs, as long as they are not in conflict.
10. **Reduced precision.** Using ISINGLE or BLR can save time and storage when using the default, multi-frontal sparse solver (LSOLVR = 7). The trade-off is that they reduce the precision of the factors of the matrix, turning them into high-quality preconditioners for a subsequent iterative linear solver. If the symmetric matrix is positive definite, the conjugate gradient method is used. If the symmetric matrix is indefinite, the iterative refinement method is used. There is no guarantee that the iterative solver will converge to the requested tolerance. If the iterative solver does not converge, LS-DYNA implicit may not either.
11. **Deprecated keyword name.** *CONTROL_IMPLICIT_LINEAR is a deprecated name for this keyword.

References:

- [1] P. R. Amestoy, I. S. Duff, J. Koster, and J.-Y. L'Excellent, A fully asynchronous multifrontal solver using distributed dynamic scheduling, SIAM Journal on Matrix Analysis and Applications, Vol 23, No 1, pp 15-41 (2001).
- [2] P. R. Amestoy, A. Buttari, J.-Y. L'Excellent, and T. Mary, Performance and scalability of the block low-rank multifrontal factorization on multicore architectures, ACM Transactions on Mathematical Software, Vol 45, Issue 1, pp 2:1-2:26 (2019).

*CONTROL

*CONTROL_IMPLICIT_SSD_DIRECT

*CONTROL_IMPLICIT_SSD_DIRECT

Purpose: Request a direct complex solution to steady state vibration. Currently, this keyword has limited loading and damping capabilities. It is only available for the double precision, SMP versions. Refer to Appendix W for a list of keywords this feature supports.

Card 1	1	2	3	4	5	6	7	8
Variable	ISSFLG	FMIN	FMAX	NFREQ	LOSS	FSPACE	FRACTN	
Type	I	F	F	I	F	I	I	
Default	0	none	FMIN	1	0.0	0	1	

VARIABLE	DESCRIPTION
ISSFLG	Complex steady state vibration flag: EQ.0: Off EQ.1: On
FMIN	Minimum frequency in the solution. Units are Hertz.
FMAX	Maximum frequency in the solution. Units are Hertz.
NFREQ	Number of frequencies in the solution
LOSS	Structural loss factor
FSPACE	Solution frequency assignment strategy: EQ.0: The frequency is interpolated linearly between FMIN and FMAX. This is the default strategy. EQ.1: The frequency is interpolated on a log scale between FMIN and FMAX, so they are biased to lower frequencies. EQ.2: The frequency is interpolated on a fractional octave scale starting with FMIN. Integer FRACTN is the octave fraction. The formula for the active frequency in Hertz is $\text{FACTIVE} = \text{FMIN} \left(2.0^{1/\text{FRACTN}}\right)^{(\text{IFREQ}-1)}$. IFREQ is the i^{th} frequency in the solution. FMAX is ignored.

VARIABLE	DESCRIPTION
	LT.0: FSPACE is a load curve ID for assigning active frequencies. The abscissa is frequencies in the solution and the ordinate is the active frequency in Hertz. FMIN and FMAX are ignored.
FRACTN	Octave fraction. For example, FRACTN = 3 means 1/3 octave spacing.

Remarks:

1. **Plot Files.** The complex solution is written to file d3ssd at every solution step. If *DATABASE_BINARY_PLOT output is also requested, then the real part of the solution is written to the d3plot file.
2. **Structural Loading.** Currently, the structural loads are identified with *LOAD_... keywords and limited to zero phase angle.
3. **Acoustics.** Acoustic elements (ELFORMs 8 and 14 on *SECTION_SOLID) with *MAT_ACOUSTIC as the material model are supported for *CONTROL_IMPLICIT_SSD_DIRECT analyses. The coupling of the acoustic fluid and the structural elements is achieved with *BOUNDARY_ACOUSTIC_COUPLING_MISMATCH or by merging acoustic and structural nodes with compatible element faces.

*CONTROL

*CONTROL_IMPLICIT_STABILIZATION

*CONTROL_IMPLICIT_STABILIZATION_{OPTION}

Available options include:

<BLANK>

DYN

SPR

Purpose: This optional card applies to implicit calculations. Artificial stabilization is required for multi-step unloading in implicit springback analysis (see also *CONTROL_IMPLICIT_GENERAL). The DYN option allows setting controls specifically for the dynamic relaxation phase. The SPR option allows setting controls specifically for the springback phase.

Card 1	1	2	3	4	5	6	7	8
Variable	IAS	SCALE	TSTART	TEND				
Type	I	F	F	F				
Default	2	1.0	↓	↓				

VARIABLE	DESCRIPTION
IAS	Artificial Stabilization flag (see Remark 2): EQ.1: active EQ.2: inactive (default)
SCALE	Scale factor for artificial stabilization (see Remark 3). For flexible parts with large springback, like outer body panels, a value of 0.001 may be required. EQ.-n: curve ID = n gives SCALE as a function of time
TSTART	Start time. (Default: immediately upon entering implicit mode)
TEND	End time. (Default: termination time)

Remarks:

1. **Artificial Stabilization.** Artificial stabilization allows springback to occur over several steps. This is often necessary to obtain convergence during equilibrium iterations on problems with large springback deformation. Stabilization is introduced at the start time TSTART, and slowly removed as the end time TEND is approached. Intermediate results are not accurate representations of the fully unloaded state. The end time TEND must be reached exactly for total springback to be predicted accurately.
2. **IAS Defaults.** The default for IAS depends on the analysis type in *CONTROL_IMPLICIT_GENERAL. For “seamless” springback analysis, automatic time step control and artificial stabilization are activated by default. Otherwise, IAS is inactive by default.
3. **Scale Factor.** SCALE is a penalty scale factor similar to that used in contact interfaces. If modified, it should be changed in order-of-magnitude increments at first. Large values suppress springback deformation until very near the termination time, making convergence during the first few steps easy. Small values may not stabilize the solution enough to allow equilibrium iterations to converge.

*CONTROL

*CONTROL_IMPLICIT_STATIC_CONDENSATION

*CONTROL_IMPLICIT_STATIC_CONDENSATION_{OPTION}

NOTE: As of R16, this keyword is obsolete. For SC_FLAG = 1, LS-DYNA automatically converts this keyword to *CONTROL_IMPLICIT_MODES. SC_FLAG = 2 leads to an error termination. To obtain the behavior of SC_FLAG = 2, compute the superelement with *CONTROL_IMPLICIT_MODES and then create a second run that uses that superelement.

Available options include:

<BLANK>

BINARY

Purpose: Request static condensation of a part to build a reduced linearized model for later computation with *ELEMENT_DIRECT_MATRIX_INPUT. Optionally, the analysis can continue using the linearization for the current analysis.

Note: Implicit Static Condensation is not supported in MPP. Please use *CONTROL_IMPLICIT_MODES.

Card Summary:

Card 1. This card is required.

SC_FLAG	SC_NSID	SC_PSID	SE_MASS	SE_STIFF	SE_INERT		
---------	---------	---------	---------	----------	----------	--	--

Card 1.1. Include this card if at least one of SE_MASS, SE_STIFF, or SE_INERT is not blank.

SE_FILENAME

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	SC_FLAG	SC_NSID	SC_PSID	SE_MASS	SE_STIFF	SE_INERT		
Type	I	I	I	A10	A10	A10		
Default	0	0	0	↓	↓	↓		

Card 1.1	1	2	3	4	5	6	7	8
Variable				SE_FILENAME				
Type				A80				

VARIABLE	DESCRIPTION
SC_FLAG	Static Condensation Control Flag: EQ.0: No static condensation will be performed. EQ.1: Create superelement representation based on static condensation. EQ.2: Use static condensation to build a linearized representation for a part and use that linearized representation in the following analysis.
SC_NSID	Node set ID for nodes to be preserved in the static condensation procedure. Required when SC_FLAG = 1.
SC_PSID	Part set ID for parts to be included in the static condensation procedure. When SC_FLAG = 1, SC_PSID can be used to specify a subset of the model with the default being the entire model. When SC_FLAG = 2, SC_PSID is required. SC_PSID = 0 implies that the entire model is condensed.
SE_MASS	Name of the superelement mass matrix. If left blank, it is not generated.
SE_STIFF	Name of the superelement stiffness matrix. If left blank, it is not generated.

*CONTROL

*CONTROL_IMPLICIT_STATIC_CONDENSATION

VARIABLE	DESCRIPTION
SE_INERT	Name of the superelement inertia matrix, required for gravity loading applications of the superelement. If left blank, it is not generated.
SE_FILENAME	If any of SE_MASS, SE_STIFF, or SE_INERT is blank, then the second line is required and contains the file name for the superelement.

Remarks:

1. **Input and output files.** To use this feature, an implicit analysis must be requested using IMFLAG = 1 on *CONTROL_IMPLICIT_GENERAL, and a non-zero termination time must be specified on *CONTROL_TERMINATION. A double-precision version of LS-DYNA should be used for best accuracy. The superelement model is written to file SE_FILENAME.
2. **Modes.** Static condensation is the reduction of the global stiffness and mass matrices to a specified set of rows and columns associated with the nodes in the node set SC_NSID. The first and second node set attribute parameters can be optionally used to restrict the translational and rotational degrees of freedom for which modes are requested, respectively, according to the following syntax:
 - a) Node set attribute parameters DA1 and A1: translational degree of freedom codes
 - b) Node set attribute parameters DA2 and A2: rotational degree of freedom codes

Code	Modes Computed
0	(See note below.)
1	X degree of freedom only
2	Y degree of freedom only
3	Z degree of freedom only
4	X, Y degrees of freedom only
5	Y, Z degrees of freedom only
6	X, Z degrees of freedom only
7	X, Y, Z degrees of freedom

Setting both node set attributes to zero is equivalent to setting both node set attributes to 7 (X, Y, and Z for translational and rotational degrees of freedom).

If one node set attribute is nonzero (codes 1 to 7) and the other node set attribute is zero, then the zero attribute means NO degrees of freedom are considered. For example, if DA1 = 2 and DA2 = 0, then only the Y-translational degree of freedom modes are calculated.

3. **Superelements.** The user can create the superelement representation of the reduced model by specifying the SE_MASS, SE_STIFF, SE_INERT and SE_FILENAME fields. This implementation does not include SE_DAMP. The file, by default is written in the Nastran DMIG file format and can be used as input to [***ELEMENT_DIRECT_MATRIX_INPUT**](#). The keyword option BINARY can be used to create a binary representation for the superelement that can be used with [***ELEMENT_DIRECT_MATRIX_INPUT_BINARY**](#) to reduce the file size.
4. **Comparison to *CONTROL_IMPLICIT_MODES.** Static condensation is equivalent to using only constraint modes with [***CONTROL_IMPLICIT_MODES**](#). Static condensation does have the ability to continue the analysis using the linear representation for a part set.

*CONTROL

*CONTROL_IMPLICIT_TERMINATION

*CONTROL_IMPLICIT_TERMINATION

Purpose: Specify termination criteria for implicit transient simulations.

Card 1	1	2	3	4	5	6	7	8
Variable	DELTAU	DELT A1	KETOL	IETOL	TETOL	NSTEP	ABSTOL	
Type	F	F	F	F	F	I	F	
Default	0.0	0.0	0.0	0.0	0.0	3	0.0	

VARIABLE	DESCRIPTION
DELTAU	Terminate based on relative total displacement in the Euclidean norm. GT.0.0: terminate when displacement in the Euclidean norm for last time step relative to the total displacement in the Euclidean norm is less than DELTAU.
DELT A1	Terminate based on relative total displacement in the max norm. GT.0.0: terminate when displacement in the max norm for last time step relative to the total displacement in the max norm is less than DELTAU.
KETOL	Terminate based on kinetic energy GT.0.0: terminate when kinetic energy drops below KETOL for NSTEP consecutive implicit time steps.
IETOL	Terminate based on internal energy GT.0.0: terminate when internal energy drops below IETOL for NSTEP consecutive implicit time steps.
TETOL	Terminate based on total energy GT.0.0: terminate when total energy drops below TETOL for NSTEP consecutive implicit time steps.
NSTEP	Number of steps used in the early termination tests for kinetic, internal, and total energy.

VARIABLE	DESCRIPTION
ABSTOL	Terminate based on absolute total displacement in the Euclidean norm. GT.0.0: terminate when displacement in the Euclidean norm for last time step is less than ABSTOL.

Remarks:

For some implicit applications it is useful to terminate when there is no change in displacement or low energy. This keyword provides the ability to specify such a stopping criteria to terminate the simulation prior to ENDTIM.

*CONTROL

*CONTROL_LSDA

*CONTROL_LSDA

This keyword is used to globally control data present in included dynain.lsda files generated by *INTERFACE_SPRINGBACK_LSDYNA with FTTYPE = 3. The primary function is to provide the flexibility of not having to know *a priori* how to define *INTERFACE_SPRINGBACK but be able to modify this *a posteriori* without having to rerun the first simulation. For examples, see remarks.

Card 1	1	2	3	4	5	6	7	8
Variable	NPEXCL							
Type	I							
Default	0							

Excluded Parts Card. If NPEXCL $\neq 0$, include as many of this card as required to exclude NPEXCL part IDs.

Card 1.1	1	2	3	4	5	6	7	8
Variable	P1	P2	P3	P4	P5	P6	P7	P8
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
NPEXCL	Number of parts to exclude from the dynain.lsda files
P1 - P8	Part IDs to exclude from dynain.lsda

Remarks:

1. **Excluding Parts Example.** Assume a forming simulation is performed with 2 deformable parts (say part 1 and part 2) and both are for some reason included in the resulting dynain.lsda. In a later stage, a springback of only part 1 is desired without the need of part 2. Instead of rerunning the forming simulation and outputting only part 1 to dynain.lsda, the springback can be performed by excluding part 2 from the springback simulation. This is done through NPEXCL = 1 and P1 = 2.

***CONTROL_MAPPING_SHIFT_BY_DT**

Purpose: For [*INITIAL_ALE_MAPPING](#) and [*INITIAL_LAG_MAPPING](#), two subsequent models are run with the second run initialized by data from the last cycle of the first run. This keyword shifts the initial time of the second run by the time step read from the mapping file, meaning the final time step of the first run.

*CONTROL

*CONTROL_MAT

*CONTROL_MAT

Purpose: Define global control parameters for properties related to the material model.

Card 1	1	2	3	4	5	6	7	8
Variable	MAEF		UMCHK	OLDINTP				
Type	I		I	I				
Default	0		0	0				

VARIABLE	DESCRIPTION
MAEF	Failure options: EQ.0: All *MAT_ADD_EROSION, *MAT_ADD_DAMAGE_Diem, and *MAT_ADD_DAMAGE_GISSMO definitions are active. EQ.1: Switch off all *MAT_ADD_EROSION, *MAT_ADD_DAMAGE_Diem, and *MAT_ADD_DAMAGE_GISSMO definitions globally. This feature is useful for larger models where removing those cards is inconvenient.
UMCHK	User material check. In the first calculation cycle, LS-DYNA checks if user-defined material models are applied or whether only the default, unmodified subroutines already present in the native dyn21 files are called. It also works for user-defined friction in MPP. EQ.0: Warning is issued if only unmodified subroutines are called. EQ.1: Error termination occurs if only unmodified subroutines are called.
OLDINTP	Because of changes in results when correcting the interpolation of tables in the piecewise plasticity model (*MAT_024), you can set this flag to 1 to invoke the old behavior. Note that the old behavior may result in an incorrect stress response, so we generally do not recommend it.

***CONTROL_MPP**

Purpose: Set control parameters for MPP-specific features.

All options with DECOMPOSITION in the name control the distribution of the model to the processors. However, these keywords do not affect the decomposition of seat-belt-related elements because these types of elements are decomposed in an unalterable, pre-determined way. All seat belt elements in a single seat belt are allocated to a single processor. For a model with more than one seat belt, the decomposition algorithm assigns each seat belt to a different processor if available.

- *CONTROL_MPP_CONTACT_GROUPABLE**
- *CONTROL_MPP_DECOMPOSITION_ADJUST_MTYPE_COST**
- *CONTROL_MPP_DECOMPOSITION_ADJUST_PART_COST**
- *CONTROL_MPP_DECOMPOSITION_ARRANGE_PARTS**
- *CONTROL_MPP_DECOMPOSITION_AUTOMATIC**
- *CONTROL_MPP_DECOMPOSITION_BAGREF**
- *CONTROL_MPP_DECOMPOSITION_CHECK_SPEED**
- *CONTROL_MPP_DECOMPOSITION_CONTACT_DISTRIBUTE**
- *CONTROL_MPP_DECOMPOSITION_CONTACT_ISOLATE**
- *CONTROL_MPP_DECOMPOSITION_DEFORMED_GEOMETRY**
- *CONTROL_MPP_DECOMPOSITION_DISABLE_UNREF_CURVES**
- *CONTROL_MPP_DECOMPOSITION_DISTRIBUTE_ALE_ELEMENTS**
- *CONTROL_MPP_DECOMPOSITION_DISTRIBUTE_SPH_ELEMENTS**
- *CONTROL_MPP_DECOMPOSITION_ELCOST**
- *CONTROL_MPP_DECOMPOSITION_FILE**
- *CONTROL_MPP_DECOMPOSITION_FLAG_STRESS_STRAIN_CURVE**
- *CONTROL_MPP_DECOMPOSITION_METHOD**
- *CONTROL_MPP_DECOMPOSITION_NODISTRIBUT DES_ELEMENTS**

***CONTROL**

***CONTROL_MPP**

*CONTROL_MPP_DECOMPOSITION_NUMPROC
*CONTROL_MPP_DECOMPOSITION_OUTDECOMP
*CONTROL_MPP_DECOMPOSITION_PARTS_DISTRIBUTE
*CONTROL_MPP_DECOMPOSITION_PARTSET_DISTRIBUTE
*CONTROL_MPP_DECOMPOSITION_RCBLOG
*CONTROL_MPP_DECOMPOSITION_REDECOMPOSITION
*CONTROL_MPP_DECOMPOSITION_SCALE_CONTACT_COST
*CONTROL_MPP_DECOMPOSITION_SCALE_FACTOR_SPH
*CONTROL_MPP_DECOMPOSITION_SHOW
*CONTROL_MPP_DECOMPOSITION_TRANSFORMATION
*CONTROL_MPP_IO_LSTC_REDUCE
*CONTROL_MPP_IO_NOBEAMOUT
*CONTROL_MPP_IO_NOD3DUMP
*CONTROL_MPP_IO_NODUMP
*CONTROL_MPP_IO_NOFULL
*CONTROL_MPP_IO_NOTIEDIO
*CONTROL_MPP_IO_SWAPBYTES
*CONTROL_MPP_MATERIAL_MODEL_DRIVER
*CONTROL_MPP_PFILE
*CONTROL_MPP_REBALANCE

***CONTROL_MPP_CONTACT_GROUPABLE**

Purpose: Allow for global specification that the GROUPABLE algorithm should be enabled/disabled for contacts when running MPP.

Card 1	1	2	3	4	5	6	7	8
Variable	GRP							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
GRP	<p>The sum of these available options (in any combination that makes sense):</p> <ul style="list-style-type: none"> EQ.1: Turn on GROUPABLE for all non-tied contacts. This flag does not apply to segment-to-segment contact (SOFT = 2). EQ.2: Turn on GROUPABLE for all tied contacts EQ.4: Turn off GROUPABLE for all non-tied contacts. This flag does not apply to segment-to-segment contact (SOFT = 2). EQ.8: Turn off GROUPABLE for all tied contacts. EQ.16: Turn on GROUPABLE for segment-to-segment contact (SOFT = 2). <p>GRP overrides the setting of GRPABLE on the MPP Cards of *CONTACT_..._MPP for GRPABLE ≥ 0.</p>

Remarks:

1. **Availability.** The groupable algorithm is an alternate MPP communication algorithm for various single surface (including AUTOMATIC_GENERAL), NODES_TO_SURFACE, SURFACE_TO_SURFACE, ERODING, and option SOFT = 2 contacts. This groupable algorithm does not support all contact options, including MORTAR, and is still under development. It can be significantly faster and scale better than the normal algorithm when there are more than two or three applicable contact types defined in the model. Its intent is to speed up the contact processing but not to change the behavior of the contact.

2. **Priority of this keyword.** This keyword overrides GRPABLE on the MPP cards of *CONTACT_..._MPP for $\text{GRPABLE} \geq 0$. It is intended as a way to quickly experiment with this feature. The equivalent pfile option is “contact { groupable GRP }” where GRP is an integer as described above.
3. **Shell reference surface for tied contacts in MPP.** Unless the MPP contact is groupable (see GRPABLE on the MPP cards of *CONTACT_..._MPP and GRP here), MPP tied contacts ignore the influence of NLOC on *SECTION_SHELL, OFFSET on *ELEMENT_SHELL_OFFSET, CNTCO on *CONTROL_SHELL, and SHLOFF on [Optional Card G](#) on the shell reference surface. Non-groupable tied contacts always use the plane determined by the nodes for the shell reference surface. Thus, these fields for non-groupable tied contacts have no effect on determining what is tied or not. Groupable tied contacts honor the settings of these fields.

CONTROL_MPP_DECOMPOSITION_ADJUST_MTYPE_COST *CONTROL**CONTROL_MPP_DECOMPOSITION_ADJUST_MTYPE_COST**

Purpose: Modify the decomposition cost of elements according to their material type.

Any number of cards may appear with one material per card.

Card 1	1	2	3	4	5	6	7	8
Variable	TYPEID	A	B					
Type	I	F	F					
Default	none	1.0	0.0					

VARIABLE	DESCRIPTION
TYPEID	Material type ID as a number. For example, 1 here indicates *MAT_ELASTIC.
A	Cost multiplier
B	Added cost

Remarks:

1. **Adjusting computational cost.** During decomposition, all elements having the specified material type have their computational cost adjusted according to this formula:

$$\text{new cost} = A \times (\text{original cost}) + B$$

2. **Same material type in more than one card.** If a material appears in more than one instance of this card, only the first instance is used, and the others are ignored.

CONTROL**CONTROL_MPP_DECOMPOSITION_ADJUST_PART_COST*****CONTROL_MPP_DECOMPOSITION_ADJUST_PART_COST**

Purpose: Modify the decomposition cost of elements according to their part ID.

Any number of cards may appear with one part per card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	A	B					
Type	I	R	R					
Default	none	1.0	0.0					

VARIABLE	DESCRIPTION
PID	Part ID
A	Cost multiplier
B	Added cost

Remarks:

1. **Adjusting computational cost.** During decomposition, all elements belonging to the specified part ID have their computational cost adjusted according to this formula:

$$\text{new cost} = A \times (\text{original cost}) + B$$

2. **Same part in more than one card.** If a part appears in more than one instance of this card, only the first instance is used, and the others are ignored. Any values given here override conflicting data entered with [*CONTROL_MPP_DECOMPOSITION_ADJUST_MTYPE_COST](#) because this keyword is more specific.

***CONTROL_MPP_DECOMPOSITION_ARRANGE_PARTS_{OPTION}**

Purpose: Allow users to distribute certain part(s) to all processors or to isolate certain part(s) in a single processor. This keyword supports multiple entries. Each entry is be processed as a separate region for decomposition.

When this keyword is part of an included file and the LOCAL option is given, the decomposition will be done in the coordinate system of the included file, which may be different from the global system if the file is included using the *INCLUDE_TRANSFORM keyword.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	TYPE	NPROC	FRSTP	NPTOT			
Type	I	I	I	I	I			
Default	none	none	none	none	0			

VARIABLE	DESCRIPTION
ID	Part ID/part set ID
TYPE	EQ.0: part ID to be distributed to all processors EQ.1: part set ID to be distributed to all processors EQ.10: part ID to be lumped into one processor EQ.11: part Set ID to be lumped into one processor EQ.20: part ID to be lumped into one processor with MPP load balanced EQ.21: part set ID to be lumped into one processor with MPP load balanced.
NPROC	Used only for TYPE equal to 0 or 1. Number of processors that will be used for decomposition. This part ID/part set ID will be distributed to NPROC of the processors.
FRSTP	Used only for TYPE equal to 0 or 1. Starting MPP rank ID (rank starts from 0).

*CONTROL

*CONTROL_MPP_DECOMPOSITION_ARRANGE_PARTS

VARIABLE	DESCRIPTION
NPTOT	Total number of processors for (NPROC, FRSTP): EQ.0: use current numproc. GT.0: if <code>max(numproc,numproc_dc)</code> is different than NPTOT, (NPROC, FRSTP) will be rescaled by NPTOT. This feature allows the same input to be used with changing computing resources.

Remarks:

Since this keyword supports multiple entries, each line will be treated as a region for the decomposition. Therefore, the equivalent method using a pfile will be different depending on TYPE and whether entity being decomposed is a part or parts set as shown in the table below.

TYPE	PFILE EQUIVALENT
0	<code>region { parts PID nproc NPROC FRSTP }</code>
1	<code>region { partsets PSID nproc NPROC FRSTP }</code>
10	<code>region { parts PID lumped }</code>
11	<code>region { partsets PSID lumped }</code>
20	<code>region { parts PID together }</code>
21	<code>region { partsets PSID together }</code>

***CONTROL_MPP_DECOMPOSITION_AUTOMATIC**

Purpose: Instructs the program to apply a simple heuristic to try to determine the proper decomposition for the simulation.

There are no input parameters. The existence of this keyword triggers the automated decomposition. This option should not be used if there is more than one occurrence of any of the following options in the model:

*INITIAL_VELOCITY

*CHANGE_VELOCITY

*BOUNDARY_PRESCRIBED_MOTION

And the following control card must not be used:

*CONTROL_MPP_DECOMPOSITION_TRANSFORMATION

For the general case, it is recommended that you specify the proper decomposition using the command *CONTROL_MPP_DECOMPOSITION_TRANSFORMATION instead.

*CONTROL

*CONTROL_MPP_DECOMPOSITION_BAGREF

*CONTROL_MPP_DECOMPOSITION_BAGREF

Purpose: With this card LS-DYNA performs decomposition according to the airbag's reference geometry, rather than the folded geometry.

Other than BAGID values this card takes no input parameters. The initial geometry may lead to a poor decomposition once the bag is deployed. This option will improve load balancing for the fully deployed geometry.

Optional card(s) for selected reference geometry ID

Card 1	1	2	3	4	5	6	7	8
Variable	BAGID1	BAGID2	BAGID3	BAGID4	BAGID5	BAGID6	BAGID7	BAGID8
Type	I	I	I	I	I	I	I	I
Default	none							

VARIABLE	DESCRIPTION
BAGID i	ID defined in *AIRBAG_REFERENCE_GEOMETRY_ID or *AIRBAG_SHELL_REFERENCE_GEOMETRY_ID

Bags specified in the optional cards will be decomposed based on the reference geometry. If there is no card given, *all* bags will be decomposed by their reference geometry.

Remarks:

Command in partition file (pfile): BAGREF. The option for selecting particular airbags is only available when using keyword input.

***CONTROL_MPP_DECOMPOSITION_CHECK_SPEED**

Purpose: Modifies the decomposition depending on the relative speed of the processors involved.

There are no input parameters. Use of this keyword activates a short floating point timing routine to be executed on each processor. The information gathered is used during the decomposition, with faster processors being given a relatively larger portion of the problem. This option is not recommended on homogeneous systems.

***CONTROL *CONTROL_MPP_DECOMPOSITION_CONTACT_DISTRIBUTE**

***CONTROL_MPP_DECOMPOSITION_CONTACT_DISTRIBUTE_{OPTION}**

Purpose: Ensures that the indicated contact interfaces are distributed across all processors, which can lead to better load balance for large contact interfaces. If this appears in an included file and the LOCAL option is used, the decomposition will be done in the coordinate system of the included file, which may be different from the global system if the file is included through *INCLUDE_TRANSFORM.

Card 1	1	2	3	4	5	6	7	8
Variable	ID1	ID2	ID3	ID4	ID5			
Type	I	I	I	I	I			
Default	none	none	none	none	none			

VARIABLE	DESCRIPTION
ID1	First contact interface ID to distribute. If no contact ID's are specified, the number given here corresponds to the order of the interfaces as they appear in the input, with the first being 1.
ID2, ID3, ID4, ID5	Remaining interfaces ID's to distribute.

Remarks:

Up to 5 contact interface ID's can be specified. The decomposition is modified as follows: First, all the elements involved in the first contact interface are decomposed across all the processors. Then all the elements involved in the second contact interface (excluding any already assigned to processors) are distributed, and so on. After all the contact interfaces given are processed, the rest of the input is decomposed in the normal manner. This will result in each processor having possibly several disjoint portions of the input assigned to it, which will increase communications somewhat. However, this can be offset by improved load balance for the contact. It is generally recommended that at most one or two interfaces be specified, and then only if they are of substantial size relative to the whole problem.

***CONTROL_MPP_DECOMPOSITION_CONTACT_ISOLATE**

Purpose: Ensures that the indicated contact interfaces are isolated on a single processor, which can lead to decreased communication.

Card 1	1	2	3	4	5	6	7	8
Variable	ID1	ID2	ID3	ID4	ID5			
Type	I	I	I	I	I			
Default	none	none	none	none	none			

VARIABLE	DESCRIPTION
ID1	First contact interface ID to isolate. If no contact ID's are specified, the number given here corresponds to the order of the interfaces as they appear in the input, with the first being 1.
ID2, ID3, ID4, ID5	Remaining interfaces ID's to isolate.

Remarks:

Up to 5 contact interfaces can be specified. The decomposition is modified as follows: First, all the elements involved in the first contact interface ID are assigned to the first processor. Then all the elements involved in the second contact interface ID (excluding any already assigned to processors) are assigned to the next processor, and so on. After all the contact interfaces given are processed, the rest of the input is decomposed in the normal manner. This will result in each of the interfaces being processed on a single processor. For small contact interfaces this can result in better parallelism and decreased communication.

***CONTROL *CONTROL_MPP_DECOMPOSITION_DEFORMED_GEOMETRY**

***CONTROL_MPP_DECOMPOSITION_DEFORMED_GEOMETRY**

Purpose: Because of distortion due to deformation or changing contacts due to rotation, a new decomposition of the deformed geometry may need to be performed during a full deck restart for better load balancing. This option causes the new load balance. This feature may be useful, for instance, with bird strike since the contacts change as the fan blades rotate.

SPH Card. This card is optional.

Card 1	1	2	3	4	5	6	7	8
Variable	SPH							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
SPH	Flag for inclusion of inactive SPH elements: EQ.0: Exclude inactive SPH elements in the new decomposition (default). EQ.1: Include inactive SPH elements in the new decomposition.

Remarks:

Command in partition file (pfile): decomposition { defgeo } or decomposition { defgeo 1 }.

***CONTROL_MPP_DECOMPOSITION_DISABLE_UNREF_CURVES *CONTROL**

***CONTROL_MPP_DECOMPOSITION_DISABLE_UNREF_CURVES**

Purpose: Disable unreferenced time dependent load curves for the following keywords:

*BOUNDARY_PRESCRIBED_MOTION_NODE

*LOAD_NODE

*LOAD_SHELL_ELEMENT

*LOAD_THERMAL_VARIABLE_NODE

The details of this operation are reported in each processor's scratch scr#### file. This will skip the curve evaluation on each cycle, and improve the parallel efficiency.

Remarks:

Command in partition file (pfile): DUNREFLC.

*CONTROL

*CONTROL_MPP_DECOMPOSITION_DISTRIBUTE_ALE_ELEMENTS

*CONTROL_MPP_DECOMPOSITION_DISTRIBUTE_ALE_ELEMENTS

Purpose: Ensure ALE elements are evenly distributed to all processors.

There are no input parameters and the card below is optional. ALE elements usually have a larger computational cost than other element types, and it is better to distribute them to all CPUs for better load balance. This keyword causes DYNA/MPP to extract ALE parts from the input and then evenly distribute them to all the processors.

FSI/ALE Mesh Card. This card is optional.

Card 1	1	2	3	4	5	6	7	8
Variable	OVERLAP							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
OVERLAP	For FSI models where structures are inside ALE meshes (see *CONSTRAINED_LAGRANGE_IN_SOLID), flag to decompose the structure and ALE domains together instead of first the structure and then the ALE (see Remark 2).
	EQ.0: Off
	EQ.1: On

Remarks:

1. **Partition file.** Command in partition file (pfile): ALEDIST.
2. **ALE/FSI overlap.** Most of the processors will have to deal with MPP subdomains from the structure and ALE meshes: a portion of the ALE computational domain and a portion of the structure meshes. The default decomposition (first divide the structures, then ALE) does not always overlap these subdomains. The more they overlap, the less the MPP communications due to the coupling cost. Cutting the ALE and structure meshes together allows their MPP subdomains to be as inclusive as possible.

***CONTROL_MPP_DECOMPOSITION_DISTRIBUTE_SPH_ELEMENTS**

Purpose: Ensures SPH elements are evenly distributed to all processors

There are no input parameters. SPH elements usually have higher computational cost than other type of elements and it is better to distribute them to all CPU for better load balance. The existence of this keyword causes DYNA/MPP to extract SPH parts from input and then evenly distributed to all processors.

Remarks:

Command in partition file (pfile): SPHDIST.

***CONTROL**

***CONTROL_MPP_DECOMPOSITION_ELCOST**

***CONTROL_MPP_DECOMPOSITION_ELCOST**

Purpose: Instructs the program to use a hardware specific element cost weighting for the decomposition

Card 1	1	2	3	4	5	6	7	8
Variable	ITYPE							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
ITYPE	<p>Hardware specific cost profile.</p> <p>EQ.1: Fujitsu PrimePower</p> <p>EQ.2: Intel IA 64, AMD Opteron</p> <p>EQ.3: Intel Xeon 64</p> <p>EQ.4: General profile</p>

Remarks:

Command in partition file (pfile): elcost itype.

***CONTROL_MPP_DECOMPOSITION_FILE_{OPTION}**

Purpose: Allow for pre-decomposition and a subsequent run or runs without having to do the decomposition.

OPTION specifies the action type. The following options are accepted:

<BLANK>

READ

WRITE

When the keyword option READ is used, LS-DYNA expects you to have placed the pre-decomposition file in the working directory. The job will be terminated if the file is not there. When WRITE is used, LS-DYNA will create the file. The job will be terminated if the pre-decomposition file is in the working directory.

Card 1	1	2	3	4	5	6	7	8
Variable				NAME				
Type				A80				
Default				none				

VARIABLE**DESCRIPTION**

NAME Name of a file containing (or to contain) a decomposition record.

Remarks:

If the indicated file does not exist, it is created with a copy of the decomposition information from this run. If the file exists, it is read and the decomposition steps can be skipped. The original run that created the file must be for a number of processors that is a multiple of the number of processors currently being used. Thus, a problem can be decomposed once for, say, 48 processors. Subsequent runs are then possible on any number that divides 48: 1, 2, 3, 4, 6, etc. Since the decomposition phase generally requires more memory than execution, this allows large models to be decomposed on one system and run on another (provided the systems have compatible binary formats).

***CONTROL**

***CONTROL_MPP_DECOMPOSITION_FLAG_STRESS_STRAIN_CURVE**

***CONTROL_MPP_DECOMPOSITION_FLAG_STRESS_STRAIN_CURVE**

Purpose: Flag to skip evaluating non-time dependent stress strain curves on every cycle to save CPU time. The stress strain curves will only be evaluated as needed in the material routines. Currently, this keyword only affects curves used with *MAT_024.

Remarks:

Command in partition file (pfile): SORTSSLC.

***CONTROL_MPP_DECOMPOSITION_METHOD**

Purpose: Specify the decomposition method to use.

Card 1	1	2	3	4	5	6	7	8
Variable					NAME			
Type					A80			
Default					RCB			

VARIABLE**DESCRIPTION**

NAME

Name of the decomposition method to use. There are currently two options:

EQ.“RCB”: recursive coordinate bisection

EQ.“GREEDY”: a simple heuristic method

In almost all cases the RCB method is superior and should be used.

***CONTROL**

***CONTROL_MPP_DECOMPOSITION_NODISTRIBUTEDES_ELEMENTS**

***CONTROL_MPP_DECOMPOSITION_NODISTRIBUTEDES_ELEMENTS**

Purpose: By default, DES elements are treated as a single region and distributed to all processors. For traditional MPP execution in which LS-DYNA performs a single decomposition at the beginning of the run, the default treatment of distributing to all processors yields better load balancing. For MPP that includes repartition of the model during execution with *CONTROL_MPP_DECOMPOSITION_REDECOMPOSITION, putting DES elements and couple FE elements on the same processor leads to better performance. This keyword is for the second case. It disables the default distribution and causes current geometry to be used for decomposition.

There are no input parameters.

Remarks:

Command in partition file (pfile): DESNODIST.

***CONTROL_MPP_DECOMPOSITION_NUMPROC**

Purpose: Specify the number of processors for decomposition.

Card 1	1	2	3	4	5	6	7	8
Variable	N							
Type	I							
Default	none							

VARIABLE**DESCRIPTION**

N Number of processors for decomposition.

Remarks:

This is used in conjunction with the CONTROL_MPP_DECOMPOSITION_FILE command to allow for later runs on different numbers of processors. By default, the decomposition is performed for the number of processors currently being used. However, a different value can be specified here. If N > 1 and only one processor is currently being used, the decomposition is done and then the program terminates. If N is *not* a multiple of the current number of processors, then it is ignored the execution proceeds with the current number of processors. Otherwise, the decomposition is performed for N processors, and the execution continues using the current number of processors.

*CONTROL

*CONTROL_MPP_DECOMPOSITION_OUTDECOMP

*CONTROL_MPP_DECOMPOSITION_OUTDECOMP

Purpose: Instructs the program to output element's ownership data to file for post-processor to show state data from different processors.

Card 1	1	2	3	4	5	6	7	8
Variable	TYPE							
Type	I							
Default	none							

VARIABLE	DESCRIPTION
ITYPE	Sets the format for the output file. EQ.1: database in LS-PrePost format: <code>decomp_parts.lsprepost</code> EQ.2: database in animator format: <code>decomp_parts.ses</code> EQ.3: database in LS-PrePost format with d3plot state number. This file allows LS-PrePost to show the matching d3plot with the decomposition for *CONTROL_MPP_DECOMPOSITION_REDECOMPOSITION <code>decomp_parts.lsprepost_s#####</code>

Remarks:

1. **PFILE.** Command in partition file (pfile): OUTDECOMP ITYPE.
2. **LS-PrePost.** When ITYPE is set to 1, the elements assigned to any particular core can be viewed and animated by LS-PrePost by (1) reading the d3plot data, and then (2) selecting *Models* → *Views* → *MPP* → *Load* → *decomp_parts.lsprepost*.

***CONTROL_MPP_DECOMPOSITION_PARTS_DISTRIBUTE_{OPTION}**

Purpose: Distribute the parts given by this keyword to all processors before the decomposition for the rest of the model is performed.

If this appears in an included file and the **LOCAL** option is used, the decomposition will be done in the coordinate system of the included file, which may be different from the global system if the file is included via ***INCLUDE_TRANSFORM**.

Card 1	1	2	3	4	5	6	7	8
Variable	ID1	ID2	ID3	ID4	ID5	ID6	ID7	ID8
Type								
Default	none							

VARIABLE	DESCRIPTION
ID1, ID2, ID3, ...	<p>For each ID:</p> <p>GT.0: ID is a part number.</p> <p>LT.0: ID is a part set number.</p> <p>All parts defined in this card will be treated as a single region to be decomposed.</p>

Remarks:

- Decomposition.** Up to 1000 parts/part sets can be specified. The decomposition is modified as follows: the elements involved in the given parts are put into a separate domain from rest of the model and then distributed to all processors to balance their computational cost. Then the remainder of the model will be distributed in the usual way.
- PFILE.** This is equivalent to the pfile command (for example, if ID1 - ID3 are part IDs and ID4 - ID6 are part set IDs):

```
decomp { region { parts ID1 ID2 ID3 or partsets ID4 ID5 ID6 } }
```

The part set IDs, however, are positive when used in the pfile.

CONTROL**CONTROL_MPP_DECOMPOSITION_PARTSET_DISTRIBUTE*****CONTROL_MPP_DECOMPOSITION_PARTSET_DISTRIBUTE_{OPTION}**

Purpose: Distribute the part sets given in this option to all processors before the decomposition for the remainder of the model is performed.

If this appears in an included file and the **LOCAL** option is given, the decomposition will be done in the coordinate system of the included file, which may be different from the global system if the file is included via ***INCLUDE_TRANSFORM**.

Card 1	1	2	3	4	5	6	7	8
Variable	ID1	ID2	ID3	ID4	ID5	ID6	ID7	ID8
Type								
Default	none							

VARIABLE	DESCRIPTION
ID1, ID2, ID3, ...	Part Set ID to be distributed. All parts in ID1 will be shared across all processors. Then all parts in ID2 will be distributed, and so on.

Remarks:

Any number of part sets can be specified. Each part set is distributed across all processors, in the order given. The order may be significant if, in particular, a part ID is in more than one set. Distribution of these parts is done before any decomposition specifications given in the pfile.

***CONTROL_MPP_DECOMPOSITION_RCBLOG**

Purpose: Causes the program to record decomposition information in the indicated file, for use in subsequent analyses.

Card 1	1	2	3	4	5	6	7	8
Variable				FILENAME				
Type				A80				
Default				none				

VARIABLE**DESCRIPTION**

FILENAME

Name of output file where decomposition history will be recorded.
This file can be used as the pfile for later analyses.

Remarks:

Command in parallel option file (pfile): rcblog filename.

CONTROL**CONTROL_MPP_DECOMPOSITION_REDECOMPOSITION*****CONTROL_MPP_DECOMPOSITION_REDECOMPOSITION_{OPTION}**

Available options are:

<BLANK>

ONCE

ONCE causes this keyword to be disabled after the model has been redecomposed based on information collected from the initial step.

Purpose: Redecompose a model at a given time interval. Other decomposition directives given in a pfile or defined using *CONTROL_MPP_DECOMPOSITION_OPTION remain in effect for each redecomposition. Redecomposition is intended to improve computational efficiency in simulations where large relative displacements between nodes develop and change as the simulation progresses.

Redecomposition relies on LS-DYNA's full deck restart capability and therefore will work correctly only to the extent that a full deck restart works correctly. We advise examining results carefully to ensure that history variable values, such as stress values, present before redecomposition are present after redecomposition.

Card 1	1	2	3	4	5	6	7	8
Variable	FREQ	DEFGEO	WEIGHT	REMSPH	STIME	SAMPT		
Type	F	I	F	I	F	F		
Default	none	1	1.0	0	0.0	None		

VARIABLE**DESCRIPTION**

FREQ

Determines the number of redecompositions during the solution:

LT.0: |FREQ| rounded to the nearest integer is the number of redecompositions during the solution.

GT.0: FREQ is the time interval between redecompositions.

DEFGEO

Geometry for decomposition:

EQ.1: Use current geometry for decomposition. When applied to a model containing SPH, deactivated SPH elements are not considered in the partition. This will give

VARIABLE	DESCRIPTION
	better load balancing if SPH elements are deleted during the simulation. EQ.2: Use current geometry for decomposition (same as 1 if applied to a non-SPH model). When applied to a model containing SPH, all SPH elements are considered in the partition. This will give better load balancing if SPH elements are reactivated during the simulation.
WEIGHT	Element cost scale factor for an element in contact or an element that has undergone plastic strain. If the element is under contact and has plastic strain, the weight will be doubled. Since the element cost is measured from calculated quantities, the results will remain consistent between runs with the same input and decomposition unlike using SAMPT below.
REMSPH	Flag to remove deactivated SPH particles: EQ.0: Keep deactivated particles. EQ.1: Remove deactivated particles. EQ.2: Remove deactivated particles. Also, particles generated by the *DEFINE_SPH_MESH_BOX keyword are fragmented and added to the input file as needed during each redecomposition phase.
STIME	Start time for redecomposition
SAMPT	Time interval for collecting element cost profile to use in the next REDECOMP step. Since the cost profile is based on the measurement of elapsed time, the timing profile may change between runs. It may produce different numerical results from multiple runs using identical input. GT.0: Sampling from beginning of each redecomposition for length SAMPT (t to $t + \text{SAMPT}$). If $\text{SAMPT} \geq \text{FREQ}$, then the sampling will occur for the entire time interval, FREQ. LT.0: Sampling from before ending of each redecomposition through to the next redecomposition ($t + \text{FREQ} - \text{SAMPT}$ to $t + \text{FREQ}$)

***CONTROL *CONTROL_MPP_DECOMPOSITION_SCALE_CONTACT_COST**

***CONTROL_MPP_DECOMPOSITION_SCALE_CONTACT_COST**

Purpose: Instructs the program to apply a scale factor to the list of contacts to change the partition weight for the decomposition.

Card 1	1	2	3	4	5	6	7	8
Variable	SF	ID1	ID2	ID3	ID4	ID5	ID6	ID7
Type	F	I	I	I	I	I	I	I
Default	none							

VARIABLE	DESCRIPTION
SF	Scale factor for the contact segments listed in the interface ID.
ID1, ID2, ...	interfaces ID's to be considered for scaling. Include second card if necessary.

Remarks:

Up to 15 contact interfaces ID can be specified. The decomposition is modified by applying this scale factor to the default computational cost of elements for the given contact interface ID.

Command in partition file (pfile): CTCOST ID1, ID2, ..., SF.

***CONTROL_MPP_DECOMPOSITION_SCALE_FACTOR_SPH**

Purpose: Instructs the program to apply a scale factor to SPH elements to change the partition weight for the decomposition.

Card 1	1	2	3	4	5	6	7	8
Variable	SF							
Type	F							
Default	none							

VARIABLE	DESCRIPTION
SF	Scale factor

Remarks:

Command in partition file (pfile): SPHSF SF.

***CONTROL**

***CONTROL_MPP_DECOMPOSITION_SHOW**

***CONTROL_MPP_DECOMPOSITION_SHOW**

Purpose: The keyword writes the final decomposition to the d3plot database. There are no input parameters.

This keyword causes MPP LS-DYNA to terminate immediately after the decomposition phase without performing an analysis. The resulting d3plot database is designed to allow visualization of the decomposition by making each part correspond to the group of solids, shells, beams, thick shells, or SPH particles assigned to a particular processor. For example, in a model that includes various element types including solids, part 1 corresponds to the solid elements assigned to processor 1, part 2 corresponds to the solid elements assigned to processor 2, and so on.

This command can be used in conjunction with the *CONTROL_MPP_DECOMPOSITION_NUMPROC command to run on one processor and produce a d3plot file to visualize the resulting decomposition for the number of processors specified in *CONTROL_MPP_DECOMPOSITION_NUMPROC.

***CONTROL_MPP_DECOMPOSITION_TRANSFORMATION**

Purpose: Specifies transformations to apply to modify the decomposition.

There are 10 different kinds of decomposition transformations available. For a detailed description of each, see Appendix O the LS-DYNA MPP user guide.

The data cards for this keyword consist of transformation operations. Each operation, depending on its type, involves either one or two additional cards. The input deck may include an arbitrary number of transformations with the next keyword, "*" card, terminating this input.

Transformation Card 1. For each transformation this card is required.

Card 1	1	2	3	4	5	6	7	8
Variable	TYPE	V1	V2	V3	V4	V5	V6	
Type	A10	F	F	F	F	F	F	
Default	none	0.0	0.0	0.0	0.0	0.0	0.0	

Transformation Card 2. Additional card for TYPE set to one of VEC3, C2R, S2R, MAT.

Card 2	1	2	3	4	5	6	7	8
Variable	V7	V8	V9					
Type	F	F	F					
Default	0.0	0.0	0.0					

VARIABLE	DESCRIPTION
TYPE	Which transformation to apply. The allowed values are RX, RY, RZ, SX, SY, SZ, VEC3, C2R, S2R, and MAT.
V1 - V9	For type set to either RX, RY, RZ, SX, SY, or SZ: The parameter V1 gives either the angle of rotation (RX, RY, RZ) or the magnitude for the scaling (SX, SY, SZ). The remaining parameters are ignored.

***CONTROL**

***CONTROL_MPP_DECOMPOSITION_TRANSFORMATION**

VARIABLE	DESCRIPTION
	<p>For type set to either VEC3, C2R, S2R, or MAT: All parameters are used. See the appendix for the "pfile."</p>

Remarks:

When using scale factors (SX, SY, SZ), specify a non-zero scale factor; if left blank, it will be set to use 0.0.

***CONTROL_MPP_IO_LSTC_REDUCE**

Purpose: Use LST's own reduce routine to consistently sum floating point data among processors. There are no input parameters.

Remarks:

1. **PFILE.** The related command in the partition file (pfile) is: lstdc_reduce.
2. **Motivation behind this Keyword.** LS-DYNA/MPP uses the Recursive Coordinates Bisection (RCB) algorithm to divide the model element-wise into sub-domains; each sub-domain (group of elements) will be executed in its own CPU core. For shared nodes with data on more than one processor, both internal and external forces must be added across the processors to determine the correct loading; this is called the "REDUCE" operation in MPI.

Adding up the values across processors uses the standard MPI interface provided by different vendors, such as Platform MPI, Intel MPI, Open MPI, etc. Each vendor has its own algorithm for better communication. Therefore, the order of summation for shared nodes varies depending on the vendor's algorithm as well as hardware differences, namely, the number of cores per socket. This numerical noise triggers undesired bifurcation for explicit analysis with lots of integration cycles.

LSTC_REDUCED ensures a deterministic summation order to provide numerical consistency. This keyword gives identical results while using the same decomposition and core counts but different MPI algorithms.

***CONTROL**

***CONTROL_MPP_IO_NOBEAMOUT**

***CONTROL_MPP_IO_NOBEAMOUT**

Purpose: Suppress beam, shell, and solid element failure messages in the d3hsp and message files. There are no parameters for this keyword.

Remarks:

Command in parallel option file (pfile): nobeamout.

***CONTROL_MPP_IO_NOD3DUMP**

Purpose: Suppresses the output of the d3dump and runrsf files.

There are no input parameters for this keyword.

***CONTROL**

***CONTROL_MPP_IO_NODUMP**

***CONTROL_MPP_IO_NODUMP**

Purpose: Suppresses the output of all dump files and full deck restart files.

There are no input parameters. The existence of this keyword causes the d3dump and runrsf file output routines to be skipped. It also suppresses output of the full deck restart files d3full and runfull.

***CONTROL_MPP_IO_NOFULL**

Purpose: Suppresses the output of the full deck restart files.

There are no input parameters. The existence of this keyword suppresses the output of the full deck restart files d3full and runfull.

***CONTROL**

***CONTROL_MPP_IO_NOTIEDIO**

***CONTROL_MPP_IO_NOTIEDIO**

Purpose: Store the tied_nodes temp files in memory instead of outputting to disk to avoid problems on distributed file systems. This keyword has no input parameters.

Remarks:

1. **PFILE.** The related command in the partition file (pfile) is: notiedio.

***CONTROL_MPP_IO_SWAPBYTES**

Purpose: Swap bytes on some of the output files.

There are no input parameters. The existence of this keyword causes the d3plot file and the “interface component analysis” file to be output with bytes swapped. This is to allow further processing of data on a different machine that has big endian vs. little endian incompatibilities compared to the system on which the analysis is running.

***CONTROL**

***CONTROL_MPP_MATERIAL_MODEL_DRIVER**

***CONTROL_MPP_MATERIAL_MODEL_DRIVER**

Purpose: Enable this feature in MPP mode. To allow MPP reader to pass the input phase even without any nodes and elements but using only one processor.

***CONTROL_MPP_PFILE**

Purpose: Provide keyword support for the MPP “p=” pfile options

All lines of input up to the next keyword card will be copied to a temporary file which is effectively pre-pended to the “p=” file given on the command line (even if no such file is given). This allows all options available via the “p=” file to be specified in the keyword input. The only restriction is that pfile directives in the “directory” section are not available, as those must be processed before the keyword input file is read. See the “LS-DYNA MPP User Guide” in the appendix for details of the available pfile commands and their syntax.

*CONTROL

*CONTROL_MPP_REBALANCE

*CONTROL_MPP_REBALANCE

Purpose: Specify parameters associated with dynamic load balancing

Card 1	1	2	3	4	5	6	7	8
Variable	NCYCLE	ICOOR	ICOST	THRES				
Type	I	I	I	I				
Default	optional	0	0	1				

VARIABLE	DESCRIPTION
NCYCLE	Number of cycles between rebalance steps
ICOOR	Coordinates used in rebalance: EQ.0: Current coordinates NE.0: Coordinates at $t = 0$
ICOST	Element costs used in rebalance: EQ.0: Time costs EQ.1: Original
THRES	Percent threshold for rebalancing when performing in-core adaptivity (see Remark 1). For in-core adaptivity, only include this field.

Remarks:

1. **In-Core Adaptivity Threshold.** After each adaptive step, the percent increase in the number of shell elements since the last rebalance is computed on each processor. If the maximum of these values minus the minimum exceeds this threshold, then a rebalance is performed.
2. **Supported Features.** This keyword is supported for the features listed below. Models that contain features not included in this list will have unpredictable results.
 - Nodes
 - Shells, solids, beams, thick shells, and discrete elements

- Seatbelt features (sliprings, retractors, etc.)
- Contact except SOFT = 2, CONSTRAINT, and beam
- Control volumes
- Velocity boundary conditions
- Nodal SPCs
- Files: d3plot, elout, nodout, bndout, glstat, abstat, deforc, jntforce, rcfrc.
sbtout, and sleout
- Adaptive constraints
- In-core adaptivity
- Rigid walls
- Joints
- Nodal rigid bodies
- Pressure loads
- Loads applied to nodes
- Lumped masses
- Cross sections
- Constrained sets (welds)

***CONTROL**

***CONTROL_NONLOCAL**

***CONTROL_NONLOCAL**

Purpose: Allocate additional memory for *MAT_NONLOCAL option.

Card 1	1	2	3	4	5	6	7	8
Variable	MEM							
Type	I							
Default	none							

VARIABLE

DESCRIPTION

MEM

This parameter is no longer used and any value input will be ignored. Memory is allocated dynamically as needed.

CONTROL_OUTPUT**CONTROL*****CONTROL_OUTPUT**

Purpose: Set miscellaneous output parameters.

Card Summary:

Card 1. This card is required.

NPOPT	NEECHO	NREFUP	IACCOP	OPIFS	IPNINT	IKEDIT	IFLUSH
-------	--------	--------	--------	-------	--------	--------	--------

Card 2. This card and all remaining cards are optional.

IPRTF	IERODE	TET10S8	MSGMAX	IPCURV	GMDT	IP1DBLT	EOCS
-------	--------	---------	--------	--------	------	---------	------

Card 3. This card is optional.

TOLEV	NEWLEG	FRFREQ	MINFO	SOLSIG	MSGFLG	CDETOL	IGEOM
-------	--------	--------	-------	--------	--------	--------	-------

Card 4. This card is optional.

PHSCHNG	DEMDEN	ICRFILE	SPC2BND	PENOUT	SHLSIG	HISNOUT	ENGOUT
---------	--------	---------	---------	--------	--------	---------	--------

Card 5. This card is optional.

INSF	ISOLSF	IBSF	ISSF	MLKBAG	KINENG	ISFCNT	
------	--------	------	------	--------	--------	--------	--

Card 6. This card is optional

IELOGKEY	IELOGINI	IELOGSOL					
----------	----------	----------	--	--	--	--	--

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	NPOPT	NEECHO	NREFUP	IACCOP	OPIFS	IPNINT	IKEDIT	IFLUSH
Type	I	I	I	I	F	I	I	I
Default	0	0	0	0	0.	0	100	5000

VARIABLE**DESCRIPTION**

NPOPT

Print suppression during input phase flag for the d3hsp file:

*CONTROL

*CONTROL_OUTPUT

VARIABLE	DESCRIPTION
	EQ.0: No suppression, EQ.1: Nodal coordinates, element connectivities, rigid wall definitions, nodal SPCs, initial velocities, initial strains, adaptive constraints, and SPR2/SPR3 constraints are not printed.
NEECHO	Print suppression during input phase flag for echo file: EQ.0: All data printed, EQ.1: Nodal printing is suppressed, EQ.2: Element printing is suppressed, EQ.3: Both nodal and element printing is suppressed.
NREFUP	Flag to update reference node coordinates for beam formulations 1, 2, and 11. This option requires that each reference node is unique to the beam: EQ.0: Do not update reference node. EQ.1: Update reference node. This update is required for proper visualization of the beam cross-section orientation in LS-PrePost beyond the initial ($t = 0.0$) plot state. NREFUP does not affect the internal updating of the beam cross-section orientation in LS-DYNA.
IACCOP	Flag to average or filter nodal accelerations output to file nodout and the time history database d3thdt: EQ.0: No average (default), EQ.1: Averaged between output intervals, EQ.2: Accelerations for each time step are stored internally and then filtered over each output interval using a filter from General Motors [Sala, Neal, and Wang, 2004] based on a low-pass Butterworth frequency filter. See also [Neal, Lin, and Wang, 2004]. DT2MS in *CONTROL_TIMESTEP must be set to a negative value when IACCOP = 2 so that the maximum possible number of time steps for an output interval is known and adequate memory can be allocated.
OPIFS	Output time interval for interface file written per *INTERFACE_COMPONENT_OPTION.

CONTROL_OUTPUT**CONTROL**

VARIABLE	DESCRIPTION
IPNINT	Flag controlling output of initial time step sizes for elements to d3hsp: EQ.0: 100 elements with the smallest time step sizes are printed. EQ.1: Time step sizes for all elements are printed. GT.1: IPNINT elements with the smallest time step sizes are printed.
IKEDIT	Problem status report interval steps to the d3hsp file. This flag is ignored if the glstat file is written; see *DATABASE_GLSTAT .
IFLUSH	Number of time steps interval for flushing I/O buffers. The default value is 5000. If the I/O buffers are not emptied and an abnormal termination occurs, the output files can be incomplete. The I/O buffers for restart files are emptied automatically whenever a restart file is written so these files are not affected by this option.

This card is optional.

Card 2	1	2	3	4	5	6	7	8
Variable	IPRTF	IERODE	TET10S8	MSGMAX	IPCURV	GMDT	IP1DBLT	EOCS
Type	I	I	I	I	I	F	I	I
Default	0	0	2	50	0	0.	0	0

VARIABLE	DESCRIPTION
IPRTF	Default print flag for rbdout and matsum files. This flag defines the default value for the print flag which can be defined in the part definition section; see *PART . This option is meant to reduce the file sizes by eliminating data which is not of interest. EQ.0: Write part data into both matsum and rbdout EQ.1: Write data into rbdout file only EQ.2: Write data into matsum file only EQ.3: Do not write data into rbdout and matsum
IERODE	Output the eroded internal and kinetic energies into the matsum file. Also, (1) under the heading of part ID 0 in matsum, output the

*CONTROL

*CONTROL_OUTPUT

VARIABLE	DESCRIPTION
	kinetic energy from nonstructural mass, lumped mass elements, and lumped inertia elements; and (2) under the heading of part ID -1in matsum, output the kinetic energy associated with distributed mass from *ELEMENT_MASS_PART. See Remark 3.
	EQ.0: Do not output extra data. EQ.1: Output the eroded internal and kinetic energies and other additional kinetic energies.
TET10S8	Output ten connectivity nodes for the 10-node solid tetrahedral (solid formulations 16/17) and the eight connectivity nodes for the 8-node shell (shell formulation 23) into the d3plot, d3part, d3eigv, and d3mode databases. The current default is set to 2 since this change in the databases may make the data unreadable for many popular post-processors and older versions of LS-PrePost. EQ.1: Write the full node connectivity into the databases EQ.2: Write only the corner nodes of the elements into the databases
MSGMAX	Maximum number of each error/warning message: GT.0: Number of messages to screen output; all messages written to d3hsp/messag LE.0: Number of messages to screen output and d3hsp/messag EQ.0: Default, 50
IPCURV	Flag to output digitized curve data to messag and d3hsp files: EQ.0: Off EQ.1: On
GMDT	Output interval for recorded motions from *INTERFACE_SSI-AUX
IP1DBLT	Output information of 1D (bar-type) seatbelt created for 2D (shell-type) seatbelt to sbtout. EQ.0: The analysis results of internally created 1D seatbelts are extracted and processed to yield the 2D belt information. The 2D belt information is stored in sbtout. EQ.1: The analysis results of internally created 1D retractors and slip rings are stored in sbtout. Belt load can be yielded by

CONTROL_OUTPUT**CONTROL**

VARIABLE	DESCRIPTION
	<p>*DATABASE_CROSS_SECTION. This might lead to different results from that of IP1DBLT = 0 in MPP if the model is not robust.</p> <p>EQ.2: Same as IP1DBLT = 1, but the model is decomposed in the same way as IP1DBLT = 0 in MPP, which guarantees result consistency.</p>
EOCS	<p>elout Coordinate System: controls the coordinate system to be used when writing out shell data to the elout file. EOCS has no effect on eloutdet. EOCS has no effect on elout if OPTION2 in *DATABASE_ELOUT is greater than zero.</p> <p>EQ.0: Default (local element coordinate system, or if an orthotropic material model and CMPFLG = 1, then material coordinate system)</p> <p>EQ.1: Local element coordinate system</p> <p>EQ.2: Global coordinate system</p>

This card is optional.

Card 3	1	2	3	4	5	6	7	8
Variable	TOLEV	NEWLEG	FRFREQ	MINFO	SOLSIG	MSGFLG	CDETOL	IGEOM
Type	I	I	I	I	I	I	F	I
Default	2	0	1	0	0	0	10.0	1

VARIABLE	DESCRIPTION
TOLEV	Timing Output Levels: controls the number of levels output in the timing summary at termination. The default is 2.
NEWLEG	New Legends: controls the format of the LEGEND section of various ASCII output files.
	<p>EQ.0: Use the normal format</p> <p>EQ.1: Use the optional format with extra fields.</p>
FRFREQ	Output frequency for failed element report in cycles. The default is to report the summary every cycle in which an element fails.

VARIABLE	DESCRIPTION
	If > 1 , the summary is reported every FRFREQ cycles whether an element fails that cycle or not, provided some element has failed since the last summary report. Individual element failure is still reported as it occurs.
MINFO	<p>Output penetration and memory information for Mortar contact after each implicit step. Reporting this information is also available for explicit analysis when MINFO is a negative number.</p> <p>LT.0: Report for each contact interface at the time frequency MINFO. This is available for explicit and implicit analysis.</p> <p>EQ.0: No information reported.</p> <p>EQ.1: Report for each contact interface at each implicit time step. No information is output for explicit analysis.</p>
SOLSIG	<p>Flag to extrapolate stresses and other history variables for multi-integration point solids from integration points to nodes. These extrapolated nodal values replace the integration point values normally stored in d3plot. When a nonzero SOLSIG is invoked, NINTSLD in *DATABASE_EXTENT_BINARY should be set to 8 as any other value of NINTSLD will result in only one value being reported for each element. Supported solid formulations are -1, -2, 2, 3, 4, 16, 17, -18, 18, 23, and 62.</p> <p>EQ.0: No extrapolation.</p> <p>EQ.1: Extrapolate the stress for linear materials only.</p> <p>EQ.2: Extrapolate the stress if plastic strain is zero.</p> <p>EQ.3: Extrapolate the stress always.</p> <p>EQ.4: Extrapolate all history variables.</p>
	<p>NOTE: Do not use "Setting – Extrapolate" in LS-PrePost when this field, SOLSIG, is nonzero.</p>
MSGFLG	<p>Flag for writing detailed error/warning message to d3msg. MSGFLG has no effect on output of standard length error/warning messages; such messages are written to messag or mes****. NOTE: Most errors/ warnings offer only standard length messages. Only a few also offer optional, detailed messages.</p> <p>EQ.0: Do not write detailed messages to d3msg.</p>

CONTROL_OUTPUT**CONTROL**

VARIABLE	DESCRIPTION
	EQ.1: Write detailed messages to d3msg at the conclusion of the run. Each detailed message is written only once even in cases where the associated error or warning occurs multiple times. A detailed message written to d3msg should only be used to help interpret the standard-length message better. The information in d3msg could contain fictitious IDs and names.
CDETOL	Tolerance for output of *DEFINE_CURVE discretization warnings. After each curve is discretized, the resulting curve is evaluated at each of the original definition points, and the values compared. A warning will be issued for any curve where this comparison results in an error of more than CDETOL/100 × M, where the curve-specific value M is computed as the median of the absolute values of the non-zero curve values.
IGEOM	Flag to control whether nodal coordinates or displacements for the nodes in the mesh are output to d3plot, d3part, and d3drif: EQ.1: Nodal coordinates (default) EQ.2: Displacements. IGEOM = 2 is useful when the precision of the binary files is insufficient to accurately calculate the displacements from the coordinates in post-processing. This problem arises when the displacements are very small relative to the coordinate values. Note that IGEOM = 2 is supported for post-processing in LS-PrePost 4.10 and later.

This card is optional.

Card 4	1	2	3	4	5	6	7	8
Variable	PHSCHNG	DEMDEN	ICRFILE	SPC2BND	PENOUT	SHLSIG	HISNOUT	ENGOUT
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

VARIABLE	DESCRIPTION
PHSCHNG	Message to messag file when materials 216, 217, and 218 change phase.

VARIABLE	DESCRIPTION
	EQ.0: No message (default). EQ.1: The time and element ID are written.
DEMDEN	Output DEM density data to d3plot database. EQ.0: No output (default). EQ.1: Output data. EQ.2: Output data with modification of the boundary density calculation to avoid the low-density distribution near the DES domain boundary.
ICRFILE	Flag to output node sets and element sets used in computing secforc data; see *DATABASE_CROSS_SECTION_OPTION and *DATABASE_SECFORC . These sets are written in keyword format (*SET...) and thus can be displayed using LS-PrePost. The assigned set IDs are the same as the ID of the cross-section. EQ.0: Do not write sets (default). EQ.1: Write a separate file for each cross-section called cross_section_# where # is the cross-section ID. EQ.2: Write sets for all cross-sections to a file called cross_sections.
SPC2BND	Flag to convert constraints on rigid bodies (see CMO/CON1/CON2 on *MAT_RIGID and *CONSTRAINED_NODAL_RIGID_BODY_SPC) to equivalent *BOUNDARY_PRESCRIBED_MOTION_RIGID motion. The purpose of this field is to obtain reaction forces in bndout, without changing the results of the simulation. EQ.0: Not active EQ.1: Active
PENOUT	Flag to output contact penetration to sleout (binout format only) and d3plot for Mortar contact. In sleout the maximum absolute and/or relative penetration per interface is output, in magnitude only. In d3plot a nodal vector field is output for absolute and/or relative penetration, respectively, each giving the maximum penetration (magnitude and direction) for all nodes in any sliding interface. See also NPEN on *DATABASE_EXTENT_INTFOR . EQ.0: Do not output.

VARIABLE	DESCRIPTION
	<p>EQ.1: Output absolute penetration.</p> <p>EQ.2: Output absolute and relative penetration.</p>
SHLSIG	<p>Flag to extrapolate stresses for shells with 8 integration points to nodes. These extrapolated nodal values replace the integration point values normally stored in d3plot. When a nonzero SHLSIG is invoked, MAXINT in *DATABASE_EXTENT_BINARY should be set to -2 to indicate 4 in-plane integration points and 2 in the thickness direction. Supported shell formulations are: 16, 20, and 21.</p> <p>EQ.0: No extrapolation.</p> <p>EQ.1: Extrapolate the stress for linear materials only.</p> <p>EQ.2: Extrapolate the stress if plastic strain is zero.</p> <p>EQ.3: Extrapolate the stress always.</p> <p>EQ.4: Extrapolate all history variables.</p> <div style="border: 1px solid black; padding: 10px; margin-top: 10px;"><p>NOTE: Do not use “Setting – Extrapolate” in LS-PrePost when this field, SHLSIG, is nonzero.</p><p>SHLSIG is only supported for Gauss integration.</p></div>
HISNOUT	<p>Flag to invoke output of extra history variable names. Usually, the extra history variables of material models are given as just numbers. The corresponding meaning of these variables can be determined, for example, using this website: www.dynasupport.com/howtos/material/history-variables.</p> <p>As an alternative, this option allows the output of the history variable names, listed for each part separately, to d3hsp. In addition, XML files that can be read by a post-processor can be output. Currently, two XML files are available:</p> <ul style="list-style-type: none">• hisnames.xml (read by LS-PrePost and GS Animator), which lists the history variable names in each part (as with d3hsp), and• d3labels.xml (read by LS-PrePost), which is similar to hisnames.xml but contains additional information regarding history variable names for each element and integration point when necessary (such as in the case of composites).

*CONTROL

*CONTROL_OUTPUT

VARIABLE	DESCRIPTION							
	The number of supported material models is continuously increasing.							
	EQ.0: No output (default)							
	EQ.1: Information written to d3hsp							
	EQ.2: Information written to d3hsp and XML file hisnames.xml							
	EQ.3: Information written to d3hsp and XML file d3labels.xml (and hisnames.xml).							
ENGOUT	Flag to output contact sliding energy densities to d3plot for Mortar contact. If set to 1, a nodal scalar field is output giving the minimum sliding energy density for each node in any sliding interface. See also NENG on *DATABASE_EXTENT_INTFOR .							

This card is optional.

Card 5	1	2	3	4	5	6	7	8
Variable	INSF	ISOLSF	IBSF	ISSF	MLKBAG	KINENG	ISFCNT	
Type	I	I	I	I	I	I	I	
Default	0	0	0	0	0	0	1	

VARIABLE	DESCRIPTION
INSF	Flag to invoke output of *SET_NODE data: EQ.0: No output (default) EQ.1: Information written to file. See Remark 1 .
ISOLSF	Flag to invoke output of *SET_SOLID data: EQ.0: No output (default) EQ.1: Information written to file. See Remark 1 .
IBSF	Flag to invoke output of *SET_BEAM data: EQ.0: No output (default) EQ.1: Information written to file. See Remark 1 .

CONTROL_OUTPUT**CONTROL**

VARIABLE	DESCRIPTION
ISSF	Flag to invoke output of *SET_SHELL data: EQ.0: No output (default) EQ.1: Information written to file. See Remark 1 .
MLKBAG	Flag to invoke output of accumulated airbag mass leakage in AB-STAT: EQ.0: Airbag mass leakage rate is output (default). EQ.1: Accumulated airbag mass leakage is output.
KINENG	Flag to output kinetic energy density as a nodal field: EQ.0: Do not output (default). EQ.1: Output, will appear as “kinetic energy density” as a nodal field in the d3plot and d3eigv databases.
ISFCNT	Continuity level in applying Interface Linking Data: LE.1: Continuity in displacements. Velocities and accelerations excluded. EQ.2: Continuity in displacements and velocities. Accelerations excluded. EQ.3: Continuity in displacements, velocities, and accelerations. This option may result in a smoother response overall.

This card is optional.

Card 6	1	2	3	4	5	6	7	8
Variable	IELOGKEY	IELOGINI	IELOGSOL					
Type	I	I	I					
Default	0	0	0					

VARIABLE	DESCRIPTION
IELOGKEY	Flag to invoke diagnostic output to a consolidated error file for MPP called <code>error.log</code> during the keyword input phase (see Remark 2):

VARIABLE	DESCRIPTION
	<p>EQ.0: No output during this phase (default)</p> <p>EQ.1: Output errors only</p> <p>EQ.2: Output errors and warnings</p> <p>EQ.3: Output errors, warnings, and informational output</p>
IELOGINI	<p>Flag to invoke diagnostic output to a consolidated error file for MPP called <code>error.log</code> during the initialization phase (see Remark 2):</p> <p>EQ.0: No output during this phase (default)</p> <p>EQ.1: Output errors only</p> <p>EQ.2: Output errors and warnings</p> <p>EQ.3: Output errors, warnings, and informational output</p>
IELOGSOL	<p>Flag to invoke diagnostic output to a consolidated error file for MPP called <code>error.log</code> during the solution phase (see Remark 2):</p> <p>EQ.0: No output during this phase (default)</p> <p>EQ.1: Output errors only</p> <p>EQ.2: Output errors and warnings</p> <p>EQ.3: Output errors, warnings, and informational output</p>

Remarks:

1. **Set output format.** Set data is written to an output file based on the type of set (see table below). Note that the output file has the job ID as a prefix.

Set type	Output file name	Set prefix(ID) in the file
node	<code>nodeset_file</code>	SetN
solid	<code>solidset_file</code>	SetH
beam	<code>beamset_file</code>	SetB
shell	<code>shellset_file</code>	SetS

2. **Consolidated error file.** Nonzero IELOGKEY, IELOGINI, or IELOGSOL cause all processors to output diagnostic data in MPP to a single file named `error.log` for the corresponding phase. Depending on the setting, different information can be output to this file. By default, `error.log` is not created.

3. **Energy in glstat and matsum.** The internal and kinetic energies in `glstat` always include the eroded internal and kinetic energies. In contrast, the internal and kinetic energies in `matsum` depend on `IERODE`. Thus, setting `IERODE` to 0 causes the energies in `glstat` to differ from that in `matsum`. This difference is intentional because `glstat` reflects the global system energy, while in `matsum`, the `IERODE` flag dictates outputting the internal and kinetic energies of interest.

*CONTROL

*CONTROL_PARALLEL

*CONTROL_PARALLEL

Purpose: Control parallel processing usage by defining the number of processors and invoking the optional consistency of the global vector assembly. This command applies only to shared memory parallel (SMP) LS-DYNA. It does not apply to distributed memory parallel (MPP) LS-DYNA.

Card 1	1	2	3	4	5	6	7	8
Variable	NCPU	NUMRHS	CONST	PARA				
Type	I	I	I	I				
Default	1	0	2	0				
Remarks		1	2	3				

VARIABLE	DESCRIPTION
NCPU	Number of cpus used. (This parameter is disabled in 971 R5 and later versions. Set number of cpus using "ncpu =" on the execution line — see Execution Syntax section of Getting Started — or on the *KEYWORD line of the input.)
NUMRHS	Number of right-hand sides allocated in memory: EQ.0: Same as NCPU, always recommended, EQ.1: Allocate only one.
CONST	Consistency flag. (Including "ncpu = n " on the execution line or on the *KEYWORD line of input overrides CONST. The algebraic sign of n determines the consistency setting.) EQ.1 or $n < 0$: On (recommended) EQ.2 or $n > 0$: Off, for a faster solution (default).
PARA	Flag for parallel force assembly if CONST = 1. (Including "para =" on the execution line overrides PARA.) EQ.0: Off EQ.1: On (see Remark 3)

VARIABLE	DESCRIPTION
	EQ.2: On (see Remark 3)

Remarks:

1. **NUMHRS.** It is recommended to always set NUMRHS = NCPU because then the force assembly is done in parallel which greatly improves the parallel performance. Setting NUMRHS to one reduces storage by one right-hand side vector for each additional processor after the first. If the consistency flag is active, that is, CONST = 1, NUMRHS defaults to unity.
2. **Consistency.** For any given problem with the consistency option off, i.e., CONST = 2, slight differences in results are seen when running the same job multiple times with the same number of processors and also when varying the number of processors. Comparisons of nodal accelerations often show wide discrepancies; however, it is worth noting that the results of accelerometers often show insignificant variations due to the smoothing effect of the accelerometers which are generally attached to nodal rigid bodies.

The accuracy issues are not new and are inherent in numerical simulations of automotive crash and impact problems where structural bifurcations under compressive loads are common. This problem can be easily demonstrated by using a perfectly square thin-walled tubular beam of uniform cross-section under a compressive load. Typically, every run on one processor that includes a minor input change (i.e., element or hourglass formulation) will produce dramatically different results in terms of the final shape, and, likewise, if the same problem is again run on a different brand of computer. If the same problem is run on multiple processors the results can vary dramatically from run to run WITH NO INPUT CHANGE. The problem here is due to the randomness of numerical round-off which acts as a trigger in a “perfect” beam.

Since summations will (CONST = 2) occur in a different order from run to run, the round-off is also random. The consistency flag, CONST = 1, provides for identical results (or nearly so) whether one, two, or more processors are used while running in the shared memory parallel (SMP) mode. This is done by requiring that all contributions to global vectors be summed in a precise order independently of the number of processors used. When checking for consistent results, nodal displacements or element stresses should be compared. The NODOUT and ELOUT files should be digit-to-digit identical. However, the GLSTAT, SECFORC, and many of the other ASCII files will not be identical since the quantities in these files are summed in parallel for efficiency reasons and the ordering of summation operations is not enforced. The biggest drawback of this option is the CPU cost penalty which is at least 15 percent if PARA = 0 and is much less if PARA = 1 and 2 or more processors are used. Unless the PARA flag

is on (for non-vector processors), parallel scaling is adversely affected. The consistency flag does not apply to MPP parallel.

3. **PARA.** The PARA flag set to 1 or 2 will cause the force assembly for the consistency option to be performed in parallel for the SMP version, so better scaling will be obtained. However, PARA = 1 will increase memory usage while PARA = 2 will not. This flag does not apply to the MPP version. If PARA = 0, CONST = 0, and NUMRHS = NCPU, the force assembly by default is done in parallel, but without consistency. The value of the flag may also be given by including “para = <value>” on the execution line, and the value given in this manner will override the value of PARA in *CONTROL_PARALLEL.

***CONTROL_PORE_AIR**

Purpose: Set parameters for pore air pressure calculations.

Card 1	1	2	3	4	5	6	7	8
Variable	AIR_RHO	AIR_P	ETERM	ANAMSG	OPTAPA			
Type	F	F	F	I	I			
Default	none	none	endtim	0	0			

VARIABLE	DESCRIPTION
AIR_RHO	Density of atmospheric air, = 1.184 kg/m ³ at 25°C
AIR_P	Pressure of atmospheric air, = 101.325 kPa at 25°C
ETERM	Event termination time. The default is ENDTIM of *CONTROL_TERMINATION . If ETERM is defined and smaller than ENDTIM, LS-DYNA terminates, by default, when the simulation time reaches ETERM. However, OTAPA below provides options for continuing the simulation beyond ETERM.
ANAMSG	Flag to turn off the printing of pore air analysis status message, including the analysis time, the node with the highest pressure change. EQ.0: Status messages are printed, the default value. EQ.1: Status messages are not printed
OPTAPA	Option when 0.0 < ETERM < ENDTIM: EQ.0: The LS-DYNA simulation terminates when time reaches ETERM. EQ.1: The LS-DYNA simulation continues beyond time ETERM. However, pore air pressure is not updated after ETERM, and the last updated pore air pressure applies for the rest of the simulation. EQ.2: The LS-DYNA simulation continues beyond time ETERM, but pore air pressure does not apply after time ETERM.

*CONTROL

*CONTROL_PORE_FLUID

*CONTROL_PORE_FLUID

Purpose: Set parameters for pore water pressure calculations.

This control card is intended for soil analysis but is also applicable to any other materials containing pore fluid. The pore-pressure capabilities invoked by this card are available for explicit analysis only, not for implicit, and are restricted to solid element formulations 1, 2, 10, and 15 and thick shell formulations 1, 2, 3, 5, 6 and 7.

LS-DYNA uses Terzaghi's Effective Stress to model materials with pore pressure. The pore fluid and soil skeleton are assumed to occupy the same volume and to carry loads in parallel. Thus, the total stress in an element is the sum of the "effective stress" in the soil skeleton, plus the pressure in the pore fluid. LS-DYNA calculates the "effective stress" with standard material models. The pore fluid treatment, then, is independent of material model. The pore pressure is calculated at nodes and interpolated onto the elements. The pore fluid's hydrostatic stress is equal to the negative of the element pore pressure.

Card 1	1	2	3	4	5	6	7	8
Variable	ATYPE	(blank)	WTABLE	PF_RHO	GRAV	PF_BULK	OUTPUT	TMF
Type	I	F	F	F	F	F	I	F
Default	0	0.0	0.0	none	none	none	0	1.0

Card 2	1	2	3	4	5	6	7	8
Variable	TARG	FMIN	FMAX	FTIED	CONV	CONMAX	ETERM	THERM
Type	F	F	F	F	F	F	F	F
Default	TMF	0.0	0.0	0.0	10^{-4}	10^{20}	0.0	0.0

CONTROL_PORE_FLUID**CONTROL**

Card 3 is optional

Card 3	1	2	3	4	5	6	7	8
Variable	ETFLAG							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
ATYPE	Analysis type for pore water pressure calculations (see Analysis Types): EQ.0: No pore water pressure calculation EQ.1: Undrained analysis EQ.2: Drained analysis EQ.3: Time dependent consolidation (coupled) EQ.4: Consolidate to steady state (uncoupled) EQ.5: Drained in dynamic relaxation, undrained in transient EQ.6: Same as 4 but do not check convergence, continue to end time
WTABLE	Default z-coordinate of water table (where pore pressure is zero)
PF_RHO	Default density for pore water
GRAV	Gravitational acceleration used to calculate hydrostatic pore water pressure
PF_BULK	Default bulk modulus of pore fluid (stress units)
OUTPUT	Flag controlling stresses output to d3plot, d3thdt, and elout: EQ.0: Total stresses are output. EQ.1: Effective stresses are output; see Output section below.
TMF	Initial Time Magnification Factor for seepage (ATYPE = 3 and 4 only):

*CONTROL

*CONTROL_PORE_FLUID

VARIABLE	DESCRIPTION
	GT.0: Factor (can be used with automatic control, see TARG, FMIN, FMAX)
	LT.0: Load curve ID (see *DEFINE_CURVE) giving Time Magnification Factor as a function of analysis time
TARG	Target for maximum change of excess pore pressure head at any node, per timestep. Head is defined in Remark 2 . If the actual change falls below the target, the time factor for the seepage calculation will be increased (see Time Factoring). If zero, the constant value of TMF is used. If non-zero, TMF is taken as the initial factor.
FMIN	Minimum time factor for seepage calculation
FMAX	Maximum time factor for seepage calculation
FTIED	Analysis type for pore water pressure calculations (see Remark 1): EQ.0.0: Tied contacts act as impermeable membranes, EQ.1.0: Fluid may flow freely through tied contacts.
CONV	Convergence tolerance for ATYPE = 4. Maximum head change per time step at any node. See Remark 2 below for the definition of "head".
CONMAX	Maximum factor on permeability for ATYPE = 4
ETERM	Event time termination (ATYPE = 3)
THERM	Thermal expansion: Volumetric strain per degree increase for undrained soil.
ETFLAG	Flag for interpretation of time (see Time Factoring): EQ.0: Time means analysis time. EQ.1: Time means event time.

Analysis Types (see ATYPE):

1. **Undrained.** For analyses of the “undrained” type the pore fluid is trapped within the material. Volume changes result in pore pressure changes. This approximation is used to simulate the effect of rapidly applied loads on relatively impermeable soil.

2. **Drained.** For analyses of the “drained” type the pore fluid is free to move within the material such that the user-defined pressure as a function of z-coordinate relationship is always maintained. This approximation is used to model high-permeability soils.
3. **Time-Dependent Consolidation.** For the analysis type “time dependent consolidation” pressure gradients cause pore fluid to flow through the material according to Darcy’s law:

$$\mathbf{v} = \kappa \nabla(p + z)$$

where

\mathbf{v} = fluid seepage velocity vector
 κ = permeability
 p = pressure head
 z = z-coordinate.

The seepage velocity is defined as fluid volume flow rate per unit cross-sectional area. Net inflow or outflow at a node leads to a theoretical volume gain or loss. The analysis is coupled, that is, any difference between actual and theoretical volume leads to pore pressure change, which in turn affects the fluid flow. The result is a prediction of response as a function of time.

4. **Steady-State Consolidation.** For the analysis type “steady-state consolidation” an iterative method is used to calculate the steady-state pore pressure. The analysis is uncoupled, that is, only the final state is meaningful, not the response as a function of time.

Time Factoring:

Consolidation occurs over time intervals of days, weeks, or months. To simulate this process using explicit time integration, a time factor is used. The permeability of the soil is increased by the time factor so that consolidation occurs more quickly. The output times in d3plot, d3thdt, and the ascii files are modified to reflect the time factor. The factored time (“Event Time”) is intended to represent the time taken in the real-life consolidation process and will usually be much larger than the analysis time (the analysis time is the sum of the LS-DYNA timesteps). The time factor may be chosen explicitly (using TMF), but we recommend using automatic factoring instead. The automatic scheme adjusts the time factor according to how quickly the pore pressure is changing. Usually at the start of consolidation the pore pressure changes quickly, and the time factor is low. The time factor increases gradually as the rate of pore pressure change reduces. Automatic time factoring is input by setting TARG (the target pore pressure head change per timestep) and maximum and minimum allowable time factors; for example, TARG = 0.001 to 0.01 m head, FMIN = 1.0, and FMAX = 10^6 . Optimum settings for these are model-dependent.

Loading, other input data from load curves, and output time-intervals on *DATABASE cards use the analysis time by default (for example, the *x*-axis of a load curve used for pressure loading is analysis time). When performing consolidation with automatic time-factoring, the relationship between analysis time and event time is unpredictable. Termination based on event time may be input using ETERM.

It may also be desired to apply loads as functions of event time rather than analysis time, since the event time is representative of the real-life process. By setting ETFLAG = 2, the time axis of all load curves used for any type of input as a function of time, and output intervals on *DATABASE cards, will be interpreted as event time. This method also allows consolidation to be used as part of a staged construction sequence – when ETFLAG = 2, the stages begin and end at the “real time” stage limits and input curves of pore pressure analysis type as a function of time may be used to enforce, for example, consolidation in some stages and undrained behavior in others.

Output:

Five extra variables for solid elements are automatically written to the d3plot and d3thdt files when the model contains *CONTROL_PORE_FLUID, in addition to the NEIPH variables requested on *DATABASE_EXTENT_BINARY. The first of the five is the pore pressure in stress units; the second is the excess pore pressure, meaning actual minus hydrostatic pressure. These follow the NEIPH extra variables requested by the user, so for example if NEIPH = 3 then there will be a total of 8 extra variables in the d3plot and d3thdt files of which the fourth is pore pressure. Even if NEIPH = 0, the d3plot and d3thdt files will still contain the five extra variables related to pore pressure. The same five extra variables are also written to the elout file, but only if OPTION1 > 0 on *DATABASE_ELOUT.

Further optional output to d3plot, d3thdt, and nodout files is available for nodal pore pressure variables; see *DATABASE_PWP_OUTPUT.

For time-dependent and steady-state consolidation, information on the progress of the analysis is written to d3hsp file.

Remarks:

1. **Tied Contacts.** By default, the mesh discontinuity at a tied contact will act as a barrier to fluid flow. If the flag FTIED is set to 1.0, then pore fluid will be transmitted across tied nodes in tied contacts (*CONTACT_TIED_SURFACE_TO_SURFACE and *CONTACT_TIED_NODES_TO_SURFACE, including OFFSET and non-OFFSET types). This algorithm has an effect only when the analysis type of at least one of the contacting parts is 3, 4 or 6.

2. **Pressure and Head.** The term “head” means pressure, defined in terms of the height of fluid needed to generate that pressure hydrostatically. Head is given in length units. Head relates to pressure measured in stress units as follows:

$$h = \frac{p}{\rho g}$$

where h is head (in length units), p is pressure (in stress units), ρ is pore fluid density (PF_RHO), and g is gravitational acceleration (GRAV).

3. **See also *BOUNDARY_PORE_FLUID.** Only the parts for which *BOUNDARY_PORE_FLUID is defined will be treated as containing pore fluid. Parts not included in a *BOUNDARY_PORE_FLUID definition will be treated as dry, meaning they do not have pore fluid; the settings on *CONTROL_PORE_FLUID will not affect them. Pore fluid properties, water table and analysis type are input on a per part basis on *BOUNDARY_PORE_FLUID. Nonzero values given on *BOUNDARY_PORE_FLUID override those on *CONTROL_PORE_FLUID for that particular part.
4. **Material Properties and Density.** For parts containing pore fluid, the properties given on the *MAT card determine the effective stresses. For example, the bulk modulus on the *MAT card should not include the bulk stiffness of the pore water. The only material property on the *MAT card that relates to the soil/water mixture (as opposed to the soil skeleton) is the density, RO, for which the density of saturated soil should be input. This is somewhat higher than the density of dry soil. The mass and weight of the soil are determined solely from the RO on the *MAT card, and not from the pore fluid density PF_RHO given on *CONTROL_PORE_FLUID or *BOUNDARY_PORE_FLUID.
5. **Part Associativity.** Pore pressure is a nodal variable, but analysis type and other pore pressure related inputs are properties of parts. When a node is shared by elements of different parts, and those parts have different pore pressure inputs, the following rules are followed to determine which part’s properties should be applied to the node.
 - a) Dry parts (meaning parts without a *BOUNDARY_PORE_FLUID card) will never be used (lowest priority).
 - b) If a part is initially dormant (due to staged construction inputs), it has next-lowest priority.
 - c) Parts with analysis type set to drained have highest priority.
 - d) Next, higher permeability gives higher priority.
 - e) If two or more parts have equal-highest priority at a node, the part with lowest ID will win.

6. **Related Keywords.** The following are cards related to this keyword:

*BOUNDARY_PORE_FLUID

*BOUNDARY_PWP_OPTION

*DATABASE_PWP_OUTPUT

*DATABASE_PWP_FLOW

*MAT_ADD_PERMEABILITY

***CONTROL_PZELECTRIC**

Purpose: Set solver options for a piezoelectric material; see *MAT_ADD_PZELECTRIC.

Card 1	1	2	3	4	5	6	7	8
Variable	SOLVER	MSGITR	MAXITR	ABSTOL	RELTOL	NDTRFK	EPZMSG	
Type	I	I	I	F	I	I	I	
Default	11	0	500	10^{-20}	10^{-10}	1	0	

VARIABLE	DESCRIPTION
SOLVER	Piezoelectric solver type: EQ.11: Direct solver EQ.12: Diagonal scaling conjugate gradient iterative, recommended for MPP for better scalability
MSGITR	Output iteration message level for SOLVER = 12: EQ.0: No output (default) EQ.1: Summary information
MAXITR	Maximum number of iterations for SOLVER = 12. EQ.0: Use default value 100.
ABSTOL	Absolute convergence tolerance, for SOLVER = 12. EQ.0.0: Use default value 10^{-20} .
RELTOL	Relative convergence tolerance, for SOLVER = 12. EQ.0.0: Use default value 10^{-10} .
NDTRFK	Reform the dielectric stiffness matrix every NDTRFK time steps. LT.0: Curve NDTRFK defines the stiffness reformation time step as a function of time.
EPZMSG	Flag to determine if electric flux and electric field at the element center of piezoelectric material is output to d3plot: EQ.0: No electric flux or electric field output to d3plot

***CONTROL**

***CONTROL_PZ ELECTRIC**

VARIABLE

DESCRIPTION

EQ.1: x , y , and z strain slots in d3plot store the electric flux along the x , y , and z directions, respectively. xy , yz , and zx strain slots in d3plot store the electric field along the x , y , and z directions, respectively.

***CONTROL_REFERENCE_CONFIGURATION_{OPTION}**

Available options include:

<BLANK>

ITER

Purpose: Find an approximate reference geometry given a measured/known deformed geometry. A solution to this inverse problem is found iteratively by means of an optimization method. This keyword needs to be in the main keyword file, not in any of the include files. `case` must be used on the command line (see [Remark 2](#)).

Card 1	1	2	3	4	5	6	7	8
Variable	MAXITER			TARGETFILE				
Type	I			A70				
Default	none			none				

Card 2	1	2	3	4	5	6	7	8
Variable	METHOD	STEP	TOL					
Type	I	F	F					
Default	0	1.0	0.0					

Optional card for the ITER keyword option.

Card 3	1	2	3	4	5	6	7	8
Variable			ITERFILE					
Type			A80					
Default			none					

*CONTROL

*CONTROL_REFERENCE_CONFIGURATION

VARIABLE	DESCRIPTION
MAXITER	Maximum number of iterations. See Remark 1 .
TARGET-FILE	File containing all nodes of the target geometry. See Remark 2 .
METHOD	Iterative method: EQ.1: Sellier's method EQ.2: Rausch's method EQ.3: Rausch's method with an additional line search.
STEP	Step size used in the iterations to update the current approximate reference geometry for Sellier's method. It must be > 0 . See Remark 1 .
TOL	Tolerance used to determine convergence of the iterative method. This is given in the unit of length. See Remark 1 .
ITERFILE	Base name of two files for the ITER keyword option. These files are used to start, or restart, the iterative method. ITERFILE.guess gives an initial guess of the approximate stress free reference geometry. It has the same format as the TARGET file. The second file, ITERFILE.algo , gives algorithmic parameters and uses an internal format. ITERFILE is optional. If ITERFILE is not supplied, TARGET is used as an initial guess. See Remark 2 and 3 for details.

Remarks:

1. **Iterative Method.** The implemented iterative method is a backward displacement method of either Sellier [1] (**METHOD = 1**) or Rausch et.al. [2] (**METHOD = 2** or **3**). It attempts to solve the inverse problem by a sequence of so-called forward solves. In each forward solve the underlying mechanical problem is solved. The reference geometry is then adjusted by subtracting a suitable amount of the obtained displacement.

Let x^* be the target geometry specified by **TARGETFILE**. Then we want to find the unknown reference geometry X^* . We can state the inverse problem as a minimization of the difference $\mathfrak{I}(X) = \|x - x^*\|$ under the constraint that x must be in equilibrium with respect to the applied load. In iteration n of the optimization method, we have a guess of the reference geometry, X_n . For $n = 0$ this geometry is given by **ITERFILE.guess** if specified; otherwise it is given by **TARGETFILE**.

The stated problem provides the solution \mathbf{x}_n as the resulting deformed geometry when applying the prescribed load. It can be shown that an approximate descent direction for \mathfrak{J} yields the update $\mathbf{X}_{n+1} = \mathbf{X}_n - \alpha_n(\mathbf{x}_n - \mathbf{x}^*)$. α_n is the step size STEP for Sellier's method, while for Rausch's method it is given as

$$\alpha_n = \alpha_{n-1} \times \frac{(\mathbf{x}_{n-1} - \mathbf{x}^*)^T(\mathbf{x}_n - \mathbf{x}_{n-1})}{(\mathbf{x}_n - \mathbf{x}_{n-1})^T(\mathbf{x}_n - \mathbf{x}_{n-1})}.$$

For METHOD = 3, a line search is also attempted if the error $\mathbf{x}_n - \mathbf{x}^*$ does not decrease at iteration n . It amounts to bisecting the step at iteration n into $\alpha_n/2^k$ and re-solving with $k = 1, 2, 3, \dots$ until either the error decreases or $k = 5$.

For all METHOD types, the iterations continue until $\mathfrak{J}(\mathbf{X}_n)$ is smaller than the tolerance TOL, or until the maximum number of iterations MAXITER is exceeded. In the implementation, \mathbf{x} and \mathbf{X} are vectors of nodal positions.

2. **Iteration using Cases.** The overall iterative structure is based on the *CASE construct. Starting from a main keyword file (*.key, *.k, *.dyn) which acts as a template, the iterates are automatically generated by parsing this input file. Each iterate is a "case" with a corresponding *.inp file containing the necessary keywords. The input for iterate number n is named jobid.iter{n}.inp, where jobid is an optional job ID. This is essentially a copy of the main keyword file, but with minor modifications reflecting input that is unique to iterate n .

You must have *CONTROL_REFERENCE_CONFIGURATION in the main keyword file. For disk space reasons we recommend reducing the size of the main keyword file to a minimum. The modifications to the main keyword file chiefly pertain to change of node coordinates for the current iterate and associated iteration parameters. At the end of the iteration jobid.iter{n+1}.guess and jobid.iter{n+1}.algo are written with the updated (reference) geometry and algorithmic parameters, respectively. These files are used as starting guess in the next iterate, $n + 1$.

3. **Node Files.** The TARGETFILE and ITERFILE.guess files both contain the nodes subjected to optimization and have the same simple structure

```
*KEYWORD
*NODE
$#    nid          x           y           z
      1        -11.776     -23.849     14.234
...
*END
```

The ITERFILE.algo files have an internal format whose description is currently omitted.

Example 1:

Below is a sketch of a main file showing the use of the *CONTROL_REFERENCE_CONFIGURATION keyword. To run it, the syntax on the command line should be: lsdyna case i=main.k

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
*KEYWORD
*CONTROL_REFERENCE_CONFIGURATION
$# maxiter
        4target.k                                              target node file
$# method      step      tol
        1          1.0     0.001
$ all nodes (ie. target nodes as well as any other nodes)
*INCLUDE
target.k
*ELEMENT_SOLID
$# eid      pid      n1      n2      n3      n4      n5      n6      n7      n8
        1          1          1     127    136     10       2     128     137     11
*CONTROL_IMPLICIT_GENERAL
$# imflag    dt0      imform    nsbs      igs      cnstn      form      zero_v
        1          0.1         2           0         1           0           0           0
*CONTROL_TERMINATION
$# endtim   endcyc    dtmin      endeng    endmas      nosol
        1.0          0         0.0        0.0       0.0           0
*BOUNDARY_SPC_NODE
$# nid      cid      dofpx      dofy      dofz      dofrx      dofry      dofrz
        30          0          1           1           1           1           1           1
*SECTION_SOLID
$# secid    elform      aet
        1          -2           0
*MAT_ELASTIC
$# mid      ro      e      pr      da      db      not used
        11.00000E-0     100.0     0.3     0.0     0.0     0.0      not used
*LOAD_SEGMENT_ID
$# id      oneseg
$# lcid      sf      at      n1      n2      n3      n4      n5
        1          1.0     0.0     185     186     192     191      0
*PART
$# title
boxsolid
$# pid      secid      mid      eosid      hgid      grav      adpopt      tmid
        1          1          1
*END
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

References:

- [1] Sellier, M, “An iterative method for the inverse elasto-static problem,” Journal of Fluids and Structures, Volume 27, issue 8, (2011)
- [2] Rausch, M.K., Genet M., and Humphrey J.D, “An augmented iterative method for identifying a stress-free reference configuration in image-based biomechanical modeling” Journal of Biomechanics 58 (2017) 227-231.

CONTROL_REFINE_ALE**CONTROL*****CONTROL_REFINE_ALE**

Purpose: Refine ALE hexahedral solid elements locally. Each parent element is replaced by 8 child elements with a volume equal to 1/8th the parent volume. If only the 1st card is defined, the refinement occurs during the initialization. The 2nd card defines a criterion CRITRF to automatically refine the elements during the run. If the 3rd card is defined, the refinement can be removed if a criterion CRITRM is reached, that is, the child elements can be replaced by their parents.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	TYPE	NLVL	MMSID	IBOX	IELOUT		
Type	I	I	I	I	I	I		
Default	none	0	1	0	0	0		

Remaining cards are optional.[†]

Automatic refinement card. Optional card for activating automatic refinement whereby each element satisfying certain criteria is replaced by a cluster of 8 child elements (2 × 2 × 2 elements)

Card 2	1	2	3	4	5	6	7	8
Variable	NTOTRF	NCYCRF	CRITRF	VALRF	BEGRF	ENDRF	LAYRF	DELAYRF
Type	I	F	I	F	F	F	I	F
Default	0	0.0	0	0.0	0.0	0.0	0	0.0

*CONTROL

*CONTROL_REFINE_ALE

Automatic Refinement Remove Card. Optional card for activating automatic refinement removal whereby, when, for a cluster of 8 child elements, certain criteria are satisfied, the cluster ($2 \times 2 \times 2$ child elements) is replaced by its parent.

Card 3	1	2	3	4	5	6	7	8
Variable	MAXRM	NCYCRM	CRITRM	VALRM	BEGRM	ENDRM	MMSRM	DELAYRM
Type	I	F	I	F	F	F	I	F
Default	0	0.0	0	0.0	0.0	0.0	0	0.0

VARIABLE	DESCRIPTION
ID	Set ID. LT.0: Parent elements can be hidden in LS-PrePost as they are replaced by their children.
TYPE	Set type: EQ.0: ALE part set EQ.1: ALE part EQ.2: Lagrangian part set coupled to ALE (see Remarks 1 and 2) EQ.3: Lagrangian part coupled to ALE (see Remarks 1 and 2) EQ.4: Lagrangian shell set coupled to ALE (see Remarks 1 and 2) EQ.5: ALE solid set
NLVL	Number of refinement levels (see Remark 3).
MMSID	Multi-Material Set ID (see Remark 4): LT.0: Only ALE elements with all the multi-material groups listed in *SET_MULTI-MATERIAL_GROUP_LIST can be refined (or removed otherwise) GT.0: ALE elements with at least one of the multi-material groups can be refined (or removed)
IBOX	Box ID (See *DEFINE_BOX) defining a region in which the ALE elements are refined. The options LOCAL and ADAPTIVE for *DEFINE_BOX are supported.
IELOUT	Flag to handle child data in elout (see Remark 9).

VARIABLE	DESCRIPTION
NTOTRF	Total number of ALE elements to refine (see Remark 5): GT.0: Number of elements to refine EQ.0: Number of solid elements in IBOX (see Remark 2) EQ.-1: Add clusters of 8 solids for the refinement during the run as needed. LT.-1: NTOTRF is the ID of *CONTROL_REFINE_MPP_DISTRIBUTION that computes the number of extra elements required by the processors.
NCYCRF	Number of cycles between each refinement. LT.0: NCYCRF is the time interval.
CRITRF	Refinement criterion: EQ.-3: Volume fraction < VALRF EQ.-2: Relative volume (V/V_0) > VALRF EQ.-1: Pressure < VALRF EQ.0: Static refinement (as if only Card 1 is defined) EQ.1: Pressure > VALRF EQ.2: Relative volume (V/V_0) < VALRF EQ.3: Volume fraction > VALRF EQ.5: User defined criterion. The fortran routine <code>alerfn_criteria5</code> in the file <code>dynrfn_user.f</code> should be used to develop the criterion. The file is part of the general package <code>usermat</code> .
VALRF	Criterion value to reach for the refinement
BEGRF	Time to begin the refinement
ENDRF	Time to end the refinement
LAYRF	Number of element layers to refine around an element satisfying the refinement criterion (see Remark 6).
DELAYRF	Period of time after removing the refinement of an element, during which this element will not be refined again.

*CONTROL

*CONTROL_REFINE_ALE

VARIABLE	DESCRIPTION
MAXRM	Maximum number of child clusters ($2 \times 2 \times 2$ child elements) to remove (see Remark 8): LT.0: For the whole run GT.0: Every NCYCRM cycles
NCYCRM	Number of cycles between each check for refinement removal. LT.0: $ NCYCRM $ is the time interval.
CRITRM	Criterion for refinement removal (a negative CRITRM reverses the conditions below): EQ.-3: Volume fraction > VALRM EQ.-2: Relative volume $(V/V_0) < VALRM$ EQ.-1: Pressure > VALRM EQ.0: No refinement removal (as if only Cards 1 and 2 are defined) EQ.1: Pressure < VALRM EQ.2: Relative volume $(V/V_0) > VALRM$ EQ.3: Volume fraction < VALRM EQ.5: User defined criterion. The fortran routine <code>alermv_criteria5</code> in the file <code>dynrfn_user.f</code> should be used to develop the criterion. The file is part of the general package <code>usermat</code> .
VALRM	Criterion value to reach in each child element of a cluster for its removal (child elements replaced by parent element).
BEGRM	Time to begin the check for refinement removal. LT.0: $ BEGRM $ represents a critical percent of NTOTRF below which the check for refinement removal should begin ($0.0 < BEGRM < 1.0$). See Remark 7 .
ENDRM	Time to end the check for refinement removal
MMSRM	Multi-Material Set ID for the refinement removal. See Remark 4 . LT.0: $ MMSRM $ represents the radius of a sphere centered on a newly refined element, in which the refinement can not be removed.

VARIABLE	DESCRIPTION
DELAYRM	Period of time after refining an element, during which this refinement will not be removed

Remarks:

1. **First Card Definition Only and TYPE.** If only the 1st card is defined, only TYPE = 0, 1, and 5 can be defined.
2. **Lagrangian Coupling.** *CONSTRAINED_LAGRANGE_IN_SOLID needs to be defined for TYPE = 2, 3, and 4. If an ALE element has at least one coupling point (see NQUAD in *CONSTRAINED_LAGRANGE_IN_SOLID), this element will be selected to be refined (or removed). The number of elements to refine is computed during the initialization. NTOTRF can be zero. Otherwise it can be used to add more elements.
3. **Refinement Levels.** If NLVL = 1, there is only one level of refinement: the ALE elements in *ELEMENT_SOLID are the only ones to be replaced by clusters of 8 child elements. If NLVL > 1, there are several levels of refinement; not only the initial ALE elements in *ELEMENT_SOLID are refined but also their child elements.
4. **Multi-Material Groups.** If only Card 1 is defined, a multi-material set ID is not used. It can be left as zero. For Cards 2 and 3, MMSID is the ID of *SET_MULTI-MATERIAL_GROUP_LIST in which the multi-material group IDs (as defined in *ALE_MULTI-MATERIAL_GROUP) are listed to select the ALE elements to be refined (or removed). If MMSID < 0, only mixed ALE elements containing all the multi-material groups can be refined. Otherwise clusters of 8 elements without a mix of the listed multi-material groups can be removed.

If MMSRM = 0, all the child clusters meeting the removal criteria can be deleted. If MMSRM is defined, only ALE child clusters fully filled by the multi-material groups listed by the set MMSRM can be removed (if the refinement removal criterion is reached).

5. **Number of ALE Elements to be Refined.** NTOTRF defines the total number of ALE elements to be refined. For example, NTOTRF = 100 with NLVL = 1 means that only 100 ALE elements can be replaced by 800 finer ALE elements (or 100 clusters of 8 child elements). For NLVL = 2, these 800 elements can be replaced by 6400 finer elements.
6. **Neighbor Element Refinement.** If an element is refined, the neighbor elements can be refined as well. LAYRF defines the number of neighbor layers to refine. For example:

- a) with LAYRF = 1 an element that meets the refinement criterion at the center of a block of $3 \times 3 \times 3$ elements will cause the refinement of these 27 elements.
 - b) with LAYRF = 2 an element that meets the refinement criterion at the center of a block of $5 \times 5 \times 5$ elements will cause the refinement of these 125 elements.
 - c) with LAYRF = 3 an element that meets the refinement criterion at the center of a block of $7 \times 7 \times 7$ elements will cause the refinement of these 343 elements.
7. **Refinement Removal Activation.** If BEGRM < 0, the check for refinement removal is activated when the number of 8 element clusters for the refinement is below a limit defined by $|BEGRM| \times NTOTRF$. If $|BEGRM| = 0.1$, then the check for refinement removal starts when 90% of the stock of clusters is used for the refinement.
8. **Number of Child Clusters to Remove.** MAXRM < 0 defines a total number of child clusters to remove for the whole run. If positive, MAXRM defines an upper limit for the number of child clusters to remove every NCYCRM cycles.
9. **Output Child Element Data.** If only Card 1 is defined, the child data is always output in elout, that is, IELOUT is always activated. Since the refinement occurs during the initialization, every refined element is replaced by its 8 children in the set defined for *DATABASE_ELOUT.

If the optional cards are included, the child data is output in elout if the IELOUT flag is set to 1. Since the refinement occurs during the run, the parent IDs in the set defined for *DATABASE_ELOUT are duplicated 8^{NLVL} times. The points of integration in the elout file are incremented to differentiate the child contributions to the database.

CONTROL_REFINE_ALE2D**CONTROL*****CONTROL_REFINE_ALE2D**

Purpose: Refine ALE quadrilateral shell elements locally. Each parent element is replaced by 4 child elements with a volume equal to 1/4th the parent volume. If only the 1st card is defined, the refinement occurs during the initialization. The 2nd card defines a criterion CRITRF to automatically refine the elements during the run. If the 3rd card is defined, the refinement can be removed if a criterion CRITRM is reached, that is, the child elements can be replaced by their parents.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	TYPE	NLVL	MMSID	IBOX	IELOUT	INITRN	
Type	I	I	I	I	I	I	I	
Default	none	0	1	0	0	0	0	

Remaining cards are optional.[†]

Automatic refinement card. Optional card for activating automatic refinement whereby each element satisfying certain criteria is replaced by a cluster of 4 child elements (2 × 2 child elements)

Card 2	1	2	3	4	5	6	7	8
Variable	NTOTRF	NCYCRF	CRITRF	VALRF	BEGRF	ENDRF	LAYRF	
Type	I	F	I	F	F	F	I	
Default	0	0.0	0	0.0	0.0	0.0	0	

*CONTROL

*CONTROL_REFINE_ALE2D

Automatic Refinement Remove Card. Optional card for activating automatic refinement removal whereby, when, for a cluster of 4 child elements, certain criteria are satisfied the cluster (2×2 child elements) is replaced by its parent.

Card 3	1	2	3	4	5	6	7	8
Variable	MAXRM	NCYCRM	CRITRM	VALRM	BEGRM	ENDRM	MMSRM	
Type	I	F	I	F	F	F	I	
Default	0	0.0	0	0.0	0.0	0.0	0	

VARIABLE	DESCRIPTION
ID	Set ID. LT.0: Parent elements can be hidden in LS-PrePost as they are replaced by their children.
TYPE	Set type: EQ.0: ALE part set EQ.1: ALE part EQ.2: Lagrangian part set coupled to ALE (see Remarks 1 and 2) EQ.3: Lagrangian part coupled to ALE (see Remarks 1 and 2) EQ.4: Lagrangian beam set coupled to ALE (see Remarks 1 and 2) EQ.5: ALE shell set
NLVL	Number of refinement levels (see Remark 3)
MMSID	Multi-Material Set ID (see Remark 4): LT.0: Only ALE elements with all the multi-material groups listed in *SET_MULTI-MATERIAL_GROUP_LIST can be refined (or removed otherwise) GT.0: ALE elements with at least one of the multi-material groups can be refined (or removed)
IBOX	Box ID (see *DEFINE_BOX) defining a region in which the ALE elements are refined. The options LOCAL and ADAPTIVE for *DEFINE_BOX are supported.

VARIABLE	DESCRIPTION
IELOUT	Flag to handle child data in elout (see Remark 9)
INITRN	Flag to add mesh transitions for CRITRF = 0: EQ.0: Do not add mesh transitions. EQ.1: Add mesh transitions.
NTOTRF	Total number of ALE elements to refine (see Remark 5): GT.0: Number of elements to refine EQ.0: Number of shell elements in IBOX (see Remark 2) EQ.-1: Add clusters of 4 shells for the refinement during the run as needed
NCYCRF	Number of cycles between each refinement. LT.0: -NCYCRF is the time interval.
CRITRF	Refinement criterion: EQ.-3: Volume fraction < VALRF EQ.-2: Relative volume (V/V_0) > VALRF EQ.-1: Pressure < VALRF EQ.0: Static refinement (as if only the 1st card is defined) EQ.1: Pressure > VALRF EQ.2: Relative volume (V/V_0) < VALRF EQ.3: Volume fraction > VALRF EQ.5: User defined criterion. The fortran routine al2rfn_criteria5 in the file dynrfn_user.f should be used to develop the criterion. The file is part of the general package usermat.
VALRF	Criterion value to reach for the refinement
BEGRF	Time to begin the refinement
ENDRF	Time to end the refinement
LAYRF	Number of element layers to refine around an element that has satisfied the refinement criterion (see Remark 6)

*CONTROL

*CONTROL_REFINE_ALE2D

VARIABLE	DESCRIPTION
MAXRM	Maximum number of child clusters (cluster is 2×2 child elements) to remove (see Remark 8): LT.0: For the whole run GT.0: Every NCYCRM cycles
NCYCRM	Number of cycles between each check for refinement removal. LT.0: $ NCYCRM $ is the time interval.
CRITRM	Criterion for refinement removal (a negative CRITRM reverses the conditions below): EQ.3: Volume fraction > VALRM EQ.2: Relative volume (V/V_0) < VALRM EQ.1: Pressure > VALRM EQ.0: No refinement removal (as if only Cards 1 and 2 are defined) EQ.1: Pressure < VALRM EQ.2: Relative volume (V/V_0) > VALRM EQ.3: Volume fraction < VALRM EQ.5: User defined criterion. The fortran routine al2rmv_criteria5 in the file dynrfn_user.f should be used to develop the criterion. The file is part of the general package usermat.
VALRM	Criterion value to reach in each child element of a cluster for its removal (child elements of a cluster replaced by parent element)
BEGRM	Time to begin the check for refinement removal. LT.0: $ BEGRM $ represents a critical percent of NTOTRF below which the check for refinement removal should begin ($0.0 < BEGRM < 1.0$). See Remark 6 .
ENDRM	Time to end the check for refinement removal
MMSRM	Multi-Material Set ID for the refinement removal. See Remark 4 .

Remarks:

1. **First card definition only and TYPE.** If only the 1st card is defined, only TYPE = 0, 1, and 5 can be defined.

2. **Lagrangian coupling.** [*CONSTRAINED_LAGRANGE_IN_SOLID](#) needs to be defined for TYPE = 2, 3, and 4. If an ALE element has at least one coupling point (see NQUAD in [*CONSTRAINED_LAGRANGE_IN_SOLID](#)), this element will be selected to be refined (or removed).
3. **Refinement levels.** If NLVL = 1, there is only one level of refinement; the ALE elements in [*ELEMENT_SHELL](#) are the only ones to be replaced by clusters of 4 child elements. If NLVL > 1, there are several levels of refinement; not only the initial ALE elements in [*ELEMENT_SHELL](#) are refined but also their child elements. If NLVL = 2, for example, an initial ALE element can be replaced by a cluster of 16 child elements.
4. **Multi-material groups.** If only Card 1 is defined, a multi-material set ID is not used. It can be left as zero. For Cards 2 and 3, MMSID is the ID of [*SET_MULTI-MATERIAL_GROUP_LIST](#) in which the multi-material group IDs (as defined in [*ALE_MULTI-MATERIAL_GROUP](#)) are listed to select the ALE elements to be refined (or removed). If MMSID < 0, only mixed ALE elements containing all the multi-material groups can be refined. Otherwise, clusters of 4 elements without a mix of the listed multi-material groups can be removed.

If MMSRM = 0, all the child clusters meeting the removal criterion can be deleted. If MMSRM is defined, only ALE child elements fully filled by the multi-material groups listed by the set MMSRM can be removed (if the refinement removal criterion is reached).

5. **Number of ALE elements to be refined.** NTOTRF defines the total number of ALE elements to be refined. For example, NTOTRF = 100 means that only 100 ALE elements will be replaced by 400 ALE finer elements (or 100 clusters of 4 child elements). For NLVL = 2, these 400 elements can be replaced by 1600 finer elements.
6. **Neighbor element fefinement.** If an element is refined, neighbor elements can be refined as well. LAYRF defines the number of neighbor layers to refine. For example:
 - a) with LAYRF = 1 an element that meets the refinement criterion at the center of a block of 3×3 elements will cause the refinement of these 9 elements.
 - b) with LAYRF = 2 an element that meets the refinement criterion at the center of a block of 5×5 elements will cause the refinement of these 25 elements.
 - c) with LAYRF = 3 an element that meets the refinement criterion at the center of a block of 7×7 elements will cause the refinement of these 49 elements
7. **Refinement removal activation.** If BEGRM < 0, the check for refinement removal is activated when the number of four element clusters for the refinement

is below a limit defined by $|BEGRM| \times NTOTRF$. If $|BEGRM| = 0.1$, then the check for refinement removal starts when 90% of the stock of clusters is used for the refinement.

8. **Number of child clusters to remove.** MAXRM < 0 is the exact opposite of NTOTRF > 0; it defines a total number of child clusters to remove for the whole run. If positive, MAXRM defines an upper limit for the number of child clusters to remove every NCYCRM cycles
9. **Output child element data.** If only Card 1 is defined, the child data is output in elout, meaning IELOUT is always activated. Because the refinement occurs during the initialization phase, every refined element is replaced by its 4 children in the set defined for [***DATABASE_ELOUT**](#).

If the optional cards are included, the child data is output in elout if the IELOUT is set to 1. Since the refinement occurs during the run, the parent IDs in the set defined for [***DATABASE_ELOUT**](#) are duplicated 4^{NLVL} times. The points of integration in the elout file are incremented to differentiate the child contributions to the database.

***CONTROL_REFINE_MPP_DISTRIBUTION**

Purpose: Distribute the elements for the refinement over the MPP processes. This keyword addresses the following situation:

If TYPE = 2, 3, or 4 in *CONTROL_REFINE_ALE, the refinement occurs around a structure. The number of elements for this refinement is computed for each process according the initial position of the structure in each MPP subdomain (after the MPP decomposition of the ALE mesh during phase 3 of the initialization, each process has a subdomain that is a portion of the ALE mesh). If the structure is not in a subdomain, the related process receives no extra element for the refinement. If the structure moves into this subdomain during the computation, the refinement around the structure can not occur. To avoid this problem, the structure can be considered within a box (the structure maxima and minima give the box dimensions and positions). This box moves and expands during the computation to keep the structure inside. An estimation of the maximal displacement and expansion will allow the code to evaluate which subdomains the structure will likely cross and how many extra elements a process may need to carry out the refinement.

The computation of the number of extra elements per process occurs in 2 steps:

- If a file called refine_mpp_distribution does not exist in the working directory, it will be created to list the number of elements by process. Each line in this file matches a process rank (starting from 0). After phase 3 of the MPP decomposition, the run terminates as if *CONTROL_MPP_DECOMPOSITION_SHOW was activated.
- The model can be run again and the file refine_mpp_distribution will be read to allocate the memory for the extra elements and distribute them across the processes.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	DX	DY	DZ	EX	EY	EZ	
Type	I	F	F	F	F	F	F	
Default	none	0.0	0.0	0.0	1.0	1.0	1.0	

VARIABLE**DESCRIPTION**

ID

ID = -NTOTRF in *CONTROL_REFINE_ALE

VARIABLE	DESCRIPTION
DX	Dimensionless x -displacement of the box. (see Remark 1).
DY	Dimensionless y -displacement of the box. (see Remark 1).
DZ	Dimensionless z -displacement of the box. (see Remark 1).
EX	Dimensionless x -expansion of the box. (see Remark 2).
EY	Dimensionless y -expansion of the box. (see Remark 2).
EZ	Dimensionless z -expansion of the box. (see Remark 2).

Remarks:

1. **Box Displacements.** DX, DY and DZ are the maximal displacements of the box center. These displacements are ratio of the box dimensions. If, for example, the largest length of the structure in the x -direction is 10 m and the maximal displacement in this direction is 2 m, DX should be equal to 0.2.
2. **Maximal Box Dilations.** EX, EY and EZ represent the maximal dilatations of the box in each direction. These expansions are the ratio of the box dimensions. The box expands around its center. If, for example, the maximal thickness of a structure along the z -direction is 1 cm and the structure deforms 30 times the thickness in z -direction, EZ should be equal to 30, and DZ = 15 accounts for the box center motion. The xy -plane is a plane of symmetry for this deformation; DZ can be zero.

CONTROL_REFINE_SHELL**CONTROL*****CONTROL_REFINE_SHELL**

Purpose: Refine quadrilateral shell elements locally. Each parent element is replaced by 4 child elements, each with a volume equal to 1/4 the parent volume.

Card Summary:

Card 1. This card is required. If only this card is defined, the refinement occurs during the initialization.

ID	TYPE	NLVL	IBOX	IELOUT			
----	------	------	------	--------	--	--	--

Card 2. This card is optional. If defined, this card sets a criterion, CRITRF, to automatically refine the elements during the run.

NTOTRF	NCYCRF	CRITRF	VALRF	BEGRF	ENDRF	LAYRF	
--------	--------	--------	-------	-------	-------	-------	--

Card 3. This card is optional. If defined, this card sets a criterion, CRITRM, for which the refinement can be removed; that is, the child elements can be replaced by their parents.

MAXRM	NCYCRM	CRITRM	VALRM	BEGRM	ENDRM		
-------	--------	--------	-------	-------	-------	--	--

Data Cards:

Card 1	1	2	3	4	5	6	7	8
Variable	ID	TYPE	NLVL	IBOX	IELOUT			
Type	I	I	I	I	I			
Default	none	0	1	0	0			

VARIABLE	DESCRIPTION
ID	Set ID. LT.0: parent elements can be hidden in LS-PrePost as they are replaced by their children.
TYPE	Set type: EQ.0: part set

*CONTROL

*CONTROL_REFINE_SHELL

VARIABLE	DESCRIPTION
	EQ.1: part EQ.2: shell set
NLVL	Number of refinement levels (see Remark 1)
IBOX	Box ID (See *DEFINE_BOX) defining a region in which the elements are refined. The options LOCAL and ADAPTIVE for *DEFINE_BOX are supported.
IELOUT	Flag to handle child data in the elout file (see Remark 5)

Automatic refinement card. Optional card for activating automatic refinement whereby each element satisfying certain criteria is replaced by a cluster of 4 child elements

Card 2	1	2	3	4	5	6	7	8
Variable	NTOTRF	NCYCRF	CRITRF	VALRF	BEGRF	ENDRF	LAYRF	
Type	I	F	I	F	F	F	I	
Default	0	0.0	0	0.0	0.0	0.0	0	

VARIABLE	DESCRIPTION
NTOTRF	Total number of elements to refine (see Remark 2): GT.0: number of elements to refine EQ.0: number of shell elements in IBOX EQ.-1: add clusters of 4 shells for the refinement during the run.
NCYCRF	Number of cycles between each refinement. LT.0: NCYCRF is the time interval.
CRITRF	Refinement criterion: EQ.-4: criterion similar to ADPTYP = 4 in *CONTROL_ADAPTIVE with VALRF replacing ADPTOL (VALRF = ADPTOL), except refinement occurs when the shell error in the energy norm is less than VALRF/100 times the mean energy norm within the part

VARIABLE	DESCRIPTION
	EQ.-3: von Mises stress < VALRF
	EQ.-1: pressure < VALRF
	EQ.0: static refinement (as if only the 1st card is defined)
	EQ.1: pressure > VALRF
	EQ.2: undefined
	EQ.3: von Mises stress > VALRF
	EQ.4: criterion similar to ADPTYP = 4 in *CONTROL_ADAPTIVE with VALRF replacing ADPTOL (VALRF = ADPTOL)
	EQ.5: user defined criterion. The fortran routine shlrfn_criteria5 in the file dynrfn_user.f should be used to develop the criterion. The file is part of the general package usemat.
VALRF	Criterion value to reach for the refinement
BEGRF	Time to begin the refinement
ENDRF	Time to end the refinement
LAYRF	Number of element layers to refine around an element reaching the refinement criterion (see Remark 3)

Automatic Refinement Remove Card. Optional card for activating automatic refinement removal whereby, when, for a cluster of 4 child elements, certain criteria are satisfied the clusters is replaced by its parent.

Card 3	1	2	3	4	5	6	7	8
Variable	MAXRM	NCYCRM	CRITRM	VALRM	BEGRM	ENDRM		
Type	I	F	I	F	F	F		
Default	0	0.0	0	0.0	0.0	0.0		

CONTROL**CONTROL_REFINE_SHELL**

VARIABLE	DESCRIPTION
MAXRM	Maximum number of child clusters to remove: LT.0: $ MAXRM $ is the total number of child clusters to be removed for the whole run, GT.0: max number removed every NCYCRM cycles
NCYCRM	Number of cycles between each check for refinement removal: LT.0: $ NCYCRM $ is the time interval.
CRITRM	Criterion for refinement removal: EQ.-4: criterion similar to ADPTYP = 4 in *CONTROL_ADAPTIVE with VALRM replacing ADPTOL (VALRM = ADPTOL) EQ.-3: von Mises stress > VALRM EQ.-1: pressure > VALRM EQ.0: no refinement removal (as if only the 1st and 2nd cards are defined) EQ.1: pressure < VALRM EQ.2: undefined EQ.3: von Mises stress < VALRM EQ.4: criterion similar to ADPTYP = 4 in *CONTROL_ADAPTIVE with VALRM replacing ADPTOL (VALRM = ADPTOL), except refinement occurs when the shell error in the energy norm is less than $VALRM/100$ times the mean energy norm within the part EQ.5: user defined criterion. The fortran routine shlrmv_criteria5 in the file dynrfn_user.f should be used to develop the criterion. The file is part of the general package usermat.
VALRM	Criterion value to reach in each child elements of a cluster for its removal (child elements replaced by parent element)
BEGRM	Time to begin the check for refinement removal. LT.0: $ BEGRM $ represents a critical percent of NTOTRF below which the check for refinement removal should begin ($0.0 < BEGRM < 1.0$). See Remark 4 .

VARIABLE	DESCRIPTION
ENDRM	Time to end the check for refinement removal

Remarks:

1. **Refinement Level.** If NLVL = 1, there is only one level of refinement; the elements in *ELEMENT_SHELL are the only ones to be replaced by clusters of 4 child elements. If NLVL > 1, there are several levels of refinement; not only the initial elements in *ELEMENT_SHELL are refined but also their child elements. If NLVL = 2 for example, the initial elements can be replaced by clusters of 16 child elements.
2. **Number of Parent Elements.** NTOTRF > 0 defines the total number of elements to be refined. For example, NTOTRF = 100 with NLVL = 1 means that only 100 elements can be replaced by 400 finer elements (or 100 clusters of 4 child elements). For NLVL = 2, these 400 elements can be replaced by 1600 finer elements.
3. **Neighboring Element Refinement.** If an element is refined, neighboring elements may be refined as well. LAYRF defines the number of neighbor layers to refine. For example:
 - a) with LAYRF = 1 an element that meets the refinement criterion at the center of a block of 3×3 elements will cause the refinement of these 9 elements.
 - b) with LAYRF = 2 an element that meets the refinement criterion at the center of a block of 5×5 elements will cause the refinement of these 25 elements.
 - c) with LAYRF = 3 an element that meets the refinement criterion at the center of a block of 7×7 elements will cause the refinement of these 49 elements
4. **Refinement Removal Activation.** If BEGRM < 0, the check for refinement removal is activated when the number of 4-element clusters for the refinement is below a limit defined by $|BEGRM| \times NTOTRF$. If $|BEGRM| = 0.1$, the check for refinement removal starts when 90% of the stock of clusters is used for the refinement.
5. **IELOUT.** If only the 1st card is defined, the code for IELOUT is always activated. Since the refinement occurs during the initialization, every refined element is replaced by its 4 children in the set defined for *DATABASE_ELOUT.

If more than the 1st card is defined, the code for IELOUT is activated if the flag is equal to 1. Since the refinement occurs during the run, the parent IDs in the set defined for *DATABASE_ELOUT are duplicated 4^{NLVL} times. The points of

***CONTROL**

***CONTROL_REFINE_SHELL**

integration in the elout file are incremented to differentiate the child contributions to the database.

CONTROL_REFINE_SOLID**CONTROL*****CONTROL_REFINE_SOLID**

Purpose: Refine hexahedral solid elements locally. Each parent element meeting the refinement criterion is replaced by 8 child elements with a volume equal to 1/8th the parent volume. If only the 1st card is defined, the refinement occurs during the initialization. The 2nd card defines a criterion CRITRF to automatically refine the elements during the run. If the 3rd card is defined, the refinement can be removed if a criterion CRITRM is reached: the child elements can be replaced by their parents.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	TYPE	NLVL	IBOX	IELOUT			
Type	I	I	I	I	I			
Default	none	0	1	0	0			

Remaining cards are optional.[†]

Automatic refinement card. Optional card for activating automatic refinement whereby each element satisfying certain criteria is replaced by a cluster of 8 child elements

Card 2	1	2	3	4	5	6	7	8
Variable	NTOTRF	NCYCRF	CRITRF	VALRF	BEGRF	ENDRF	LAYRF	
Type	I	F	I	F	F	F	I	
Default	0	0.0	0	0.0	0.0	0.0	0	

*CONTROL

*CONTROL_REFINE_SOLID

Automatic Refinement Remove Card. Optional card for activating automatic refinement removal whereby, when, for a cluster of 8 child elements, certain criteria are satisfied the clusters is replaced by its parent.

Card 3	1	2	3	4	5	6	7	8
Variable	MAXRM	NCYCRM	CRITRM	VALRM	BEGRM	ENDRM		
Type	I	F	I	F	F	F		
Default	0	0.0	0	0.0	0.0	0.0		

VARIABLE	DESCRIPTION
ID	Set ID. LT.0: parent elements can be hidden in LS-PrePost as they are replaced by their children.
TYPE	Set type: EQ.0: part set EQ.1: part EQ.2: solid set
NLVL	Number of refinement levels. See Remark 1 .
IBOX	Box ID (see *DEFINE_BOX) defining a region in which the elements are refined. The keyword options LOCAL and ADAPTIVE for *DEFINE_BOX are supported.
IELOUT	Flag to handle child data in elout. See Remarks 6 and 7 .
NTOTRF	Total number of elements to refine. See Remark 2 . GT.0: number of elements to refine EQ.0: NTOTRF = number of solid elements in IBOX EQ.-1: add clusters of 8 solids for the refinement during the run.
NCYCRF	Number of cycles between each refinement. LT.0: NCYCRF is the time interval.

VARIABLE	DESCRIPTION
CRITRF	<p>Refinement criterion:</p> <p>EQ.-5: user defined criterion as defined in subroutine <code>sldrfin_criteria5</code> in the file <code>dynrfn_user.f</code>, which is included in the general <code>usermat</code> package. “Effective plastic strain” < VALRF is the criterion if <code>sldrfin_criteria5</code> is not modified by the user.</p> <p>EQ.-3: von Mises stress < VALRF</p> <p>EQ.-1: pressure < VALRF</p> <p>EQ.0: static refinement as if only the 1st card is defined</p> <p>EQ.1: pressure > VALRF</p> <p>EQ.2: no refinement occurs.</p> <p>EQ.3: von Mises stress > VALRF</p> <p>EQ.5: user defined criterion as defined in subroutine <code>sldrfin_criteria5</code> in the file <code>dynrfn_user.f</code>, which is included in the general <code>usermat</code> package. “Effective plastic strain” > VALRF is the criterion if <code>sldrfin_criteria5</code> is not modified by the user.</p>
VALRF	Value of the refinement criterion that triggers refinement
BEGRF	Time to begin the refinement
ENDRF	Time to end the refinement
LAYRF	Number of element layers to refine around an element reaching the refinement criterion. See Remark 3 .
MAXRM	<p>Maximum number of child clusters to remove. See Remark 5.</p> <p>LT.0: for the whole run</p> <p>GT.0: every NCYCRM cycles</p>
NCYCRM	<p>Number of cycles between each check for refinement removal:</p> <p>LT.0: NCYCRM is the time interval.</p>
CRITRM	<p>Criterion for removal of refinement:</p> <p>EQ.-3: von Mises stress > VALRM</p> <p>EQ.-1: pressure > VALRM</p>

*CONTROL

*CONTROL_REFINE_SOLID

VARIABLE	DESCRIPTION
	EQ.0: no removal of refinement as if only the 1st and 2nd cards are defined. EQ.1: pressure < VALRM EQ.2: same effect as CRITRM = 0 EQ.3: von Mises stress < VALRM EQ.5: user defined criterion as defined in subroutine sldrmv_criteria5 in the file dynrfn_user.f.
VALRM	Criterion value to reach in each child element of a cluster for its removal (replace child elements with parent element)
BEGRM	Time to begin check for refinement removal: LT.0: BEGRM represents a critical percent of NTOTRF below which the check for refinement removal should begin ($0.0 < BEGRM < 1.0$). See Remark 4 .
ENDRM	Time to end the check for refinement removal

Remarks:

1. **Number of Refinement Levels.** If NLVL = 1, there is only one level of refinement: the elements in *ELEMENT_SOLID are the only ones to be replaced by clusters of 8 child elements. If NLVL > 1, there are several levels of refinement: not only the initial elements in *ELEMENT_SOLID are refined but also their child elements. If NLVL = 2 for example, the initial elements can be replaced by clusters of 64 child elements.
2. **Maximum Number of Elements to Refine.** NTOTRF defines the total number of elements to be refined. For example, NTOTRF = 100 with NLVL = 1 means that only 100 elements can be replaced by 800 finer elements (or 100 clusters of 8 child elements). For NLVL = 2, these 800 elements can be replaced by 6400 finer elements.
3. **Number of Layers to Refine.** If an element is refined, it is possible to refine the neighbor elements as well. LAYRF defines the number of neighbor layers to refine. For example:
 - a) with LAYRF = 1 an element that meets the refinement criterion at the center of a block of $3 \times 3 \times 3$ elements will cause the refinement of these 27 elements.

- b) with LAYRF = 2 an element that meets the refinement criterion at the center of a block of $5 \times 5 \times 5$ elements will cause the refinement of these 125 elements.
 - c) with LAYRF = 3 an element that meets the refinement criterion at the center of a block of $7 \times 7 \times 7$ elements will cause the refinement of these 343 elements.
4. **Onset of Refinement Removal.** If BEGRM < 0, the check for refinement removal is activated when the number of 8-element clusters for the refinement is below a limit defined by $|BEGRM| \times NTOTRF$. If $|BEGRM| = 0.1$, it means that the check for refinement removal starts when 90% of the stock of clusters are used for the refinement.
 5. **Maximum Refinement Removal.** MAXRM < 0 defines a total number of child clusters to remove for the whole run. If positive, MAXRM defines an upper limit for the number of child clusters to remove every NCYCRM cycles.
 6. **The elout Database and Initial Refinement.** If only the 1st card is defined, the code for IELOUT is always activated. Since the refinement occurs during the initialization, every refined element is replaced by its 8 children in the set defined for *DATABASE_ELOUT.
 7. **The elout Database and Refinement at Run Time.** If there is more than 1 line, the code for IELOUT is activated if the flag is equal to 1. Since the refinement occurs during the run, the parent IDs in the set defined for *DATABASE_ELOUT are duplicated 8^{NLVL} times. The points of integration in the elout file are incremented to differentiate the child contributions to the database.
 8. **Contact.** When a refined part involves a contact that is not *CONTACT_ERODING_SINGLE_SURFACE with SOFT = 2, that contact is automatically replaced with *CONTACT_ERODING_SINGLE_SURFACE with SOFT = 2.

*CONTROL

*CONTROL_REMESHING

*CONTROL_REMESHING_{OPTION}

Available options include:

<BLANK>

EFG

Purpose: Provide control over the adaptive remeshing of solids. See also *CONTROL-ADAPTIVE and the variable ADPOPT in *PART.

There are two different types of 3D solid adaptivity affected by *CONTROL_REMESHING.

1. Tetrahedral adaptivity is invoked by ADPOPT = 2 in *PART. The initial mesh must be comprised of only tetrahedrons. We recommend FEM solid formulations 10 and 13 and EFG solid formulation 42 (see details in *SECTION_SOLID). When adaptivity is triggered by any of several available criteria (volume loss, minimum time step, etc.), the surface mesh consisting of triangles is first improved, subject to the element size criteria set forth in *CONTROL_REMESHING. Then, the automatic remesher creates the tetrahedral elements of the new mesh starting from the improved surface mesh. When the EFG option is included, additional adaptivity criteria may be invoked using Card 2 and, optionally, Card 3.
2. Axisymmetric adaptivity, sometimes called orbital adaptivity, in which remeshing is done with hexahedral and pentahedral elements, is invoked by ADPOPT = 3 in *PART. Only the variables RMIN, RMAX, CID, and SEGANG are used in this type of adaptivity. Though EFG solid formulation 41 is permissible, the EFG option of *CONTROL_REMESHING does not apply for this type of adaptivity.

Card 1	1	2	3	4	5	6	7	8
Variable	RMIN	RMAX	VF_LOSS	MFRAC	DT_MIN	ICURV	CID	SEGANG
Type	F	F	F	F	F	I	I	F
Default	none	none	1.0	0.0	0.	4	0	0.0

CONTROL_REMESHING**CONTROL**

Additional card for EFG option or FEM tetrahedral adaptivity. With FEM tetrahedral adaptivity, only MM option is available.

Card 2	1	2	3	4	5	6	7	8
Variable	IVT	IAT	IAAT	ICONOPT	MM			
Type	I	I	I	I	I			
Default	1	0	0	0	0			

Second additional card for EFG option. This card is optional.

Card 3	1	2	3	4	5	6	7	8
Variable	IAT1	IAT2	IAT3					
Type	F	F	F					
Default	10^{20}	10^{20}	10^{20}					

VARIABLE	DESCRIPTION
RMIN	Minimum edge length for the surface mesh surrounding the parts which should be remeshed. See Remark 1 . LT.0.0: RMIN is the ID of a load curve controlling the time-dependent minimum edge length for 3D <i>r</i> -adaptivity.
RMAX	Maximum edge length for the surface mesh surrounding the parts which should be remeshed. See Remark 1 .
VF_LOSS	Volume fraction loss required in a type 10/13 tetrahedral elements to trigger a remesh. For the type 13 solid elements, the pressures are computed at the nodal points; therefore, it is possible for overall volume to be conserved but for individual tetrahedrons to experience a significant volume loss or gain. The volume loss can lead to numerical problems. Recommended values for VF_LOSS in the range of 0.10 to 0.30 may be reasonable.
MFRAC	Mass ratio gain during mass scaling required for triggering a remesh. For a one percent increase in mass, set MFAC = 0.010. This variable applies to both to general three-dimensional

*CONTROL

*CONTROL_REMESHING

VARIABLE	DESCRIPTION
	tetrahedral remeshing and to three dimensional axisymmetric remeshing.
DT_MIN	Time step size required for triggering a remesh. This option applies only to general three-dimensional tetrahedral remeshing and is checked before mass scaling is applied and the time step size reset.
ICURV	Define number of elements along the radius in the adaptivity. See Remark 3 .
CID	Coordinate system ID for three dimensional axisymmetric remeshing. The z-axis in the defined coordinate system is the orbital axis and must be parallel to the global z-axis in the current axisymmetric remesher. EQ.0: Use global coordinate, and the global z-axis is the orbital axis (default)
SEGANG	<i>For axisymmetric 3D remeshing.</i> Angular element size in degrees. <i>For general (tet) 3D remeshing.</i> Critical angle specified in radians needed to preserve feature lines.
IVT	Internal variable transfer in adaptive EFG (see Remark 4): EQ.1: Moving least square approximation with Kronecker-delta property (recommended in general case) EQ.-1: Moving least square approximation without Kronecker-delta property EQ.2: Partition of unity approximation with Kronecker-delta property EQ.-2: Partition of unity approximation without Kronecker-delta property EQ.-3: Finite element approximation
IAT	Flag for interactive adaptivity (see Remark 2): EQ.0: No interactive adaptivity EQ.1: Interactive adaptivity combined with predefined adaptivity. EQ.2: Purely interactive adaptivity. The time interval between two successive adaptive steps is bounded by ADPFREQ.

VARIABLE	DESCRIPTION
	EQ.3: Purely interactive adaptivity
IAAT	<p>Interactive adaptivity adjustable tolerance:</p> <p>EQ.0: The tolerance to trigger interactive adaptivity is not adjusted.</p> <p>EQ.1: The tolerance is adjusted in run-time to avoid over-activation.</p>
ICONOPT	<p>Remeshing option for tied contact surfaces between two adaptive parts:</p> <p>EQ.0: No special remeshing</p> <p>EQ.-1: The two contact surfaces are matched at the first adaptive step and not remeshed in later adaptive steps.</p>
MM	<p>Adaptive remeshing with monotonic resizing for either EFG or FEM:</p> <p>EQ.1: The adaptive remeshing cannot coarsen a mesh. The current implementation only supports IAT = 1, 2, 3 for EFG. It is available for FEM tetrahedral remeshing too.</p>
IAT1	Shear strain tolerance for interactive adaptivity. If the shear strain in any formulation 42 EFG tetrahedral element exceeds IAT1, remeshing is triggered. (0.1 ~ 0.5 is recommended).
IAT2	L_{\max}/L_{\min} tolerance where L_{\max} and L_{\min} are the maximum and minimum edge lengths of any given formulation 42 EFG tetrahedral element. If this ratio in any element exceeds IAT2, remeshing is triggered. (RMAX/RMIN is recommended.)
IAT3	Volume change tolerance. If the normalized change in volume of any formulation 42 tetrahedral element, defined as $ v_1 - v_0 / v_0 $ where v_1 is the current element volume and v_0 is the element volume immediately after the most recent remeshing, exceeds IAT3, remeshing is triggered. (0.5 is recommended.)

Remarks:

1. **Edge length.** The value of RMIN and RMAX should be of the same order. The value of RMAX can be set to 2-5 times greater than RMIN. RMIN and RMAX may be superseded in specific regions of the part being adapted; see the variables PID, BRMIN, and BRMAX in *DEFINE_BOX_ADAPTIVE.

2. **Interactive adaptivity.** Interactive adaptivity is only supported for the adaptive 4-noded mesh-free (EFG) solid formulation (ELFORM = 42 in *SECTION_SOLID_EFG). When interactive adaptivity is invoked (IAT > 0), even if none of the tolerances IAT1, IAT2, and IAT3 for the three respective indicators (shear strain, edge length ratio, normalized volume change) are exceeded, remeshing will still be triggered if any of the three indicators over a single explicit time step changes by more than 50%, that is, if

$$\frac{|[\text{value}]_n - [\text{value}]_{n-1}|}{|[\text{value}]_{n-1}|} > 0.5$$

where $[\text{value}]_n$ denotes value of indicator in n^{th} (current) time step and $[\text{value}]_{n-1}$ denotes value of indicator in previous time step. This condition is checked only if $[\text{value}]_{n-1}$ is nonzero.

3. **ICURV.** ICURV represents a number of elements and applies only when ADPENE > 0 in *CONTROL_ADAPTIVE. The “desired element size” at each point on the SURFA contact surface is computed based on the tooling radius of curvature (see the description of ADPENE in *CONTROL_ADAPTIVE), so that ICURV elements would be used to resolve a hypothetical 90 degree arc at the tooling radius of curvature. The value of ICURV is (internally) limited to be ≥ 2 and ≤ 12 . The final adapted element size is adjusted as necessary to fall within the size range set forth by RMIN and RMAX.
4. **IVT.** IVT defines different remapping schemes for the adaptive 4-noded mesh-free (EFG) solid formulation (ELFORM = 42 in *SECTION_SOLID_EFG). The finite element approximation (IVT = -3) is the most efficient one in terms of computational cost. The moving least square approximation (IVT = 1 or -1) can significantly reduce the remapping error when there are many adaptive steps and the solution fields have local features, such as a high gradient. But this option has a high CPU and memory cost. The partition of unity approximation (IVT = 2 or -2) is slightly more accurate than the finite element approximation (IVT = -3).

***CONTROL_REQUIRE_REVISION**

Purpose: To prevent the model from being run in old versions of LS-DYNA. This might be desirable due to known improvements in the program, required capability, etc.

Card 1	1	2	3	4	5	6	7	8
Variable	RELEASE	SVNREV	GITREV					
Type	C	I	I					
Default	Rem 2	none	none					

VARIABLE	DESCRIPTION
RELEASE	The release of code required. This should be a string such as "R6.1.0" or "R7.0"
SVNREV	The minimum SVN revision required (ignored by executables made after our move to git). This corresponds to the "SVN Version" field in the d3hsp file for versions of LS-DYNA prior to our move to git for version control. R12.0 was the last release version made while we were using SVN for version control.
GITREV	The minimum git revision required corresponding to the release of code (ignored by executables made prior to our move to git). This corresponds to part of the git hash given in the "Revision" field in the d3hsp file for versions of LS-DYNA after our move to git for version control. In d3hsp, the last string is "R[Release #]-[number]-[alphanumeric]". If RELEASE is given, LS-DYNA compares GITREV to that [number]. If RELEASE is not given, see Remark 2 . R11.2 was the first release version made when we moved to git.

Remarks:

1. **Number of Lines.** Any number of lines can appear, indicating for example that a particular feature was introduced in different release branches at different times.
2. **RELEASE.** If the RELEASE field is left empty, then any executable whose development split from the main SVN or git trunk after the given SVNREV or GITREV, respectively, will be allowed.

Example:

```
*CONTROL_REQUIRE_REVISION
R6.1      79315
R7.0      78310
          78304
```

The above example would prevent execution by any R6.1 executable before r79315, any R7.0 before r78310, and all other executables whose development split from the main trunk before r78304. Note that no versions of R6.0, R6.0.0, or R6.1.0 are allowed: R6.1 does NOT imply R6.1.0, no matter what the revision of R6.1.0 – R6.1.0 would have to be explicitly listed. Similarly, R7.0.0 would not be allowed because it is not listed, and it split from the trunk in r76398. Any future R8.X executable would be allowed, since it will have split from the trunk after r78304.

CONTROL_RIGID**CONTROL*****CONTROL_RIGID**

Purpose: Special control options related to rigid bodies and linearized flexible bodies; see *PART_MODES.

Card 1	1	2	3	4	5	6	7	8
Variable	LMF	JNTF	ORTHMD	PARTM	SPARSE	METALF	PLOTEL	RBSMS
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

Remaining cards are optional.^t

Card 2	1	2	3	4	5	6	7	8
Variable	NORBIC	GJADSTF	GJADVSC	TJADSTF	TJADVSC	RCVLR2D		
Type	I	F	F	F	F	I		
Default	0	0.0	0.0	0.0	0.0	0		

VARIABLE	DESCRIPTION
LMF	Joint formulation flag for explicit analysis. This flag can be used to switch to an implicit formulation for joints (*CONSTRAINED_JOINT_OPTION), which uses Lagrange multipliers to impose prescribed kinematic boundary conditions and joint constraints. There is a slight cost overhead due to the assembly of sparse matrix equations, which are solved using standard procedures for nonlinear problems in rigid multi-body dynamics. EQ.0: Penalty formulation for joints (default) EQ.1: Lagrange-multiplier-based formulation for joints
JNTF	Generalized joint stiffness formulation; see Remark 1 below: EQ.0: Incremental update EQ.1: Total formulation (exact)

*CONTROL

*CONTROL_RIGID

VARIABLE	DESCRIPTION
	EQ.2: Total formulation intended for implicit analysis
ORTHMD	Orthogonalize modes with respect to each other: EQ.0: True EQ.1: False. The modes are already orthogonalized.
PARTM	Use global mass matrix to determine part mass distribution. This mass matrix may contain mass from other parts that share nodes. See Remark 2 below. EQ.0: True EQ.1: False
SPARSE	Use sparse matrix multiply subroutines for the modal stiffness and damping matrices. See Remark 3 . EQ.0: False. Do full matrix multiplies (frequently faster). EQ.1: True
METALF	Use fast update of rigid body nodes. Only applicable to sheet metal forming analysis using 2D and 3D shell elements. If this option is active, the rotational motion of all rigid bodies should be suppressed. EQ.0: Full treatment EQ.1: Fast update for metal forming applications
PLOTEL	Automatic generation of *ELEMENT_PLOTEL for *CONSTRAINED_NODAL_RIGID_BODY: EQ.0: No generation EQ.1: One part is generated for all nodal rigid bodies with the PID set to 10000000. EQ.2: One part is generated for each nodal rigid body in the problem with a part ID of 10000000 + PID, where PID is the nodal rigid body ID.
RBSMS	Flag to apply consistent treatment of rigid bodies in selective and conventional mass scaling, Remark 4 . EQ.0: Off EQ.1: On

VARIABLE	DESCRIPTION
NORBIC	Circumvent rigid body inertia check, see Remark 5 . EQ.0: Off EQ.1: On
GJADSTF	Rotational stiffness is added to all joints in the model to model a small resistance, such as joint friction, or to avoid zero-energy modes in implicit. This is equivalent to defining a *CONSTRAINED_JOINT_STIFFNESS_GENERALIZED to the free rotational degrees of freedom of all joints, using a slope in LCIDPH, LCIDT, and LCIDPS equal to GJADSTF. This field is advantageous for models with many joints because it only has to be defined once for all joints, whereas the equivalent keyword must be set for each joint.
GJADVSC	Rotational damping is added to all joints in the model to model a small resistance, such as joint friction, or to avoid zero-energy modes in implicit. This is equivalent to defining a *CONSTRAINED_JOINT_STIFFNESS_GENERALIZED to the free rotational degrees of freedom of all joints, using a slope in DLCIDPH, DLCIDT, and DLCIDPS equal to GJADVSC. Like GJADSTF, this field is advantageous for models with many joints because it only has to be defined once for all joints, whereas the equivalent keyword must be set for each joint.
TJADSTF	Translational stiffness is added to all joints in the model to model a small resistance, such as joint friction, or to avoid zero-energy modes in implicit. This is equivalent to defining a *CONSTRAINED_JOINT_STIFFNESS_TRANSLATIONAL to the free translational degrees of freedom of all joints, using a slope in LCIDX, LCIDY, and LCIDZ equal to TJADSTF. Like GJADSTF, this field is advantageous for models with many joints because it only has to be defined once for all joints, whereas the equivalent keyword must be set for each joint.
TJADVSC	Translational damping is added to all joints in the model to model a small resistance, such as joint friction, or to avoid zero-energy modes in implicit. This is equivalent to defining a *CONSTRAINED_JOINT_STIFFNESS_TRANSLATIONAL to the free translational degrees of freedom of all joints, using a slope in DLCIDX, DLCIDY, and DLCIDZ equal to TJADVSC. Like GJADSTF, this field is advantageous for models with many joints because it only has to be defined once for all joints, whereas the equivalent

VARIABLE	DESCRIPTION
	keyword must be set for each joint.
RCVLR2D	Recover the lead rigid body of constrained rigid bodies, which was changed due to *DEFORMABLE_TO_RIGID_AUTOMATIC (see Remark 6): EQ.0: Off EQ.1: On

Remarks:

1. **JNTF.** The default behavior is for the relative angles between the two coordinate systems to be done incrementally. This is an approximation, in contrast to the total formulation where the angular offsets are computed exactly. The disadvantage of the latter approach is that a singularity exists when an offset angle equals 180 degrees. In most applications, the stop angles exclude this possibility and JNTF=1 should not cause a problem. JNTF=2 is implemented with smooth response and especially intended for implicit analysis.
2. **PARTM.** If the determination of the normal modes included the mass from both connected bodies and discrete masses, or if there are no connected bodies, then the default is preferred. When the mass of a given part ID is computed, the resulting mass vector includes the mass of all rigid bodies that are merged to the given part ID, but does not include discrete masses. See the keyword: *CONSTRAINED_RIGID_BODIES. A lumped mass matrix is always assumed.
3. **SPARSE.** Sparse matrix multipliers save a substantial number of operations if the matrix is truly sparse. However, the overhead will slow the multipliers for densely populated matrices.
4. **RBSMS.** In selective mass scaling, rigid bodies connected to deformable elements can significantly add inertia due to missing terms in the SMS mass matrix. This problem has been observed in automotive applications where spot welds are modeled using constrained nodal rigid bodies. Applying consistent rigid body treatment can significantly improve accuracy and robustness at the expense of increased CPU intensity. One aspect of this treatment is that all different types of rigid bodies are treated equally. For example, like for *CONSTRAINED_NODAL_RIGID_BODIES, this treatment merges parts with *MAT_RIGID that share nodes into one rigid body.

RBSMS = 1 also applies the same rigid body treatment to conventional mass scaling because inconsistencies, for various reasons, may result in unstable solution schemes, even in this case.

5. **NORBIC.** During initialization, the determinant of the rigid body inertia tensor is checked. If it falls below a tolerance value of 10^{-30} , LS-DYNA issues an error message and the calculation stops. In some rare cases (for example with an unfavorable system of units), such tiny values would still be valid. In this case, NORBIC should be set to 1 to circumvent the implied inertia check.
6. **RCVLR2D.** When a deformable part PID is switched to rigid and becomes constrained to lead rigid part LRB, the rigid part PIDL sharing common nodes with PID will be constrained to LRB as well. At the same time, the rigid part PIDC having PIDL as the lead rigid body through *CONSTRAINED_RIGID_BODIES will now have LRB as its new lead rigid body (see *DEFORMABLE_TO_RIGID_AUTOMATIC and *CONSTRAINED_RIGID_BODIES). When PID is switched back to deformable, PIDC will be constrained to its old lead PIDL if RCVLR2D = 1. Otherwise, PIDC will stay constrained LRB.

*CONTROL

*CONTROL_SEGMENTS_IN_ALE_COUPLING

*CONTROL_SEGMENTS_IN_ALE_COUPLING

Purpose: Deactivate segments in the ALE penalty coupling (see CTYPE = 4, 5 and 6 in *CONSTRAINED_LAGRANGE_IN_SOLID) if the segments are face to face with other segments. By default, all the segments (element faces with 4 or 3 nodes in 3D or element sides with 2 nodes in 2D) search for the ALE material interface involved in the coupling. If the interface is close to a segment, penalty coupling forces are applied on both to keep them together. Some segments might not need to be involved in the coupling if they face another segment. For example, let 2 side-by-side cubic structures have element faces in contact. As long as these segments face each other, they do not need to apply coupling forces on the material interface. In this particular example, if inside segments near the corners were to drag the material interfaces to them, leakages could occur; see [Figure 12-85](#).

See also *CONSTRAINED_LAGRANGE_IN_SOLID (CTYPE = 4, 5 and 6), *SET_SEGMENT, and *ALE_MULTI-MATERIAL_GROUP.

Segment Set Card. This card is optional.

Card 1	1	2	3	4	5	6	7	8
Variable	RANKEY	SEGSET	NCYCHK	SYM				
Type	I	I	I	I				
Default	0	0	10	0				

Coupling Card. This card is optional card.

Card 2	1	2	3	4	5	6	7	8
Variable	NINTHK	CONTHK						
Type	I	F						
Default	0	0.0						

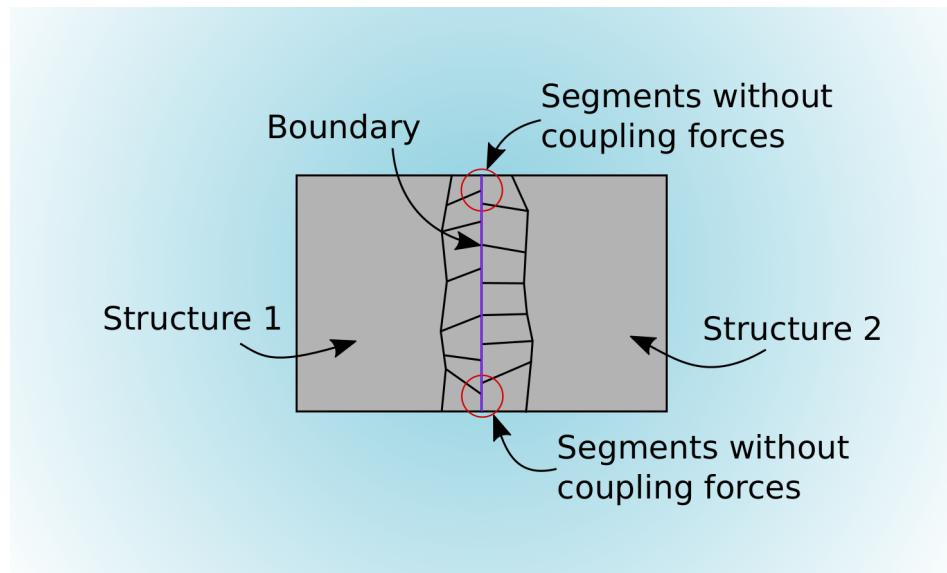


Figure 12-85. An example of when penalty coupling should be deactivated for certain segments.

VARIABLE	DESCRIPTION
RANKEY	Instantiation of *CONSTRAINED_LAGRANGE_IN_SOLID in the input deck controlled by this keyword (see Remarks 1 and 2). For example, if RANKEY = 2, then the second *CONSTRAINED_LAGRANGE_IN_SOLID definition is controlled by this keyword.
SEGSET	Set ID of *SET_SEGMENT (see Remark 2)
NCYCHK	Number of cycles between checks to activate/deactivate coupling segments (see Remark 3)
SYM	Flag to deactivate coupling segments with normal boundary constraints. EQ.0: Off EQ.1: On
NINTHK	Minimum number of coupling points in contact to deactivate the segment (see Remark 4).
CONTHK	Contact thickness (see Remark 5)

Remarks:

- Keyword without Data Cards.** If *CONTROL_SEGMENTS_IN_ALE_COUPLING is included without any data cards, namely, Cards 1 and 2, all the *CON-

STRAINED_LAGRANGE_IN_SOLID keywords are controlled by *CONTROL_SEGMENTS_IN_ALE_COUPLING.

2. **Segment Sets.** The RANKEYth *CONSTRAINED_LAGRANGE_IN_SOLID keyword definition in the keyword deck determines the segment set controlled by *CONTROL_SEGMENTS_IN_ALE_COUPLING. The parameter LSTRSID with LSTRSTYP = 2 in *CONSTRAINED_LAGRANGE_IN_SOLID refers to a segment set. If LSTRSTYP < 2, LSTRSID refers to a part or part set. However, the segments of these parts are still generated during the initialization. If RANKEY and SEGSET are both nonzero, the segments common to RANKEY and SEGSET will be controlled. If RANKEY and SEGSET do not have segments in common, their segment lists will both be controlled.
3. **Check Frequency.** The parameter NBKT in *CONTROL_ALE controls how often the bucket sorting to search segments in contact is updated.
4. **Coupling Points.** NQUAD in *CONSTRAINED_LAGRANGE_IN_SOLID defines the number of coupling points by segment (NQUAD × NQUAD in 3D and NQUAD in 2D). If NINTHK = 0, all the coupling points of a segment should be in contact to deactivate it.
5. **Contact Thickness.** For this keyword, a segment, SG1, is in contact with another segment, SG2, if at least one coupling point of SG1 is inside a cylinder formed by SG2's area as its base and SG2's normal as its generatrix (see *SET_SEGMENT for the normal definition). The cylinder height is CONTHK. If CONTHK = 0 and the segment is attached to a shell (or beam in 2D), CONTHK by default is half the element thickness. If CONTHK = 0 and the segment is attached to a solid (or shell in 2D), CONTHK by default is a thousandth of the largest segment diagonal.

CONTROL_SHELL**CONTROL*****CONTROL_SHELL**

Purpose: Provide controls for computing shell response.

Card Summary:

Card 1. This card is required.

WRPANG	ESORT	IRNXX	ISTUPD	THEORY	BWC	MITER	PROJ
--------	-------	-------	--------	--------	-----	-------	------

Card 2. This card and all the following cards are optional.

ROTAACL	INTGRD	LAMSHT	CSTYP6	THSHEL			
---------	--------	--------	--------	--------	--	--	--

Card 3. This card is optional.

PSTUPD	SIDT4TU	CNTCO	ITSFLG	IRQUAD	W-MODE	STRETCH	ICRQ
--------	---------	-------	--------	--------	--------	---------	------

Card 4. This card is optional.

NFAIL1	NFAIL4	PSNFAIL	KEEPCS	DELFRE	DRCPSID	DRCPRM	INTPERR
--------	--------	---------	--------	--------	---------	--------	---------

Card 5. This card is optional.

DRCMTH	LISPSID	NLOCDT	ISWSHL				
--------	---------	--------	--------	--	--	--	--

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	WRPANG	ESORT	IRNXX	ISTUPD	THEORY	BWC	MITER	PROJ
Type	F	I	I	I	I	I	I	I
Default	20.	0	-1	0	2	2	1	0

VARIABLE	DESCRIPTION
WRPANG	Shell element warpage angle in degrees. If a warpage greater than this angle is found, a warning message is printed.
ESORT	Sorting of triangular shell elements to automatically switch degenerate quadrilateral shell formulations to more suitable triangular

*CONTROL

*CONTROL_SHELL

VARIABLE	DESCRIPTION
	<p>shell formulations.</p> <p>EQ.0: Do not sort (default).</p> <p>EQ.1: Sort (switch to C0 triangular shell formulation 4, or if a quadratic shell, switch to shell formulation 24, or if a shell formulation with thickness stretch, switch to shell formulation 27).</p> <p>EQ.2: Sort (switch to DKT triangular shell formulation 17, or if the shell is quadratic, switch to shell formulation 24). The DKT formulation will be unstable for an uncommonly thick, triangular shell.</p>
IRNXX	<p>Shell normal update option. This option applies to the Hughes-Liu formulations (ELFORMs 1, 6, 7, and 11), the BWC formulation (ELFORM 10), and if warping stiffness is turned on, the Belytschko-Tsay formulations (ELFORMs 2 and 25 with BWC = 1 and ELFORMs -16, 16, and 26 with hourglass type 8).</p> <p>EQ.-2: Unique nodal fibers that are incrementally updated based on the nodal rotation at the location of the fiber (see Remark 1)</p> <p>EQ.-1: Recomputed fiber directions each cycle (see Remark 1)</p> <p>EQ.0: Default set to -1</p> <p>EQ.1: Compute on restarts</p> <p>EQ.n: Compute every <i>n</i> cycles (Hughes-Liu shells only)</p>
ISTUPD	<p>Shell thickness change option for deformable shells. For crash analysis, neglecting the elastic component of the strains, ISTUPD = 4, may improve energy conservation and stability. The option specified with ISTUPD applies to all shell parts unless PSTUPD (and optionally, SIDT4TU) on Card 3 is specified. See the description of those variables below.</p> <p>EQ.0: No thickness change (default)</p> <p>EQ.1: Membrane straining causes thickness change in 3 and 4 node shell elements. This option is important in sheet metal forming or whenever membrane stretching is important.</p> <p>EQ.2: Membrane straining causes thickness change in 8 node thick shell elements, types 1 and 2. We do not recommend this option for implicit or explicit solutions which use the</p>

VARIABLE	DESCRIPTION
	fully integrated type 2 elements. Types 3 and 5 thick shells are continuum-based, and thickness changes are always considered.
EQ.3:	Options 1 and 2 apply
EQ.4:	Option 1 applies, but the elastic strains are neglected for the thickness update. This option only applies to shells (not thick shells) and the most common elastic-plastic materials for which the elastic response is isotropic. See SIDT4TU for selective application of this option.
EQ.5:	Same as 1, but thickness changes are stored with double precision in single-precision binaries for shell elements with ELFORM = 2, 4, or 16. All other shell element types are stored with single precision in single-precision binaries. This change to storage for these element types should make results more comparable (single precision versus double precision) in long-lasting explicit analyses.
EQ.6:	Same as 4, but thickness changes are stored with double precision in single-precision binaries for shell elements with ELFORM = 2, 4, or 16. All other shell element types are stored with single precision in single-precision binaries. This change to storage for these element types should make results more comparable (single precision versus double precision) in long-lasting explicit analyses.
THEORY	Default shell formulation. For a complete list of shell formulations, refer to *SECTION_SHELL . For remarks on overriding this default and how THEORY may affect contact behavior, see Remark 3 .
EQ.1:	Hughes-Liu
EQ.2:	Belytschko-Tsay (default)
EQ.3:	BCIZ triangular shell (not recommended)
EQ.4:	C0 triangular shell
EQ.5:	Belytschko-Tsay membrane
EQ.6:	S/R Hughes Liu
EQ.7:	S/R co-rotational Hughes Liu
EQ.8:	Belytschko-Leviathan shell
EQ.9:	Fully integrated Belytschko-Tsay membrane
EQ.10:	Belytschko-Wong-Chiang

*CONTROL

*CONTROL_SHELL

VARIABLE	DESCRIPTION
	EQ.11: Fast (co-rotational) Hughes-Liu EQ.12: Plane stress (xy -plane) EQ.13: Plane strain (xy -plane) EQ.14: Axisymmetric solid (y -axis of symmetry) – area weighted. See Remark 8 EQ.15: Axisymmetric solid (y -axis of symmetry) – volume weighted. See Remark 8 EQ.16: Fully integrated shell element (very fast) EQ.17: Discrete Kirchhoff triangular shell (DKT) EQ.18: Discrete Kirchhoff linear shell either quadrilateral or Triangular with 6DOF per node EQ.20: C0 linear shell element with 6 DOF per node EQ.21: C0 linear shell element with 5 DOF per node with the Pian-Sumihara membrane hybrid quadrilateral membrane EQ.25: Belytschko-Tsay shell with thickness stretch EQ.26: Fully integrated shell with thickness stretch EQ.27: C0 triangular shell with thickness stretch
BWC	Warping stiffness for Belytschko-Tsay shells: EQ.1: Belytschko-Wong-Chiang warping stiffness added EQ.2: Belytschko-Tsay (default)
MITER	Plane stress plasticity option (applies to materials 3, 18, 19, and 24): EQ.1: Iterative plasticity with 3 secant iterations (default) EQ.2: Full iterative plasticity EQ.3: Radial return noniterative plasticity. May lead to false results and must be used with great care.
PROJ	Projection method for the warping stiffness in the Belytschko-Tsay shell (BWC above) and the Belytschko-Wong-Chiang elements (see Remark 2 below). This parameter only applies to explicit calculations since the implicit solver always uses the full projection method. EQ.0: Drill projection

VARIABLE		DESCRIPTION						
Card 2	1	2	3	4	5	6	7	8
Variable	ROTASCL	INTGRD	LAMSHT	CSTYP6	THSHEL			
Type	F	I	I	I	I			
Default	1.	0	0	1	0			

VARIABLE		DESCRIPTION						
ROTASCL	Define a scale factor for the rotary shell mass. This option is not for general use. The rotary inertia for shells is automatically scaled to permit a larger time step size. A scale factor other than the default, that is, unity, is not recommended.							
INTGRD	Default through thickness numerical integration rule for shells and thick shells. If more than 10 integration points are requested, a trapezoidal rule is used unless a user-defined rule is specified. EQ.0: Gauss integration. If 1 - 10 integration points are specified, the default rule is Gauss integration. EQ.1: Lobatto integration. If 3 - 10 integration points are specified, the default rule is Lobatto. For two-point integration, the Lobatto rule is very inaccurate, so Gauss integration is used instead. Lobatto integration has an advantage in that the inner and outer integration points are on the shell surfaces.							
LAMSHT	Laminated shell theory flag. Except for those using the Green-Lagrange strain tensor, laminated shell theory is available for all thin shell and thick shell materials (this feature is developed and implemented by Professor Ala Tabiei). It is activated when LAMSHT = 3, 4, or 5 and by using *PART_COMPOSITE or *INTEGRATION-SHELL to define the integration rule. See Remark 9 . EQ.0: Do not update shear corrections. EQ.1: Activate laminated shell theory (deprecated). As of R16, the keyword reader changes LAMSHT = 1 to LAMSHT = 3.							

*CONTROL

*CONTROL_SHELL

<u>VARIABLE</u>	<u>DESCRIPTION</u>
	<p>EQ.3: Activate laminated thin shells.</p> <p>EQ.4: Activate laminated shell theory for thick shells.</p> <p>EQ.5: Activate laminated shell theory for thin and thick shells.</p>
CSTYP6	<p>Coordinate system for the type 6 shell element. The default system computes a unique local system at each in-plane point. The uniform local system computes just one system used throughout the shell element. This involves fewer calculations and is therefore more efficient. The change of systems has a slight effect on results; therefore, the older, less efficient method is the default.</p> <p>EQ.1: Variable local coordinate system (default)</p> <p>EQ.2: Uniform local system</p>
THSHEL	<p>Thermal shell option (applies only to thermal and coupled structural thermal analyses). See parameter THERM on *DATABASE_EXTENT_BINARY keyword.</p> <p>EQ.0: No temperature gradient is considered through the shell thickness (default).</p> <p>EQ.1: A temperature gradient is calculated through the shell thickness.</p>

Card 3	1	2	3	4	5	6	7	8
Variable	PSTUPD	SIDT4TU	CNTCO	ITSFLG	IRQUAD	W-MODE	STRETCH	ICRQ
Type	I	I	I	I	I	F	F	I
Default	0	0	0	0	0	inactive	inactive	0

<u>VARIABLE</u>	<u>DESCRIPTION</u>
PSTUPD	<p> PSTUPD is the optional shell part set ID specifying which part IDs have or do not have their thickness updated according to ISTUPD, subject to further specification as given by SIDT4TU below. The shell thickness update option as specified by ISTUPD by default applies to all shell elements in the mesh.</p> <p>LT.0: Parts in shell part set PSTUPD are excluded from the</p>

CONTROL_SHELL**CONTROL**

VARIABLE	DESCRIPTION
	shell thickness update. EQ.0: All deformable shells use the shell thickness update as specified by ISTUPD. GT.0: Only parts in shell part set PSTUPD undergo shell thickness update.
SIDT4TU	Shell part set ID for parts that use the type 4 thickness update where elastic strains are ignored. The shell parts in part set SIDT4-TU must also be included in the part set defined by PSTUPD. SIDT4TU has no effect unless ISTUPD is set to 1 or 3. Shell parts in the shell part set PSTUPD that are not also in the shell part set SIDT4TU use the type 1 thickness update.
CNTCO	Flag affecting the location of contact surfaces for shells when NLOC is nonzero in *SECTION_SHELL or in *PART_COMPOSITE , or when OFFSET is specified using *ELEMENT_SHELL_OFFSET . CNTCO is not supported for the tracked side of NODES_TO_SURFACE type contacts, nor does it have any effect on Mortar contacts. For Mortar contacts NLOC or OFFSET completely determines the location of the contact surfaces, as if CNTCO = 1. EQ.0: NLOC and OFFSET have no effect on the location of the shell contact surfaces. EQ.1: Contact reference plane (see Remark 4) coincides with the shell reference surface. EQ.2: Contact reference plane (see Remark 4) is affected by contact thickness. This is typically not physical. EQ.3: Similar to 1 but with improved behavior at corners or folds in the mesh in certain cases. See Remarks 4 and 5 . EQ.4: Similar to 2 but with improved behavior at corners or folds in the mesh in certain cases. See Remarks 4 and 5 .
ITSFLG	Flag to activate/deactivate initial transverse shear stresses (local shell stress components σ_{yz} and σ_{zx}) from *INITIAL_STRESS-SHELL : EQ.0: Keep transverse shear stresses EQ.1: Set transverse shear stresses to zero
IRQUAD	In-plane integration rule for the 8-node quadratic shell element (shell formulation 23):

*CONTROL

*CONTROL_SHELL

<u>VARIABLE</u>	<u>DESCRIPTION</u>
	EQ.2: 2×2 Gauss quadrature EQ.3: 3×3 Gauss quadrature (default)
W-MODE	W-Mode amplitude for element deletion, specified in degrees. See Figure 12-86 and Remark 7 for the definition of the angle.
STRETCH	Stretch ratio of element diagonals for quadrilateral elements or of side edges for triangular elements that results in element deletion. This option is activated only if either NFAIL1 or NFAIL4 are non-zero and STRETCH > 0.0.
ICRQ	Continuous treatment across element edges for some specified result quantities. See Remark 10 . EQ.0: Not active EQ.1: Thickness and plastic strain EQ.2: Thickness

Card 4	1	2	3	4	5	6	7	8
Variable	NFAIL1	NFAIL4	PSNFAIL	KEEPCS	DELFRE	DRCPSID	DRCPRM	INTPERR
Type	I	I	I	I	I	I	F	
Default	inactive	inactive	0	0	0	0	1.0	

<u>VARIABLE</u>	<u>DESCRIPTION</u>
NFAIL1	Flag to check for highly distorted under-integrated shell elements, print a message, and delete the element or terminate. Generally, this flag is not needed for one point elements that do not use warping stiffness. A distorted element is one where a negative Jacobian exist within the domain of the shell, not just at integration points. The checks are made away from the CPU requirements for one point elements. If nonzero, NFAIL1 can be changed in a restart. EQ.1: Print message and delete element. EQ.2: Print message, write d3dump file, and terminate. GT.2: Print message and delete element. When NFAIL1 elements are deleted, then write d3dump file and terminate.

VARIABLE	DESCRIPTION
	These NFAIL1 failed elements also include all shell elements that failed for other reasons than distortion. Before the d3dump file is written, NFAIL1 is doubled, so the run can immediately be continued if desired.
NFAIL4	Flag to check for highly distorted fully-integrated shell elements, print a message, and delete the element or terminate. Generally, this flag is recommended. A distorted element is one where a negative Jacobian exist within the domain of the shell, not just at integration points. The checks are made away from the integration points to enable the bad elements to be deleted before an instability leading to an error termination occurs. If nonzero, NFAIL4 can be changed in a restart. EQ.1: Print message and delete element. EQ.2: Print message, write d3dump file, and terminate. GT.2: Print message and delete element. When NFAIL4 elements are deleted, then write d3dump file and terminate. These NFAIL4 failed elements also include all shell elements that failed for other reasons than distortion. Before the d3dump file is written, NFAIL4 is doubled, so the run can immediately be continued if desired.
PSNFAIL	Optional shell part set ID specifying which part IDs are checked by the NFAIL1, NFAIL4, and W-MODE options. If zero, all shell part IDs are included.
KEEPCS	Flag to keep the contact segments of failed shell elements in the calculation. The contact segments of the failed shells remain active until a node shared by the segments has no active shells attached. Only then are the segments deleted. EQ.0: Inactive EQ.1: Active
DELFRE	Flag to delete shell elements whose neighboring shell elements have failed; consequently, the shell is detached from the structure and moving freely in space. This condition is checked if NFAIL1 or NFAIL4 are nonzero. EQ.0: Inactive EQ.1: Isolated elements are deleted. EQ.2: Elements that are isolated and triangular elements that are

*CONTROL

*CONTROL_SHELL

VARIABLE	DESCRIPTION							
	connected by only one node are deleted.							
	EQ.3: Elements that are either isolated or connected by only one node are deleted.							
DRCPSID	Part set ID for drilling rotation constraint method (see Remark 6 below).							
DRCPRM	Drilling rotation constraint parameter (default = 1.0). GT.0: Constant scaling value. LT.0: DRCPRM is a *DEFINE_FUNCTION ID (FID), see Remark 6 below.							
INTPERR	Flag for behavior in case of unwanted interpolation/extrapolation of initial stresses from *INITIAL_STRESS_SHELL. EQ.0: Only warning is written, calculation continues (default). EQ.1: Error exit, calculation stops.							

Card 5	1	2	3	4	5	6	7	8
Variable	DRCMTH	LISPSID	NLOCDT	ISWSHL				
Type	I	I	I	I				
Default	0	0	0	0				

VARIABLE	DESCRIPTION
DRCMTH	Drilling rotation constraint method. Options to choose how drilling kinematics are determined. EQ.0: Generalized drilling strain rate at shell element nodes involving drill rotation at the specific node plus the translational velocities of two adjacent nodes. See more details in Erhart and Borrvall [2013]. EQ.1: Direct use of the spin tensor (for example, see Stress Update Overview in the LS-DYNA Theory Manual) with respect to the shell element's normal direction, numerically integrated at element level. A similar approach is

VARIABLE	DESCRIPTION
	described in Kanok-Nukulchai [1979].
LISPSID	Part set ID related to *INITIAL_STRESS_SHELL . For all parts in this set, the initial stress components SIGXX, SIGYY, ..., SIGZX are defined in the local (element) coordinate system.
NLOCDT	Flag for time step handling for shell elements with offset. If the shell reference surface is offset by NLOC (*SECTION_SHELL) or OFFSET (*ELEMENT_SHELL), the time step size of those shell elements is reduced to fix instabilities. The reduction of the time step size is based on numerical tests which show a dependence on the offset distance and the ratio of shell thickness to edge length (T/L). <p>EQ.0: Reduce time step size up to 10% to avoid instabilities. Care must be taken since a smaller time step will lead to larger masses due to mass scaling.</p> <p>EQ.1: No reduction of time step to restore prior behavior if necessary. Instabilities were most likely observed for aspect ratios of $T/L > 0.5$.</p>
ISWSHL	Flag for switching between formulations 16 and 30: EQ.0: Do not convert the shell formulations. EQ.1: Convert all formulation 16 shells to 30. EQ.2: Convert all formulation 30 shells to 16.

Remarks:

- IRNXX for curved geometries.** We recommend IRNXX = -1 over IRNXX = -2 for curved geometries. IRNXX = -2 initializes the fibers to be the nodal average normals. Thus, in curved geometries, it can lead to non-objective results. In contrast, IRNXX = -1 sets the fibers to the elementwise constant normals, leading to more objective results.
- Drill versus full projections for warping stiffness.** The drill projection is used in the addition of warping stiffness to the Belytschko-Tsay and the Belytschko-Wong-Chiang shell elements. This projection generally works well and is very efficient, but to quote Belytschko and Leviathan:

"The shortcoming of the drill projection is that even elements that are invariant to rigid body rotation will strain under rigid body rotation if the drill projection is applied. On one hand, the excessive flexibility rendered by the 1-point quadrature shell element is corrected by the drill projection, but on

the other hand, the element becomes too stiff due to loss of the rigid body rotation invariance under the same drill projection."

They later went on to add in the conclusions:

"The projection of only the drill rotations is very efficient and hardly increases the computation time, so it is recommended for most cases. However, it should be noted that the drill projection can result in a loss of invariance to rigid body motion when the elements are highly warped. For moderately warped configurations the drill projection appears quite accurate."

In crashworthiness and impact analysis, elements that have little or no warpage in the reference configuration can become highly warped in the deformed configuration and may affect rigid body rotations if the drill projection is used, that is, DO NOT USE THE DRILL PROJECTION. Of course, it is difficult to define what is meant by "moderately warped." The full projection circumvents these problems but at a significant cost. The cost increase of the drill projection versus no projection as reported by Belytschko and Leviathan is 12 percent and by timings in LS-DYNA, 7 percent, but for the full projection they report a 110 percent increase, and in LS-DYNA an increase closer to 50 percent is observed.

In Version 940 of LS-DYNA, the drill projection was used exclusively, but in one problem the lack of invariance was observed; consequently, the drill projection was replaced in the Belytschko-Leviathan shell with the full projection and the full projection is now optional for the warping stiffness in the Belytschko-Tsay and Belytschko-Wong-Chiang elements. Starting with version 950 the Belytschko-Leviathan shell, which now uses the full projection, is somewhat slower than in previous versions. In general, in light of these problems, the drill projection cannot be recommended. For implicit calculations, the full projection method is used in the development of the stiffness matrix.

3. **THEORY, ELFORM, and contact with tapered shells.** All shell parts need not share the same element formulation. A nonzero value of ELFORM, given either in [*SECTION_SHELL](#) or [*PART_COMPOSITE](#), overrides the element formulation specified by THEORY in [*CONTROL_SHELL](#).

When using MPP, THEORY = 1 in [*CONTROL_SHELL](#) has a special meaning when dealing with non-uniform-thickness shells: it serves to set the nodal contact thickness equal to the average of the nodal thicknesses from the shells sharing that node. Thus, when non-uniform-thickness shells comprise a contact surface, we recommend THEORY = 1 and setting the actual shell theory using ELFORM in [*SECTION_SHELL](#). (This remark concerning THEORY does not apply to SOFT = 2 contact wherein each contact segment is considered to be of uniform thickness.)

4. **Contact surfaces.** For automatic contact types, the shell contact surfaces are always, regardless of CNTCO, offset from a contact reference plane by half a contact thickness. Contact thickness is taken as the shell thickness by default but can be overridden, for example, with input on [Card 3](#) of ***CONTACT**.

The parameter CNTCO affects how the location of the contact reference plane is determined. When CNTCO = 0, the contact reference plane coincides with the plane of the shell nodes. Whereas when CNTCO = 1 or 3, the contact reference plane coincides with the shell reference surface as determined by NLOC or by OFFSET. For CNTCO = 2 or 4, the contact reference plane is offset from the plane of the nodes by

$$-\frac{\text{NLOC}}{2} \times \text{contact thickness}$$

or by

$$\text{OFFSET} \times \left(\frac{\text{contact thickness}}{\text{shell thickness}} \right)$$

whichever applies.

5. **Improvements with CNTCO = 3 or 4.** CNTCO = 3 and 4 are similar to 1 and 2, respectively, but they include improvements in determining the contact reference plane at edges and corners in the mesh (including internal) where adjacent segments are not coplanar and contact thicknesses or areas of adjacent segments differ.
6. **Drilling rotation constraint method.** The drilling rotation constraint method, which is used by default in implicit calculations (see parameter DRCM = 4 on [*CONTROL_IMPLICIT_SOLVER](#)), can be used in explicit calculations as well by defining an appropriate part set DRCPSID. This might be helpful in situations where single constraints (such as spot welds) are connected to flat shell element topologies. The additional drill force can be scaled by DRCPRM (default value is 1.0), where a moderate value should be chosen to avoid excessive stiffening of the structure. A speed penalty of 15% at maximum may be observed with this option.

As an alternative to a constant drill force scaling parameter (DRCPRM > 0), a general function can be defined (DRCPRM < 0) with [*DEFINE_FUNCTION](#) ID |DRCPRM|. The arguments of that function include elastic shear modulus *gmod*, the shear correction factor *shrf* (from [*SECTION_SHELL](#)), the mass density *rho*, the shell element area *sarea*, and the element thickness *thk* in the following order:

```
*DEFINE_FUNCTION
<FID>
func (gmod, shrf, rho, sarea, thk) =...
```

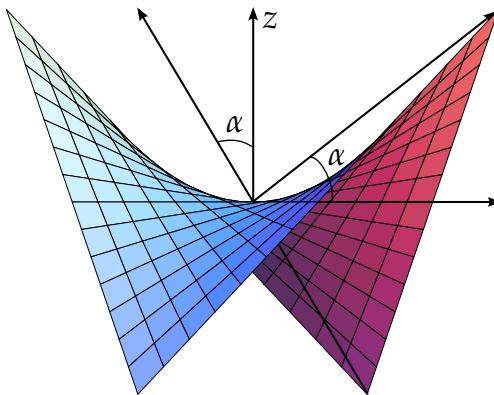


Figure 12-86. Illustration of an element in a *W*-Mode. One pair of opposite corners go up, and the other pair goes down. The angle, α , is formed by the plane of the flat element and by the vector connecting the center to the corner. See Remark 7.

The function should have the units of a force.

7. **W-mode failure criterion.** The w-mode failure criterion depends on the magnitude of the w-mode, w , compared to the approximate side length ℓ . The magnitude, w , is defined as

$$w = \frac{1}{4} [(\mathbf{x}_1 - \mathbf{x}_2) + (\mathbf{x}_3 - \mathbf{x}_4)] \cdot \mathbf{n},$$

where \mathbf{x}_i is the position vector for node i , and \mathbf{n} is the element normal vector evaluated at the centroid. The element normal is the unit vector obtained from the cross product of the diagonal vectors \mathbf{a} and \mathbf{b} as,

$$\begin{aligned}\mathbf{a} &= \mathbf{x}_3 - \mathbf{x}_1 \\ \mathbf{b} &= \mathbf{x}_4 - \mathbf{x}_2 \\ \mathbf{n} &= \frac{\mathbf{a} \times \mathbf{b}}{\|\mathbf{a} \times \mathbf{b}\|}.\end{aligned}$$

The failure criterion depends on the ratio of w to ℓ , where ℓ is defined as

$$\ell = \frac{1}{2} \left[\sqrt{2} \underbrace{\sqrt{\frac{1}{2} \|\mathbf{a} \times \mathbf{b}\|}}_{\sim \sqrt{\text{area}}} \right] \underbrace{\sim \text{diagonal length}}$$

such that the element is deleted when

$$\frac{|w|}{\ell} \geq \tan(\text{WMODE}).$$

The angle α in the figure may be identified as,

$$\alpha = \arctan \left(\frac{|w|}{\ell} \right).$$

8. **2D axisymmetric solid elements.** The 2D axisymmetric solid elements come in two types: area-weighted (type 14) and volume-weighted (type 15).
 - a) High explosive applications work best with the area-weighted approach and structural applications work best with the volume-weighted approach. The volume-weighted approach can lead to problems along the axis of symmetry under very large deformations. Often the symmetry condition is not obeyed, and the elements will kink along the axis.
 - b) The volume-weighted approach must be used if 2D shell elements are used in the mesh. Type 14 and 15 elements cannot be mixed in the same calculation.
9. **Lamination theory.** Lamination theory should be activated when the assumption that shear strain through the shell is uniform and constant becomes violated. Unless this correction is applied, the stiffness of the shell can be grossly incorrect if there are drastic differences in the elastic constants from ply to ply, especially for sandwich-type shells. Generally, without this correction, the results are too stiff. For the discrete Kirchhoff shell elements, which do not consider transverse shear, this option is ignored.
10. **Continuous result quantities.** A nodal averaging technique is used to achieve continuity for some quantities across element edges. Applying this approach to the thickness field and plastic strains (ICRQ = 1) or only thickness (ICRQ=2) can reduce alternating weak localizations sometimes observed in metal forming applications when shell elements get stretch-bended over small radii. This option currently works with shell element types 2, 4, and 16. A maximum number of 9 through-thickness integration points is allowed for this method. A maximum speed penalty of 15% may be observed with this option.

*CONTROL

*CONTROL_SOLID

*CONTROL_SOLID

Purpose: Provides controls for solid element response.

Card 1	1	2	3	4	5	6	7	8
Variable	ESORT	FMATRX	NIPTETS	SWLOCL	PSFAIL	T10JTOL	ICOH	TET13K
Type	I	I	I	I	I	F	I	I
Default	0	0	4	1	0	0.0	0	0

This card is optional.

Card 2	1	2	3	4	5	6	7	8	9	10
Variable	PM1	PM2	PM3	PM4	PM5	PM6	PM7	PM8	PM9	PM10
Type	I	I	I	I	I	I	I	I	I	I
Default	none									

This card is optional.

Card 3	1	2	3	4	5	6	7	8
Variable	TET13V	RINRT	COHEQC					
Type	I	I	I					
Default	0	0	0					

VARIABLE

DESCRIPTION

ESORT

Automatic sorting of tetrahedral and pentahedral elements to avoid using degenerate formulations for these shapes. See [*SECTION_SOLID](#).

EQ.0: No sorting (default)

EQ.1: Sort tetrahedron to type 10; pentahedron to type 15;

VARIABLE	DESCRIPTION
	cohesive pentahedron types 19 and 20 to types 21 and 22, respectively.
	EQ.2: Sort tetrahedron to type 10; 1-point integrated pentahedron to type 115; fully integrated pentahedron to type 15; cohesive pentahedron types 19 and 20 to types 21 and 22, respectively.
	EQ.3: Same as 1 but also print switched elements in message file
	EQ.4: Same as 2 but also print switched elements in message file
	EQ.11: Same as 1 except sort tetrahedron to type 13
	EQ.12: Same as 2 except sort tetrahedron to type 13
	EQ.13: Same as 3 except sort tetrahedron to type 13
	EQ.14: Same as 4 except sort tetrahedron to type 13
FMATRX	Default method used in the calculation of the deformation gradient matrix. EQ.1: Update incrementally in time. This is the default for explicit. EQ.2: Directly compute F. This is the default for implicit and implicit/explicit switching.
NIPTETS	Number of integration points used in the quadratic tetrahedron elements. Either 4 or 5 can be specified. This option applies to the types 4, 16, and 17 tetrahedron elements.
SWLOCL	Output option for stresses in solid elements used as spot welds with material *MAT_SPOTWELD. Affects elout, d3plot, d3part, etc. EQ.1: Stresses in the global coordinate system (default) EQ.2: Stresses in the element coordinate system
PSFAIL	Solid element erosion from negative volume is limited <i>only</i> to solid elements in the part set indicated by PSFAIL. This is similar to setting ERODE = 1 in *CONTROL_TIMESTEP, except that it is not global. In other words, when PSFAIL is nonzero, the time-step-based criterion for erosion (TSMIN) applies to all solid elements (except formulations 11 and 12), and the negative volume criterion

VARIABLE	DESCRIPTION
	for erosion applies only to solids in part set PSFAIL.
T10JTOL	Tolerance for Jacobian in 4-point 10-noded quadratic tetrahedra (type 16). If the quotient between the minimum and maximum Jacobian values falls below this tolerance, a warning message is issued in the messag file. This is useful for tracking (poorly shaped) elements that deteriorate convergence during implicit analysis; a value of 1.0 indicates a perfectly shaped element.
ICOH	Flag for cohesive elements to control deletion, the time step estimate, and the element type used in implicit and explicit. Breaking LS-DYNA convention, ICOH is interpreted digit-wise, namely as, $\text{ICOH} = [\text{MLK}] = K + 10 \times L + 100 \times M .$ The first digit (in the one's place), K , is interpreted as follows: <ul style="list-style-type: none"> K.EQ.0: No cohesive element deletion due to neighbor failure. K.EQ.1: Solid elements having ELF<small>ORM</small> = 19 – 22 (or ELF<small>ORM</small> = 1, 2, 15 being used with *MAT_169) are eroded when neighboring shell or solid elements fail. This works for nodewise connected parts and tied contacts. The second digit (in the ten's place), L , is interpreted as stated below. L defaults to zero if ICOH is less than 10 (having a single digit). See Remark 1 . <ul style="list-style-type: none"> L.EQ.0: Default stable time step estimate, computed from the stiffness and the nodal masses of the top and bottom as with discrete elements. L.EQ.1: Most conservative (smallest) stable time step estimate. This method calculates mass by integrating the density. L.EQ.2: Intermediate stable time step estimate. Same as the default, except reduced by a factor of $1/\sqrt{2}$ corresponding to halving the masses. This is the recommended setting. The third digit (in the hundred's place), M , is interpreted as follows: <ul style="list-style-type: none"> M.EQ.0: Explicit element variants used in explicit calculations and higher-accuracy implicit element variants used in implicit calculations. M.EQ.1: Higher-accuracy implicit element variants used in both implicit and explicit calculations.

VARIABLE	DESCRIPTION
TET13K	Set to 1 to invoke a consistent tangent stiffness matrix for the pressure-averaged tetrahedron (type 13). This feature is available only for the implicit integrator. This element type averages the volumetric strain over adjacent elements to alleviate volumetric locking; therefore, the corresponding material tangent stiffness should be treated accordingly. In contrast to a hexahedral mesh, where a node usually connects to fewer than eight elements, tetrahedral meshes offer no such regularity. Consequently, matrix assembly is computationally expensive for nonlinear implicit analysis, so this option is recommended only for linear or eigenvalue analysis to exploit the stiffness characteristics of the type 13 tetrahedron.
PM1 – PM10	Components of a permutation vector for nodes that define the 10-node tetrahedron. The nodal numbering of 10-node tetrahedron elements is somewhat arbitrary. The permutation vector allows other numbering schemes to be used. Unless specified, this permutation vector is not used. PM1 – PM10 are unique numbers between 1 to 10 inclusive that reorder the input node IDs for a 10-node tetrahedron into the order used by LS-DYNA.
TET13V	Choice of type 13 solid implementation: EQ.0: Efficient version (default). With the single precision version of LS-DYNA, a little noise could be observed in the solution for elements moving long distances with rigid body motion. EQ.1: More accurate version (smoother results) with an additional cost of about 15%.
RINRT	Option to compute rotational inertia for the nodes of solid elements. This ensures consistent results if the applied constraints assume rotational degrees of freedom, as with tied contacts using the option SHELL_EDGE_TO_SURFACE. EQ.0: By default, an average of the existing rotational inertia from the shell and beam elements in the model is distributed to the nodes of the solid elements. This method is sufficient in most situations but might lead to inconsistencies between different model assemblies in the case of rotational motion. EQ.1: Compute rotational inertia for each solid element based on its dimensions and mass density to ensure consistency.
COHEQC	Flag for cohesive element quality checks. COHEQC is interpreted

VARIABLE	DESCRIPTION
	digit-wise, namely as,
	$\text{COHEQC} = [LK] = K + 10 \times L$
	The first digit, K , controls the behavior of LS-DYNA when cohesive elements with poor quality are detected at $t = 0$:
	K.EQ.0: Error termination with a list of elements (default).
	K.EQ.1: Issue a warning and continue.
	K.EQ.2: Same as 1, but delete the concerned elements.
	The second digit, L , controls the behavior of LS-DYNA when inconsistent cohesive element connectivity is detected:
	L.EQ.0: Error termination with a list of elements (default).
	L.EQ.1: Issue a warning and continue.

Remarks:

1. **Computation of cohesive element time step.** As mentioned in the Theory Manual, we can estimate the stable time step of a cohesive element by representing the element as a mass-spring element. The critical time step is then:

$$\Delta t_e = \sqrt{\frac{2m}{k}},$$

where

$$m = \frac{2m_b m_t}{m_b + m_t}$$

is the harmonic mean of the top and bottom spring masses m_t and m_b . The equivalent spring stiffness (force per length) is

$$k = \sum_{i=1}^{N_{IP}} \max[E_n, E_t] w_i J_i = \max[E_n, E_t] A,$$

where E_n and E_t are the normal/tangential cohesive stiffness (stress per length), w_i are weights, and A is the interface area. Note that the last step assumes a constant stiffness and Jacobian determinant over the element.

The setting of L in the ICOH parameters determines the method for computing the top and bottom masses. For $L = 0$, LS-DYNA treats the cohesive element like a series of mass-spring elements stacked on each other. Two elements share each mass, resulting in

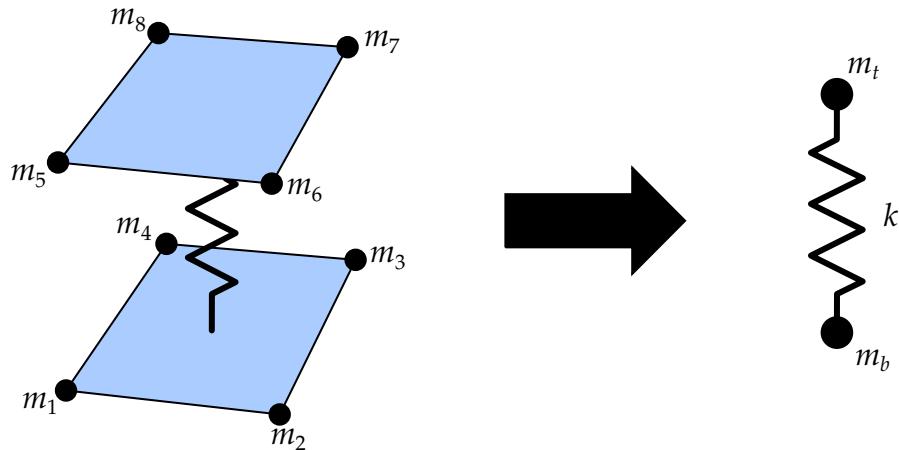


Figure 12-87. Representing a cohesive element as a mass-spring element.

$$m_t = \frac{1}{2}M_t$$

$$m_b = \frac{1}{2}M_b$$

where M_t and M_b are the sum of the top and bottom masses, respectively. That is,

$$\begin{aligned} M_t &= m_5 + m_6 + m_7 + m_8 \\ M_b &= m_1 + m_2 + m_3 + m_4 \end{aligned}$$

Note that for a pentahedral element, $m_4 = m_8 = 0$. The resulting time step is

$$\Delta t_e = \sqrt{\frac{2M_b M_t}{k(M_b + M_t)}}.$$

For $L = 1$, the mass contribution is from the cohesive element only. Thus,

$$m = \frac{1}{2}H\rho A$$

or

$$m = \frac{1}{2}\rho A ,$$

depending on whether the provided density is mass per volume or mass per area. In the above, H is the thickness. The resulting time step is

$$\Delta t_e = \sqrt{\frac{H\rho}{\max[E_n, E_t]}}$$

or

$$\Delta t_e = \sqrt{\frac{\rho}{\max[E_n, E_t]}} .$$

For $L = 2$, we assume a regular mesh such that four neighboring elements share each corner node. Thus, we have

$$m_t = \frac{1}{4} M_t$$

$$m_b = \frac{1}{4} M_b$$

These masses result in a time-step estimation of

$$\Delta t_e = \sqrt{\frac{M_b M_t}{k(M_b + M_t)}},$$

which is a factor of $\sqrt{2}$ smaller than the default $L = 0$.

The above description shows that option $L = 1$ results in the most conservative estimation, followed by $L = 2$ and, lastly, $L = 0$. However, we want to stress that the masses determined with $L = 2$ lead to the theoretical time step for two surfaces connected by cohesive interfaces, so we recommend using this option.

***CONTROL_SOLUTION**

Purpose: To specify the analysis solution procedure if thermal only or combined thermal analysis is performed. Other solution parameters including the vector length and NaN (not a number) checking can be set.

Card 1	1	2	3	4	5	6	7	8
Variable	SOLN	NLQ	ISNAN	LCINT	LCACC	NCDCF	NOCOPY	CRVP
Type	I	I	I	I	I	I	I	I
Default	0	0	0	100	0	1	0	0

VARIABLE	DESCRIPTION
SOLN	Analysis solution procedure: EQ.0: Structural analysis only EQ.1: Thermal analysis only EQ.2: Combined structural, multiphysics, and thermal analysis
NLQ	Define the vector length used in the solution. This value must not exceed the vector length of the system which varies based on the machine manufacturer. The default vector length is printed at termination in the messag file.
ISNAN	Flag to check for a NaN in the force and moment arrays after the assembly of these arrays is completed. This option can be useful for debugging purposes. A cost overhead of approximately 2% is incurred when this option is active. EQ.0: No checking EQ.1: Checking is active.
LCINT	Number of equally spaced points used in curve (*DEFINE_CURVE) rediscritization. A minimum number of 100 is always used, that is, only larger input values are possible. Curve rediscritization applies only to curves used in material models. Curves defining loads, motion, etc. are not rediscritized. Note that memory requirements increase as LCINT increases. Thus, extremely large values of LCINT should be avoided if possible.

*CONTROL

*CONTROL_SOLUTION

VARIABLE	DESCRIPTION
LCACC	<p>Flag to truncate curves to 6 significant figures for single precision and 13 significant figures for double precision. The truncation is done after applying the offset and scale factors specified in *DEFINE_CURVE. Truncation is intended to prevent curve values from deviating from the input value, such as 0.7 being stored as 0.6999999. This small deviation was seen to have an adverse effect in a particular analysis using *MAT_083. In general, curve truncation is not necessary and is unlikely to have any effect on results.</p> <p>EQ.0: No truncation NE.0: Truncate</p>
NCDCF	<p>Global option to evaluate *DEFINE_CURVE_FUNCTION every NCDCFth cycle.</p>
NOCOPY	<p><i>This field is currently disabled with NOCOPY always set to zero.</i> Avoid copying material history variables to temporary buffers for constitutive evaluations (see Remark 1):</p> <p>EQ.0: Not active EQ.1: Active</p>
CRVP	<p>Bypass time-based evaluation of non-time-dependent curves, e.g., stress-strain curves for materials. This can improve CPU performance when many curves are used (e.g. from a big material database) in smaller models.</p> <p>EQ.0: Not active EQ.1: Active EQ.2: Same as 1, but with half the memory requirement for the rediscritized curves.</p>

Remarks:

1. **NOCOPY.** Setting NOCOPY = 1 reduces overhead during element calculations and could potentially decrease the *Element processing* in *Timing Information* between 5% and 25% depending on the choice of element and material. The most substantial gain is obtained when the material is combined with a modular damage model, such as GISSMO (see [*MAT_ADD_DAMAGE_GISSMO](#)) or DIEM (see [*MAT_ADD_DAMAGE_DIEM](#)), for which another level of copying is avoided.

Currently this option is *restricted to MPP* (not hybrid). It also only works for materials 24 and 123 in combination with shell elements 2, 4 or 16, solid elements -1, -2, 1, 2, 10 or 15, and all thick shells and for materials 36 and 133 in combination with shell elements 2, 4, or 16. Other combinations of materials and elements ignore this option and will be supported in an order of priority/importance.

*CONTROL

*CONTROL_SPH

*CONTROL_SPH

Purpose: Provide controls related to SPH (Smooth Particle Hydrodynamics).

Card 1	1	2	3	4	5	6	7	8
Variable	NCBS	BOXID	DT	IDIM	NMNEIGH	FORM	START	MAXV
Type	I	I	F	I	I	I	F	F
Default	1	0	10^{20}	none	150	0	0.0	10^{15}

Remaining cards are optional.^t

Card 2	1	2	3	4	5	6	7	8
Variable	CONT	DERIV	INI	ISHOW	IEROD	ICONT	IAVIS	ISYMP
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	100

Card 3	1	2	3	4	5	6	7	8
Variable	ITHK	ISTAB	QL		SPHSORT	ISHIFT		
Type	I	I	F		I	I		
Default	0	0	0.01		0	0		

VARIABLE	DESCRIPTION
NCBS	Number of time steps between particle sorting.
BOXID	SPH approximations are computed inside a specified box (see *DEFINE_BOX). When a particle has gone outside the BOX, it is deactivated. This will save computational time by eliminating particles that no longer interact with the structure.

VARIABLE	DESCRIPTION
DT	Death time. Determines when the SPH calculations are stopped.
IDIM	Space dimension for SPH particles: EQ.3: 3D problems EQ.2: 2D plane strain problems EQ.-2: 2D axisymmetric problems (see Remark 2)
NMNEIGH	Defines the initial number of neighbors per particle (see Remark 1 below).
FORM	Particle approximation theory (Remark 2): EQ.0: Default formulation EQ.1: Renormalization approximation EQ.2: Symmetric formulation EQ.3: Symmetric renormalized approximation EQ.4: Tensor formulation EQ.5: Fluid particle approximation EQ.6: Fluid particle with renormalization approximation EQ.7: Total Lagrangian formulation EQ.8: Total Lagrangian formulation with renormalization EQ.9: Adaptive SPH formulation (ASPH) with anisotropic smoothing tensor (Remark 2g) EQ.10: Renormalization approximation for adaptive SPH formulation (ASPH) with anisotropic smoothing tensor EQ.12: Moving least-squares based formulation (MPP only, see Remark 2e) EQ.13: Incompressible formulation (MPP only, see Remark 2h) EQ.15: Enhanced fluid formulation EQ.16: Enhanced fluid formulation with renormalization
START	Start time for particle approximation. Particle approximations will be computed when time of the analysis has reached the value defined in START.

VARIABLE	DESCRIPTION
MAXV	Maximum magnitude of the velocity for SPH particles. Particles with a velocity magnitude greater than MAXV are deactivated. A value of 0.0 will turn off the velocity checking. If a negative value is used, particles with a velocity magnitude greater than MAXV will be reassigned a velocity field with a magnitude equal to MAXV .
CONT	Flag for inter-part particle interaction by “particle approximation”: EQ.0: Particle approximation is used for computing inter-part particle interaction for all SPH parts (default). EQ.1: Particle approximation is not used for inter-part particle interaction, except as specified by the INTERACTION option of *SECTION_SPH. As an alternative to particle approximation, inter-part particle interaction can be accomplished using *DEFINE_SPH_TO_SPH_COUPLING.
DERIV	Time integration type for the smoothing length: EQ.0: $\frac{d}{dt} [h(t)] = \frac{1}{d} h(t) \nabla \cdot \mathbf{v}$, (default) EQ.1: $\frac{d}{dt} [h(t)] = \frac{1}{d} h(t) (\nabla \cdot \mathbf{v})^{1/3}$
INI	Computation of the smoothing length during the initialization: EQ.0: Bucket sort based algorithm (default, very fast) EQ.1: Global computation on all the particles of the model EQ.2: Based on the mass of the SPH particle
ISHOW	Display option for deactivated SPH particles: EQ.0: No distinction in active SPH particles and deactivated SPH particles when viewing in LS-PrePost EQ.1: Deactivated SPH particles are displayed only as points and active SPH particles are displayed as spheres when Setting → SPH → Style is set to “smooth” in LS-PrePost.
IEROD	Deactivation control for SPH particles: EQ.0: Particles remain active. See Remark 3 . EQ.1: SPH particles are partially deactivated and stress states are set to 0 when erosion criteria are satisfied. See Remark 3 .

VARIABLE	DESCRIPTION
	EQ.2: SPH particles are totally deactivated and stress states are set to 0 when erosion criteria are satisfied. See Remark 3 .
	EQ.3: SPH particles are partially deactivated and stress states are set to 0 when erosion criteria are satisfied. If an EOS is defined, the volumetric response is unaffected. See Remark 3 .
ICONT	Controls contact behavior for deactivated SPH particles: EQ.0: Any contact defined for SPH remains active for deactivated particles. EQ.1: Contact is inactive for deactivated particles.
IAVIS	Defines artificial viscosity formulation for SPH elements (Remark 4): EQ.0: Monaghan type artificial viscosity formulation is used. EQ.1: Standard type artificial viscosity formulation from solid element is used (this option is not supported in SPH 2D and 2D axisymmetric elements).
ISYMP	Defines the percentage of original SPH particles used for memory allocation of SPH symmetric planes ghost nodes generation process (default is 100%). Recommended for large SPH particles models (value range 10~20) to control the memory allocation for SPH ghost particles with *BOUNDARY_SPH_SYMMETRY_PLANE keyword.
ITHK	Contact thickness option: EQ.0: The contact thickness is set to zero (default). EQ.1: The contact thickness is automatically calculated based on the volume of each SPH particle. This contact thickness calculation is ignored if a nonzero contact thickness for the SURFA surface (SAST) is provided by the contact card.
ISTAB	Stabilization type, only used when IFORM = 12: EQ.0: Incremental stabilization (default). Adequate for most materials. EQ.1: Total stabilization. Only recommended for hyperelastic materials.

VARIABLE	DESCRIPTION
QL	Quasi-Linear coefficient, only used when IFORM = 12. See Remark 5 .
SPHSORT	For the implicit solver, sort and move SPH nodes from *NODE list to the end of the list. EQ.0: No sorting (default) EQ.1: Perform sorting
ISHIFT	Flag for applying the shifting algorithm to SPH particles (available from R13.0). With the shifting algorithm, particles are advanced and then shifted slightly across streamlines. This process reduces particle clustering in the maximum compression and stretching directions. EQ.0: Do not apply shifting algorithm applied (default). EQ.1: Apply shifting algorithm applied.

Remark:

1. **NMNEIGH.** NMNEIGH is used to determine the initial memory allocation for the SPH arrays. Its value can be positive or negative. If NMNEIGH is positive, memory allocation is dynamic such that the number of neighboring particles is initially equal to NMNEIGH but that number is subsequently allowed to exceed NMNEIGH as the solution progresses. If NMNEIGH is negative, memory allocation is static and $|NMNEIGH|$ is the maximum allowed number of neighboring particles for each particle throughout the entire solution. Using this static memory option can avoid memory allocation problems.
2. **FORM.** The user must be careful to pick the right FORM which depends upon the application. Below are some guidelines for selecting the FORM value:
 - a) For most solid structure applications, FORM = 1 is recommended for more accurate results around the boundary area.
 - b) For fluid or fluid-like material applications, FORM = 15 or 16 is recommended. FORM = 16 usually has better accuracy but requires more CPU time. Also note that formulations 15 and 16 include a smoothing of the pressure field and are, therefore, not recommended for materials with failure or problems with important strain localization.
 - c) All SPH formulations with Eulerian kernel, that is, FORM with values 0 to 6, 15 and 16, can be used for large or extremely large deformation

applications but will have tensile instability issues. FORM = 2 or 3 is not recommended for any case.

- d) All SPH formulations with Lagrangian kernel (FORM = 7 or 8) can be used to avoid tensile instability issue but they cannot endure very large deformations.
 - e) For improved accuracy and tensile stability, a formulation based on moving least-squares (FORM = 12) is available. This formulation can be used for extremely large deformation applications but entails a significant computational cost. FORM = 12 is available for MPP simulations only. It is strongly recommended to keep a constant smoothing length for this formulation by setting HMIN = 1.0 and HMAX = 1.0 in *SECTION_SPH.
 - f) Only formulations 0, 1, 15 and 16 are implemented for 2D axisymmetric problems (IDIM = -2).
 - g) Formulations 9 and 10 are adaptive smoothed particle hydrodynamics formulations with an anisotropic kernel (Eulerian kernel) whose axes evolve automatically to follow the mean particles spacing as it varies in time, space and direction based on the strain rate tensors. These forms must be used with the *SECTION_SPH_ELLIPSE keyword. They have better accuracy and stability than the standard SPH. These forms can be used for extremely large deformation problems and are available for 3D case only.
 - h) Formulation 13 is an implicit incompressible fluid formulation based on operator splitting. used mainly for water wading simulations. The fluid should be assigned to a *MAT_SPH_INCOMPRESSIBLE_FLUID material, and any structure interacting with the fluid needs to be sampled with SPH particles (see the *DEFINE_SPH_MESH_SURFACE keyword) assigned to a *MAT_SPH_INCOMPRESSIBLE_STRUCTURE material. The density of the structure particles should be the same as the rest density of the interacting fluid. The time-step is calculated based on a CFL condition on the particles velocity and usually needs to be limited using the LCTM option in *CONTROL_TIMESTEP. The smoothing length of fluid and structure particles should be set to the initial inter-particle distance of the fluid, using SPHINI in *SECTION_SPH. The fluid-structure interaction is calculated entirely based on the SPH interaction of fluid and structure particles, and no *CONTACT keyword is necessary.
3. **Erosion.** The erosion criteria, which triggers particle deactivation when IEROD = 1, 2 or 3, may come from the material model, from *MAT_ADD_EROSION, or from the ERODE field in *CONTROL_TIMESTEP.
- a) For IEROD = 0, SPH particles remain active. This option is generally not recommended when materials with erosion are used, as many material

models will still reset the stress field to zero periodically. If an unaltered stress field is desired, simply remove the erosion criteria in the material model parameters.

- b) For IEROD = 1, SPH particles are partially deactivated; that is, the stress states of the deactivated SPH particles will be set to zero, but these particles still remain in the domain integration for more stable results.
 - c) For IEROD = 2, SPH particles are totally deactivated, so the stress states will be set to 0 and the deactivated particles do not remain in the domain integration.
 - d) For IEROD = 3, SPH particles remain active. The deviatoric stress is set to zero. If an equation of state is used, the volumetric response remains unaltered; otherwise the volumetric stress is set to zero as well.
 - e) Deactivated particles can be distinguished from active particles by setting ISHOW = 1.
 - f) To disable contact for deactivated particles, set ICONT = 1.
4. **Artificial Viscosity.** The artificial viscosity for standard solid elements, which is active when AVIS = 1, is given by:

$$q = \begin{cases} \rho l(Q_1 l \dot{\epsilon}_{kk}^2 - Q_2 a \dot{\epsilon}_{kk}) & \dot{\epsilon}_{kk} < 0 \\ 0 & \dot{\epsilon}_{kk} \geq 0 \end{cases}$$

where Q_1 and Q_2 are dimensionless input constants, which default to 1.5 and .06, respectively (see *CONTROL_BULK_VISCOSITY); l is a characteristic length given as the square root of the area in two dimensions and as the cube root of the volume in three; and a is the local sound speed. This formulation, which is consistent with solid artificial viscosity, has better energy balance for SPH elements.

For general applications, Monaghan type artificial viscosity is recommended since this type of artificial viscosity is specifically designed for SPH particles. The Monaghan type artificial viscosity, which is active when AVIS = 0, is defined as follows:

$$q = \begin{cases} \frac{-Q_2 \bar{c}_{ij} \phi_{ij} + Q_1 \phi_{ij}^2}{\bar{\rho}_{ij}} & v_{ij} x_{ij} < 0 \\ 0 & v_{ij} x_{ij} \geq 0 \end{cases}$$

where

$$\phi_{ij} = \frac{h_{ij} v_{ij} x_{ij}}{|x_{ij}|^2 + \varphi^2}$$

$$\begin{aligned}\bar{c}_{ij} &= 0.5(c_i + c_j) \\ \bar{\rho}_{ij} &= 0.5(\rho_i + \rho_j) \\ h_{ij} &= 0.5(h_i + h_j) \\ \varphi &= 0.1h_{ij}\end{aligned}$$

and Q_1 and Q_2 are input constants. When using Monaghan type artificial viscosity, it is recommended that the user set both Q_1 and Q_2 to 1.0 on either the *CONTROL_BULK_VISCOSITY or *HOURGLASS keywords; see for example G. R. Liu.

5. **Quasi-Linear Coefficient.** The moving least-squares based formulation contains a quasi-linear approximation term to combine accuracy with stability in extremely large deformations simulations. The default value $QL = 0.01$ gives a good compromise between accuracy and stability in most cases. For greater accuracy, its value can be reduced to $QL = 0.001$ in simulations with small deformations, or increased to $QL = 0.1$ for extreme deformations, if instabilities are present.

*CONTROL

*CONTROL_SPH_INCOMPRESSIBLE

*CONTROL_SPH_INCOMPRESSIBLE

Purpose: Provide controls relating to incompressible SPH (FORM = 13; see *CONTROL_SPH for more information).

Card 1	1	2	3	4	5	6	7	8
Variable	IBNDP	TAVG	TMAX	ROL	IHTC	IMAT	IRMV	ACMP
Type	I	F	F	F	I	I	I	F
Default	0	10^{-2}	10^{20}	10^{20}	0	0	0	0.0

VARIABLE	DESCRIPTION
IBNDP	Pressure treatment of boundary particles: EQ.0: Pressure on boundary particles is extrapolated from fluid particles. EQ.1: Pressure on boundary particles is explicitly calculated.
TAVG	Tolerance criteria for convergence. If the average relative density (ρ/ρ_0) of particles under compression is below TAVG, this condition is satisfied.
TMAX	Tolerance criteria for convergence. If the maximum relative density (ρ/ρ_0) of particles under compression is below TMAX, this condition is satisfied.
ROL	Stuck particle detection criteria. In certain scenarios, some fluid particles can end up stuck between moving structures and as a result accumulate very large pressure values. If a particle's relative density contribution from boundaries is above ROL, it is deactivated. Its displacement is prescribed from the interpolated structure motion until it is sufficiently away from the structure at which point it is activated again.
IHTC	Flag for Heat Transfer Coefficient (HTC) calculation: EQ.0: HTCs are not calculated. EQ.1: HTCs are calculated based on fluid properties given in *MAT_SPH_INCOMPRESSIBLE_FLUID input. LT.0: IHTC is a function ID defining HTCs based on seven

VARIABLE	DESCRIPTION
	arguments: heat capacity, thermal conductivity, dynamic viscosity, density, tangential velocity relative to the closest segment, distance traveled along the surface, and the ratio of the distance from the closest segment to particle radius. (See *DEFINE_FUNCTION.)
IMAT	Flag for *MAT_SPH_INCOMPRESSIBLE_FLUID formulations: EQ.0: Surface tension and surface adhesion forces are calculated based on numerical parameters given in the material cards. EQ.1: Surface tension and surface adhesion forces are calculated based on the physical properties given in the material cards.
IRMV	Remove initially interpenetrated particles. Fluid particles that are too close to structural particles will be deactivated on initialization: EQ.0: Do not remove interpenetrated particles. EQ.1: Remove interpenetrated particles.
ACMP	Artificial compressibility term. For an analysis with a time-varying time step, we strongly recommend adding artificial compressibility to avoid oscillations and instabilities. We typically recommend a small value of $1.0 \times 10^{-8} \text{ Pa}^{-1}$. Note that this parameter is unit sensitive. Thus, the recommended value needs to be converted to the unit system used in the model.

*CONTROL

*CONTROL_SPOTWELD_BEAM

*CONTROL_SPOTWELD_BEAM

Purpose: Provides factors for scaling the failure force resultants of beam spot welds as a function of their parametric location on the contact segment and the size of the segment. Also, this keyword can optionally replace beam welds with solid hexahedron element clusters.

Card 1	1	2	3	4	5	6	7	8
Variable	LCT	LCS	T_ORT	PRTFLG	T_ORS	RPBHX	BMSID	ID_OFF
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

VARIABLE	DESCRIPTION
LCT	Load curve ID for scaling the response in tension based on the shell element size.
LCS	Load curve ID for scaling the response in shear based on the shell element size.
T_ORT	Table ID for scaling the tension response (and shear response if T_ORS = 0) based on the location of the beam node relative to the centroid of the shell.
PRTFLG	Set this flag to 1 to print for each spot weld attachment: the beam, node, and shell IDs, the parametric coordinates that define the constraint location, the angle used in the table lookup, and the three scale factors obtained from the load curves and table lookup. See Figure 12-88 .
T_ORS	Optional table ID for scaling the shear response based on the location of the beam node relative to the centroid of the shell.
RPBHX	Replace each spot weld beam element with a cluster of RPBHX solid elements. The net cross-section of the cluster of elements is dimensioned to have the same area as the replaced beam. RPBHX may be set to 1, 4, or 8. When RPBHX is set to 4 or 8, a table is generated to output the force and moment resultants into the SWFORC file if this file is active. *DEFINE_HEX_SPOTWELD_ASSEMBLY describes this table. LS-DYNA uses the IDs of the beam elements as the cluster spot weld IDs, so the IDs in the

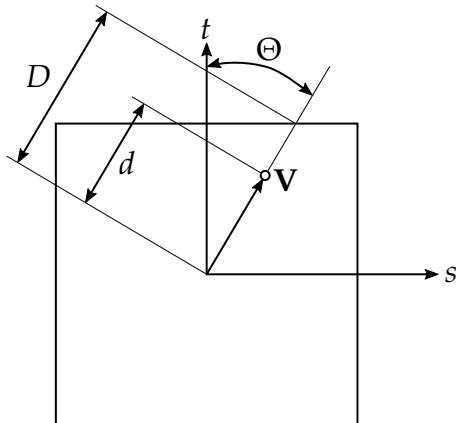


Figure 12-88. Definition of parameters for table definition.

VARIABLE	DESCRIPTION
	SWFORC file are unchanged. The beam elements are automatically deleted from the calculation, and the section and material data are automatically changed to be used with solid elements. See Figure 17-55 .
BMSID	Optional beam set ID defining the beam element IDs that are to be converted to hex assemblies. If zero, all spot weld beam elements are converted to hex assemblies.
ID_OFF	This optional ID offset applies if and only if BMSID is nonzero. Beams, which share part IDs with beams that are converted to hex assemblies, will be assigned new part IDs by adding to the original part ID the value of ID_OFF. If ID_OFF is zero, the new part ID for such beams will be assigned to be larger than the largest part ID in the model.

Remarks:

The load curves and table provide a means of scaling the response of the beam spot welds to reduce any mesh dependencies for failure model 6 in *MAT_SPOTWELD. [Figure 12-89](#) shows such dependencies that can lead to premature spot weld failure. Separate scale factors are calculated for each of the beam's nodes. The scale factors s_T , s_s , s_{OT} , and s_{OS} are calculated using LCT, LCS, T_ORT, and T_ORS, respectively, and are introduced in the failure criteria,

$$\left[\frac{s_T s_{OT} \sigma_{rr}}{\sigma_{rr}^F(\dot{\varepsilon}_{eff})} \right]^2 + \left[\frac{s_s s_{OS} \tau}{\tau^F(\dot{\varepsilon}_{eff})} \right]^2 - 1 = 0 .$$

If a curve or table is given an ID of 0, its scale factor is set to 1.0. The load curves LCT and LCS are functions of the characteristic size of the shell element used in the time step calculation at the start of the calculation. The orientation table is a function of the spot weld's

*CONTROL

*CONTROL_SPOTWELD_BEAM

isoparametric coordinate location on the shell element. We define a vector \mathbf{V} from the centroid of the shell to the contact point of the beam's node. The arguments for the orientation table are the angle:

$$\Theta = \tan^{-1} \left[\frac{\min(|s|, |t|)}{\max(|s|, |t|)} \right],$$

and the normalized distance $\bar{d} = d/D = \max(|s|, |t|)$. See [Figure 12-88](#). The table is periodic over a range of 0 (\mathbf{V} aligned with either the s or t axis) to 45 degrees (\mathbf{V} is along the diagonal of the element). The angle of \mathbf{V} in degrees, ranging from 0 to 45, specifies the table, and the individual curves give the scale factor as a function of the normalized distance of the beam node, \bar{d} , for a constant angle.

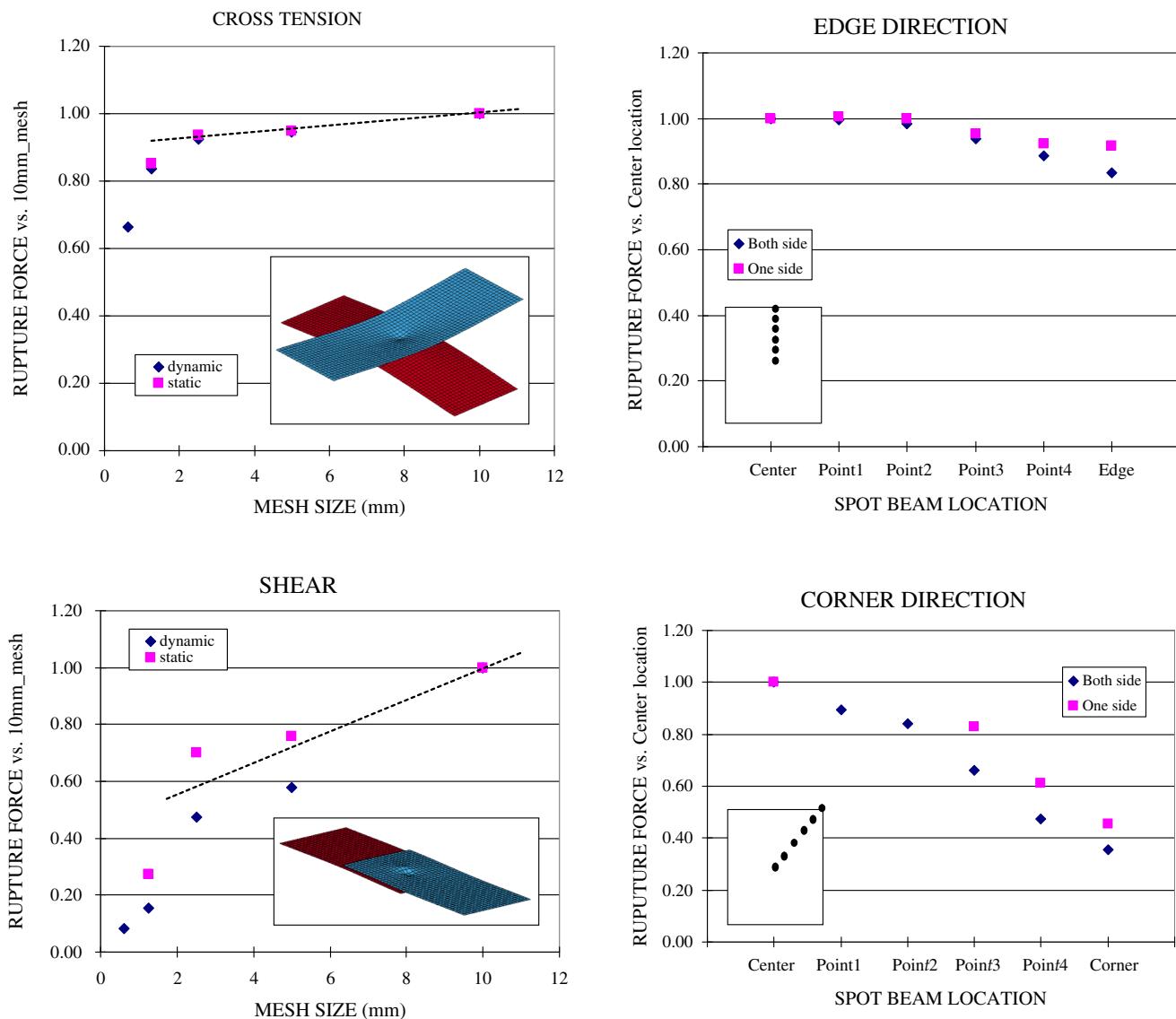


Figure 12-89 The failure force resultants can depend both on mesh size and the location of weld relative to the center of the contact segment

CONTROL_STAGED_CONSTRUCTION**CONTROL*****CONTROL_STAGED_CONSTRUCTION**

Purpose: Help break down analyses of construction processes into stages.

Card 1	1	2	3	4	5	6	7	8
Variable	TSTART	STGS	STGE	ACCEL	FACT	blank	DORDEL	NOPDEL
Type	F	I	I	F	F		I	I
Default	0.	0	0	0.0	10^{-6}		0	0

This card is optional.

Card 2	1	2	3	4	5	6	7	8
Variable	ITIME	blank	IDYNAIN					
Type	I		I					
Default	0		0					

VARIABLE	DESCRIPTION
TSTART	Time at start of analysis (normally leave blank). See Remark 5 .
STGS	Construction stage at start of analysis
STGE	Construction stage at end of analysis
ACCEL	Default acceleration for gravity loading
FACT	Default stiffness and gravity factor for parts before they are added
DORDEL	Dormant part treatment in d3plot file (see Remark 6): EQ.0: Parts not shown when dormant (flagged as "deleted"). EQ.1: Parts shown normally when dormant.

*CONTROL

*CONTROL_STAGED_CONSTRUCTION

VARIABLE	DESCRIPTION
NOPDEL	Treatment of pressure loads on deleted elements (see Remark 7): EQ.0: Pressure loads automatically deleted. EQ.1: No automatic deletion.
ITIME	Treatment of “Real Time” on *DEFINE_CONSTRUCTION_STAGES (see Remark 8): EQ.0: Real Time is ignored. EQ.1: Time in output files (d3plot, d3thdt, binout...) is converted to Real Time.
IDYNAIN	Flag to control output of dynain file at the end of every stage (see Remark 9): EQ.0: Write dynain file. EQ.1: Do not write dynain file.

Remarks:

1. **Related Keywords.** See also *DEFINE_CONSTRUCTION_STAGES and *DEFINE_STAGED_CONSTRUCTION_PART.
2. **Re-Running Analysis.** The staged construction options offer flexibility to carry out the whole construction simulation in one analysis or to run it stage by stage. Provided that at least one construction stage is defined (*DEFINE_CONSTRUCTION_STAGES), a dynain file will be written at the end of each stage (file names are end_stage001_dynain, etc.). These contain node and element definitions and the stress state; the individual stages can then be re-run without re-running the whole analysis. To do this, make a new input file as follows:
 - Copy the original input file, containing *DEFINE_CONSTRUCTION_STAGES and *DEFINE_STAGED_CONSTRUCTION_PART.
 - Delete node and element definitions as these will be present in the dynain file (*NODE, *ELEMENT_SOLID, *ELEMENT_SHELL, and *ELEMENT_BEAM).
 - Delete any *INITIAL cards; the initial stresses in the new analysis will be taken from the dynain file.
 - On *CONTROL_STAGED_CONSTRUCTION set STGS to start at the desired stage.

- Add an *INCLUDE statement referencing, for example, end_stage002_dynain if starting the new analysis from Stage 3.
 - Move or copy the dynain file into the same directory as the new input file.
3. **Starting and Ending Stages.** When STGS is > 1 the analysis starts at a non-zero time (the start of stage STGS). In this case a dynain file must be included to start the analysis from the stress state at the end of the previous stage. The end time for stage STGE overrides the termination time on *CONTROL_TERMINATION. A new dynain file will be written at the end of all stages from STGS to STGE.
- If STGS > 1 and elements have been deleted in a previous stage, these elements will be absent from the new analysis and should not be referred to (for example, *DATABASE_HISTORY_SOLID) in the new input file.
4. **ACCEL and FACT.** ACCEL is the gravity loading applied to parts referenced by *DEFINE_STAGED_CONSTRUCTION_PART. FACT, which should be a very small number such as 10^{-6} , is the factor by which stresses and gravity load are scaled when parts referenced by *DEFINE_STAGED_CONSTRUCTION_PART are inactive.
5. **Start Time.** TSTART can be used to set a non-zero start time (again, assuming a compatible dynain file is included). This option is used only if construction stages have not been defined.
6. **Dormant Parts.** By default, parts for which *DEFINE_STAGED_CONSTRUCTION_PART is defined are flagged as “deleted” in the d3plot file at time-states for which the part is not active (that is, stage STGA has not yet been reached). Parts that are deleted because STGR has been reached are also flagged as “deleted.” When animating the results, the parts become visible when they become active, as if they had been added to the model at that time; and they disappear as they are deleted. If DORDEL is non-zero, inactive parts (before STGA) are shown normally. The parts are still shown as deleted after STGR is reached.
7. **Pressure Load Segments.** By default, LS-DYNA automatically deletes pressure load “segments” if they share all four nodes with a deleted solid or shell element. In staged construction, you may want to apply a pressure load to the surface of an element, A, that is initially shared with an element, B, where B is deleted during the calculation. For example, B may be in a layer of soil that is excavated, leaving A as the new top surface. The default scheme would delete the pressure segment when B is removed, despite A still being present. NOPDEL instructs LS-DYNA to skip the automatic deletion of pressure segments, irrespective of whether the elements have been deleted due to staged construction or material failure. You must then ensure that pressure loads are not applied to nodes no longer supported by an active element.

8. **Real Time.** Construction processes that in real life take days, weeks, or months may be modelled in seconds of analysis time using a “time acceleration” method. For example, one second of analysis time might represent ten days of real time. The speed-up factor might differ at different stages of the analysis. You may, however, want to present plots and graphs with real time rather than the analysis time used in LS-DYNA’s calculations. To do this, set ITIMES to 1, and set RTS and RTE (see optional fields RTS and RTE on *DEFINE_CONSTRUCTION_STAGES) to the real time at the start and end of each stage. This feature has no effect on the calculations as it changes only the times shown in the d3plot, d3thdt and binout files.
9. **IDYNAIN.** Output of the dynain files can be suppressed by setting IDYNAIN to 1. If dynain files are required for some stages but not others, set IDYNAIN on *DEFINE_CONSTRUCTION_STAGES instead.

CONTROL_START**CONTROL*****CONTROL_START**

Purpose: Define the start time of analysis.

Card 1	1	2	3	4	5	6	7	8
Variable Type	BEGTIM F							

VARIABLE	DESCRIPTION
BEGTIM	Start time of analysis (default = 0.0). <i>Load curves are not shifted to compensate for the time offset.</i> Therefore, this keyword will change the results of any calculation involving time-dependent load curves.

*CONTROL

*CONTROL_STEADY_STATE_ROLLING

*CONTROL_STEADY_STATE_ROLLING

Card 1	1	2	3	4	5	6	7	8
Variable	IMASS	LCDMU	LCDMUR	IVEL	SCL_K			
Type	I	I	I	I	I			
Default	0	0	0	0	↓			

VARIABLE	DESCRIPTION
IMASS	Inertia switching flag: EQ.0: include inertia during an implicit dynamic simulation. EQ.1: treat steady state rolling subsystems as quasi-static during implicit dynamic simulations.
LCDMU	Optional load curve for scaling the friction forces in contact.
LCDMUR	Optional load curve for scaling the friction forces in contact during dynamic relaxation. If LCDMUR isn't specified, LCDMU is used.
IVEL	Velocity switching flag: EQ.0: eliminate the steady state rolling body forces and set the velocities of the nodes after dynamic relaxation. EQ.1: keep the steady state rolling body forces after dynamic relaxation instead of setting the velocities.
SCL_K	Scale factor for the friction stiffness during contact loading and unloading. The default values are 1.0 and 0.01 for explicit and implicit, respectively. Any scaling applied here applies only to contact involving the subsystem of parts defined for steady state rolling.

Remarks:

1. **Quasi-Static.** Treating the steady state rolling subsystems as quasi-static during an implicit simulation may eliminate vibrations in the system that are not of interest and is generally recommended.

2. **Scaling Friction and Convergence.** Ramping up the friction by scaling it with LCDMU and LCDMUR may improve the convergence behavior of implicit calculations. The values of the load curves should be 0.0 at initial contact and ramp up smoothly to a value of 1.0.
3. **Steady State Rolling Body Forces.** After dynamic relaxation, the default behavior is to initialize the nodes with the velocities required to generate the body forces on elements and remove the body forces. This initialization is skipped, and the body forces retained, after dynamic relaxation if IVEL = 1.
4. **Friction Stiffness Scale Factor.** The friction model in contact is similar to plasticity, where there is an elastic region during the loading and unloading of the friction during contact. The elastic stiffness is scaled from the normal contact stiffness. For implicit calculations, the default scale factor is 0.01, which results in long periods of time being required to build the friction force, and, in some cases, oscillations in the contact forces. A value between 10 and 100 produces smoother solutions and a faster build-up and decay of the friction force as the tire velocity or slip angle is varied, allowing a parameter study to be performed in a single run.

***CONTROL_STRUCTURED_{OPTION}**

Available options include:

<BLANK>

TERM

Purpose: Write out an LS-DYNA structured input deck that is largely or wholly equivalent to the keyword input deck. This option may be useful in debugging errors that occur during processing of the input file, particularly if error messages of the type “*** ERROR ##### (STR + ###)” are written. The name of the structured input deck is “dyna.str”.

Not all LS-DYNA features are supported in structured input format. Some data such as load curve numbers will be output in an internal numbering system.

If the TERM option is activated, termination will occur after the structured input deck is written.

Adding “outdeck = s” to the LS-DYNA execution line serves the same purpose as including *CONTROL_STRUCTURED in the keyword input deck.

***CONTROL_SUBCYCLE_{K}_{L}** or
***CONTROL_SUBCYCLE_{OPTION}**

Available options for subcycling first form with K and L

$K, L \in \{\text{<BLANK>} , 1, 2, 4, 8, 16, 32, 64\}$

Available options for multiscale (OPTION) include:

<BLANK>

MASS_SCALED_PART

MASS_SCALED_PART_SET

Purpose: This keyword is used to activate subcycling or mass scaling (multiscale). The common characteristic of both methods is that the time step varies from element to element, thereby eliminating unnecessary stepping on more slowly evolving portions of the model. These techniques are suited for reducing the computational cost for models involving large spatial variation in mesh density and/or material characteristics.

Subcycling is described in the LS-DYNA Theory Manual and in detail in Borrvall et.al. [2014] and may be seen as an alternative to using selective mass scaling, see the keyword *CONTROL_TIMESTEP.

This keyword comes in two variations:

1. **Subcycling.** Plain subcycling is activated by the *CONTROL_SUBCYCLE_{K}_{L} variant of this keyword. This form of the card should *not* be included more than once. It may be used in conjunction with mass scaling to limit the time step characteristics.

For subcycling, time steps for integration are determined automatically from the characteristic properties of the elements in the model, with the restriction that the ratio between the largest and smallest time step is limited by K . Furthermore, L determines the relative time step at which external forces such as contacts and loads are calculated

For example, *CONTROL_SUBCYCLE_16_4 limits the largest explicit integration time step to at most 16 times the smallest. Contact forces are evaluated every 4 time steps. The defaults are $K = 16$ and $L = 1$, and L cannot be specified larger than K . This option may be used without mass scaling activated but internally elements may still be slightly mass scaled to maintain computational efficiency.

2. **Mass Scaling/Multiscale.** For a multiscale simulation, mass scaling is mandatory and the time steps are directly specified in the input. The specified parts (see the PID field) or part sets (see the PSID field) run at the time step specified

*CONTROL

*CONTROL_SUBCYCLE

in the TS field. All other elements evolve with a time step set by |DT2MS|, which is set on *CONTROL_TIMESTEP card.

This feature was motivated by automotive crash simulation, wherein it is common for a small subset of *solid* elements to limit the time step size. With this card the finely meshed parts (consisting of solid elements) can be made to run with a smaller time step through mass scaling so that the rest of the vehicle can run with a time step size of |DT2MS|.

Part Card. Additional card for the MASS_SCALED_PART and MASS_SCALED_PART_SET keyword options. Provide as many cards as necessary. Input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID/PSID	TS						
Type	I	F						
Default	none	none						

VARIABLE	DESCRIPTION
PID/PSID	Part ID or part set ID if the SET option is specified.
TS	Time step size at which mass scaling is invoked for the PID or PSID

CONTROL_TERMINATION**CONTROL*****CONTROL_TERMINATION**

Purpose: Stop the job.

Card 1	1	2	3	4	5	6	7	8
Variable	ENDTIM	ENDCYC	DTMIN	ENDENG	ENDMAS	NOSOL		
Type	F	I	F	F	F	I		
Default	0.0	0	0.0	0.0	10^8	0		
Remarks	1		2					

VARIABLE**DESCRIPTION**

ENDTIM

Termination time. Mandatory.

ENDCYC

Termination cycle. The termination cycle is optional and will be used if the specified cycle is reached before the termination time. Cycle number is identical with the time step number.

DTMIN

Reduction (or scale) factor to determine minimum time step, tsmin, where $tsmin = dtstart \times DTMIN$ and dtstart is the initial step size determined by LS-DYNA. When the time step drops to tsmin, LS-DYNA terminates with a restart dump. See the exception described in [Remark 2](#). Also note that dtstart is the initial step size regardless of whether the first step is explicit or implicit; this can have implications if an analysis starts as implicit and later switches to explicit, resulting in a large drop in step size.

ENDENG

Percent change in energy ratio for termination of calculation. If undefined, this option is inactive.

ENDMAS

Percent change in the total mass for termination of calculation. This option is relevant if and only if mass scaling is used to limit the minimum time step size; see *CONTROL_TIMESTEP field DT2MS.

LT.0.0: |ENDMAS| is the load curve ID defining the percent change in the total mass as a function of the total mass.

NOSOL

Flag for a non-solution run, that is, normal termination directly after initialization.

*CONTROL

*CONTROL_TERMINATION

VARIABLE	DESCRIPTION
	EQ.0: Off (default), EQ.1: On.

Remarks:

1. **Displacement Termination.** Termination by displacement may be defined in the *TERMINATION section.
2. **Erosion.** If the erosion flag on *CONTROL_TIMESTEP is set (ERODE = 1), then solid elements and thick shell elements whose time step falls below tsmin will be eroded and the analysis will continue. This time-step-based failure option is not recommended when solid formulations 11 or 12 are included in the model. Furthermore, when PSFAIL in *CONTROL_SOLID is nonzero, regardless of the value of ERODE, then all solid elements excepting those with formulation 11 or 12, whose time step falls below tsmin will be eroded and the analysis will continue. This time-step-based erosion of solids due to a nonzero PSFAIL is not limited to solids in part set PSFAIL. Only the negative-volume-based erosion criterion is limited to solids in part PSFAIL.

***CONTROL_THERMAL_EIGENVALUE**

Purpose: Compute eigenvalues of thermal conductance matrix for model evaluation purposes.

Card 1	1	2	3	4	5	6	7	8
Variable	NEIG							
Type	I							
Default	0						.	.

VARIABLE	DESCRIPTION
NEIG	Number of eigenvalues to compute: EQ.0: No eigenvalues are computed. GT.0: Compute NEIG eigenvalues of each thermal conductance matrix.

Remarks:

1. **Feature Description.** This feature causes LS-DYNA to compute NEIG eigenvalues for each thermal conductance matrix. This tool is for model evaluation, so only a small number of thermal time steps, such as 1, should be used with this feature.

*CONTROL

*CONTROL_THERMAL_FORMING

*CONTROL_THERMAL_FORMING

Purpose: Simplify keyword input deck by defining key control parameters for hot stamping simulations. A hot stamping simulation is in general a thermo-mechanical-coupled analysis, which requires adding a set of thermal control keywords to fully define the simulation job. However, for certain situations this keyword can simplify input by reducing the number of keywords required.

Card Summary:

Card 1. This card is required.

ITS	PTYPE	TSF	THSHEL	ITHOFF	SOLVER	FWORK	
-----	-------	-----	--------	--------	--------	-------	--

Card 2.1. The card set, Cards 2.1 and 2.2, is optional. When present Card 2.1 sets the default values for the fields of “[THRM 1](#)” on *CONTACT and “THRM 1” becomes optional.

K	FRAD	H0	LMIN	LMAX	FTOSA	BC_FLG	ALGO
---	------	----	------	------	-------	--------	------

Card 2.2. The card set, Cards 2.1 and 2.2, is optional. When present Card 2.2 sets the default values for the fields of “[THRM 2](#)” on *CONTACT and “THRM 2” becomes optional.

LCFST	LCFDT	FORMULA	A	B	C	D	LCH
-------	-------	---------	---	---	---	---	-----

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	ITS	PTYPE	TSF	THSHEL	ITHOFF	SOLVER	FWORK	
Type	F	I	F	I	I	I	F	
Default	none	0	1.	0	0	3	1.0	

VARIABLE	DESCRIPTION
ITS	Initial thermal time step size
PTYPE	Thermal problem type (see Remark 1 for determining the type of problem):

VARIABLE	DESCRIPTION
	EQ.0: Linear problem EQ.1: Nonlinear problem with material properties evaluated at the temperature of the gauss point EQ.2: Nonlinear problem with material properties evaluated at the average temperature of the element
TSF	Thermal Speedup Factor. This factor multiplies all thermal parameters with units of time in the denominator (such as thermal conductivity and convection heat transfer coefficients). It is used to artificially scale the problem in time. For example, if the velocity of the stamping punch is artificially increased by 1000, then set TSF = 1000 to scale the thermal parameters.
THSHEL	Thermal shell option: EQ.0: No temperature gradient is considered through the shell thickness. EQ.1: A temperature gradient is calculated through the shell thickness.
ITHOFF	Flag for offsetting thermal contact surfaces for thick thermal shells: EQ.0: No offset; if thickness is not included in the contact, the heat will be transferred between the mid-surfaces of the corresponding contact segments (shells). EQ.1: Offsets are applied so that contact heat transfer is always between the outer surfaces of the contact segments (shells).
SOLVER	Thermal analysis solver type (see *CONTROL_THERMAL_-SOLVER). For SMP only: EQ.1: Using solver 11 (enter -1 to use the old ACTCOL solver) EQ.2: Nonsymmetric direct solver EQ.3: Diagonal scaled conjugate gradient iterative (default) EQ.4: Incomplete choleski conjugate gradient iterative EQ.5: Nonsymmetric diagonal scaled bi-conjugate gradient For SMP or MPP: EQ.11: Direct solver

*CONTROL

*CONTROL_THERMAL_FORMING

VARIABLE	DESCRIPTION							
	EQ.12: Diagonal scaling conjugate gradient iterative (default)							
	EQ.13: Symmetric Gauss-Seidel conjugate gradient iterative							
	EQ.14: SSOR conjugate gradient iterative							
	EQ.15: ILDLT0 (incomplete factorization) conjugate gradient iterative							
	EQ.16: Modified ILDLT0 (incomplete factorization) conjugate gradient iterative							
	For Conjugate Heat transfer problems in SMP or MPP:							
	EQ.17: GMRES solver							
FWORK	Fraction of mechanical work converted into heat							
	EQ.0.0: Use default value 1.0.							

Card 2.1	1	2	3	4	5	6	7	8
Variable	K	FRAD	H0	LMIN	LMAX	FTOSA	BC_FLG	ALGO
Type	F	F	F	F	F	F	I	I
Default	none	none	none	none	none	0.5	0	0

VARIABLE	DESCRIPTION
K	Thermal conductivity of fluid between the contact surfaces. If a gap with a thickness l_{gap} exists between the contact surfaces, then the conductance due to thermal conductivity between the contact surfaces is
	$h_{\text{cond}} = \frac{K}{l_{\text{gap}}}$
FRAD	Note that LS-DYNA calculates l_{gap} based on deformation.
	Radiation factor between the contact surfaces.
	$f_{\text{rad}} = \frac{\sigma}{\frac{1}{\varepsilon_1} + \frac{1}{\varepsilon_2} - 1} ,$
	where

VARIABLE	DESCRIPTION
	σ = Stefan-Boltzman constant ε_1 = emissivity of SURFB surface ε_2 = emissivity of SURFA surface
	LS-DYNA calculates a radiant heat transfer conductance
	$h_{\text{rad}} = f_{\text{rad}}(T_{\text{SURFA}} + T_{\text{SURFB}})(T_{\text{SURFA}}^2 + T_{\text{SURFB}}^2)$
H0	Heat transfer conductance for closed gaps. Use this heat transfer conductance for gaps in the range
	$0 \leq l_{\text{gap}} \leq l_{\text{min}}$
LMIN	Minimum gap, l_{min} ; use the heat transfer conductance defined (H0) for gap thicknesses less than this value. LT.0.0: -LMIN is a load curve ID defining l_{min} as a function of time.
LMAX	No thermal contact if gap is greater than this value (l_{max}).
FTOSA	Fraction, f , of sliding friction energy partitioned to the SURFA surface. Energy partitioned to the SURFB surface is $(1 - f)$. EQ.0: Default set to 0.5. The sliding friction energy is partitioned 50% - 50% to the SURFA and SURFB surfaces in contact.
	$f = \frac{1}{1 + \frac{\sqrt{(\rho C_p k)}_{\text{SURFB side material}}}{\sqrt{(\rho C_p k)}_{\text{SURFA side material}}}}$
BC_FLAG	Thermal boundary condition flag: EQ.0: Thermal boundary conditions are on when parts are in contact. EQ.1: Thermal boundary conditions are off when parts are in contact.
ALGO	Contact algorithm type. EQ.0: Two-way contact; both surfaces change temperature due to contact. EQ.1: One-way contact; SURFB surface does not change temperature due to contact. SURFA surface does change temperature.

CONTROL**CONTROL_THERMAL_FORMING**

Card 2.2	1	2	3	4	5	6	7	8
Variable	LCFST	LCFDT	FORMULA	A	B	C	D	LCH
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

VARIABLE	DESCRIPTION
LCFST	Load curve number for static coefficient of friction as a function of temperature. The load curve value multiplies the coefficient value FS.
LCFDT	Load curve number for dynamic coefficient of friction as a function of temperature. The load curve value multiplies the coefficient value FD.
FORMULA	<p>Formula that defines the contact heat conductance as a function of temperature and pressure.</p> <p>EQ.1: $h(P)$ is defined by load curve A, which contains data for contact conductance as a function of pressure.</p> <p>EQ.2: $h(P)$ is given by the following where A, B, C and D although defined by load curves are typically constants for use in this formula. The load curves are given as functions of temperature.</p> $h(P) = a + bP + cP^2 + dP^3$ <p>EQ.3: $h(P)$ is given by the following formula from [Shvets and Dyban 1964].</p> $h(P) = \frac{\pi k_{\text{gas}}}{4\lambda} \left[1. + 85 \left(\frac{P}{\sigma} \right)^{0.8} \right] = \frac{a}{b} \left[1. + 85 \left(\frac{P}{c} \right)^{0.8} \right],$ <p>where</p> <p>a: is evaluated from the load curve, A, for the thermal conductivity, k_{gas}, of the gas in the gap as a function of temperature.</p> <p>b: is evaluated from the load curve, B, for the parameter grouping $\pi/4\lambda$. Therefore, this load curve should be set to a constant value. λ is the surface roughness.</p> <p>c: is evaluated from the load curve, C, which specifies a stress metric for deformation (e.g., yield) as a function of</p>

VARIABLE	DESCRIPTION
	temperature.
	EQ.4: $h(P)$ is given by the following formula from [Li and Sellars 1996].
	$h(P) = a \left[1 - \exp \left(-b \frac{P}{c} \right) \right]^d ,$
	where
	<i>a</i> : is evaluated from the load curve, A, which defines a load curve as a function of temperature.
	<i>b</i> : is evaluated from the load curve, B, which defines a load curve as a function of temperature.
	<i>c</i> : is evaluated from the load curve, C, which defines a stress metric for deformation (e.g., yield) as a function of temperature.
	<i>d</i> : is evaluated from the load curve D, which is a function of temperature.
	EQ.5: $h(\text{gap})$ is defined by load curve A, which contains data for contact conductance as a function of interface gap.
	LT.0: This is equivalent to defining the keyword *USER_INTERFACE_CONDUCTIVITY. The user subroutine usrh-con will be called for this contact interface to define the contact heat transfer coefficient.
A	Load curve ID for the <i>a</i> coefficient used in the formula.
B	Load curve ID for the <i>b</i> coefficient used in the formula.
C	Load curve ID for the <i>c</i> coefficient used in the formula.
D	Load curve ID for the <i>d</i> coefficient used in the formula.
LCH	Load curve ID for <i>h</i> . This parameter can refer to a curve ID (see *DEFINE_CURVE) or a function ID (see *DEFINE_FUNCTION). When LCH is a curve ID (and a function ID) it is interpreted as follows: GT.0: The heat transfer coefficient is defined as a function of time, <i>t</i> , by a curve consisting of $(t, h(t))$ data pairs. LT.0: The heat transfer coefficient is defined as a function of temperature, <i>T</i> , by a curve consisting of $(T, h(T))$ data pairs. When the reference is to a function it is prototyped as follows $h = h(t, T_{\text{avg}}, T_{\text{slv}}, T_{\text{msr}}, P, g)$ where:

*CONTROL

*CONTROL_THERMAL_FORMING

VARIABLE

DESCRIPTION

t = solution time
 T_{avg} = average interface temperature
 T_{SURFA} = SURFA segment temperature
 T_{SURFB} = SURFB segment temperature
 P = interface pressure
 g = gap distance between SURFA and SURFB segments

Remarks:

- Problem Type.** The thermal problem becomes nonlinear when any of the following applies: (1) the material properties are temperature-dependent, (2) the thermal loads are temperature-dependent (for example, the convection coefficient is a function of temperature), (3) radiation exists.
- Heat Transfer at the Contact Interface between the Die and Blank.** The heat transfer coefficient h at the contact interface between the die and blank can be summarized as follows:

$$h = \begin{cases} h_0 & 0 \leq l_{\text{gap}} \leq l_{\min} \\ h_{\text{cond}} + h_{\text{rad}} & l_{\min} < l_{\text{gap}} \leq l_{\max} \\ 0 & l_{\text{gap}} > l_{\max} \end{cases}$$

Examples:

The following input defines the thermal control parameters for a hot stamping simulation using *CONTROL_THERMAL_FORMING:

```
*CONTROL_THERMAL_FORMING
$   its      ptype      tsf      THSHEL      ITHOFF      solver      fwork
  1.0e-5      1       10.          
```

This is equivalent to defining the following set of keywords:

```
*CONTROL_SOLUTION
$#   soln      nlq      isnan      lcint      lcacc
    2
*CONTROL_THERMAL_SOLVER
$#   atype      ptype      solver      cgtol      gpt      eqheat      fwork      sbc
    1       1       11      0      1.000000      1.00e-10      5.67e-11
$#   MSGLVL      MAXITR      ABSTOL      RELTOL      OMEGA      TSF
      MAXITR      ABSTOL      RELTOL      OMEGA      TSF
      10.
*CONTROL_THERMAL_TIMESTEP
$#   ts      tip      its      tmin      tmax      dtemp      tscp      lcts
      0      1.000000      1.0e-5
*CONTROL_THERMAL_NONLINEAR
$#   refmax      tol
      10      1.0e-4
```

As shown above, *CONTROL_THERMAL_FORMING simplifies the user input by setting default values to less-frequently-used parameters implicitly. However, the user can always use a thermal control keyword to overwrite the default values of these hidden parameters. For instance, *CONTROL_THERMAL_FORMING sets FWORK (Fraction of mechanical Work converted to heat) to 10^{-10} implicitly. If FWORK needs to be increased to 0.8, add the following to the input deck to override the default value:

```
*CONTROL_THERMAL_SOLVER
$#    atype      ptype      solver      cgtol      gpt      eqheat      fwork      sbc
      1          1          11           1          0     1.000000      0.8     5.67e-11
```

The thick shell option from Card 2 of *CONTROL_SHELL can also be turned on with just *CONTROL_THERMAL_FORMING with the THSHEL field as follows:

```
*CONTROL_THERMAL_FORMING
$#    its      ptype      tsf      THSHEL      ITHOFF      solver      fwork
      1.0e-5      1          10.          1           1           11
```

The ITHOFF field for this keyword changes the offset for thermal contact for shells which normally requires the inclusion of optional Card 5 in *CONTROL_CONTACT.

```
*CONTROL_THERMAL_FORMING
$#    its      ptype      tsf      TSHEL      ITHOFF      solver      fwork
      1.0e-5      1          10.          1           1           11
```

A standard stamping simulation usually has multiple blank/die contact pairs with uniform or similar thermal/friction properties at the contact interface. With *CONTROL_THERMAL_FORMING a default set of thermal/friction parameters can be defined by including the optional Cards 2.1 and 2.2. Input values in Cards 2.1 and 2.2 will be automatically applied to thermal/friction parameters of all contact pairs defined in THRM 1 and THRM 2 of *CONTACT. Therefore, the user does not need to repeat input of these parameters for every contact pair. An example is shown below:

```
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ID
$#    cid                               title
      1 Die
$#    surfa      surf fb      surfatyp      surf btyp      saboxid      sbboxid      sapr      sbpr
      4          1          2          2
$#    fs         fd         dc         vc         vdc        penchk       bt         dt
      0.400000     0.000     0.000     0.000   20.000000          0     0.0001.0000E+20
$#    sf sa      sf sb      sast        sb st      sfsat      sfsbt      fsf        vsf
      0.000     0.000     0.000     0.000   1.000000     1.000000     1.000000     1.000000
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ID
$#    cid                               title
      2 Punch
$#    surfa      surf fb      surfatyp      surf btyp      saboxid      sbboxid      sapr      sbpr
      4          2          2          2
$#    fs         fd         dc         vc         vdc        penchk       bt         dt
      0.400000     0.000     0.000     0.000   20.000000          0     0.0001.0000E+20
$#    sf sa      sf sb      sast        sb st      sfsat      sfsbt      fsf        vsf
      0.000     0.000     0.000     0.000   1.000000     1.000000     1.000000     1.000000
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ID
$#    cid                               title
```

*CONTROL

*CONTROL_THERMAL_FORMING

```
      3 Binder
$#   surfa    surfb  surfatyp  surfbtyp  saboxid  sbboxid    sapr     sbpr
      4          3        2          2
$#   fs       fd      dc        vc        vdc      penchk     bt       dt
  0.400000  0.000   0.000   0.000  20.000000      0  0.0001.0000E+20
$#   sfsa    sfsb   sast      sbst      sfsat    sfsbt     fsf      vsf
  0.000   0.000   0.000   0.000  1.000000  1.000000  1.000000  1.000000
*CONTROL_THERMAL_FORMING
$   its     ptype     tsf     TSHEL    ITHOFF    solver    fwork
  1.0e-5      1
$#   k      frad      h0      lmin     lmax    ftosa    bc_flg    algo
  0.000   0.000  4.000000  0.050000  0.050000
$#   lcfst   lcfdt   formula      a        b        c        d      lch
  115
```

The above example is equivalent to the following input where the same set of thermal/friction parameters are defined three times, each corresponding to one single contact pair.

```
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_THERMAL_FRICTION_ID
$#   cid           title
      1 Die
$#   surfa    surfb  surfatyp  surfbtyp  saboxid  sbboxid    sapr     sbpr
      4          1        2          2
$#   fs       fd      dc        vc        vdc      penchk     bt       dt
  0.400000  0.000   0.000   0.000  20.000000      0  0.0001.0000E+20
$#   sfsa    sfsb   sast      sbst      sfsat    sfsbt     fsf      vsf
  0.000   0.000   0.000   0.000  1.000000  1.000000  1.000000  1.000000
$#   k      frad      h0      lmin     lmax    ftosa    bc_flg    algo
  0.000   0.000  4.000000  0.050000  0.050000
$#   lcfst   lcfdt   formula      a        b        c        d      lch
  115
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_THERMAL_FRICTION_ID
$#   cid           title
      2 Punch
$#   surfa    surfb  surfatyp  surfbtyp  saboxid  sbboxid    sapr     sbpr
      4          2        2          2
$#   fs       fd      dc        vc        vdc      penchk     bt       dt
  0.400000  0.000   0.000   0.000  20.000000      0  0.0001.0000E+20
$#   sfsa    sfsb   sast      sbst      sfsat    sfsbt     fsf      vsf
  0.000   0.000   0.000   0.000  1.000000  1.000000  1.000000  1.000000
$#   k      frad      h0      lmin     lmax    ftosa    bc_flg    algo
  0.000   0.000  4.000000  0.050000  0.050000
$#   lcfst   lcfdt   formula      a        b        c        d      lch
  115
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_THERMAL_FRICTION_ID
$#   cid           title
      3 Binder
$#   surfa    surfb  surfatyp  surfbtyp  saboxid  sbboxid    sapr     sbpr
      4          3        2          2
$#   fs       fd      dc        vc        vdc      penchk     bt       dt
  0.400000  0.000   0.000   0.000  20.000000      0  0.0001.0000E+20
$#   sfsa    sfsb   sast      sbst      sfsat    sfsbt     fsf      vsf
  0.000   0.000   0.000   0.000  1.000000  1.000000  1.000000  1.000000
$#   k      frad      h0      lmin     lmax    ftosa    bc_flg    algo
  0.000   0.000  4.000000  0.050000  0.050000
$#   lcfst   lcfdt   formula      a        b        c        d      lch
  115
*CONTROL_THERMAL_FORMING
$   its     ptype     tsf     TSHEL    ITHOFF    solver    fwork
  1.0e-5      1
```

Although Cards 2.1 and 2.2 of *CONTROL_THERMAL_FORMING provide a convenient way to define/update thermal/friction properties at multiple contact interfaces, some contact pairs require different values for the properties. In that case, simply add the THERMAL or THERMAL_FRICTION keyword option to this particular contact pair and input the thermal/friction parameters in the THRM 1 and THRM 2 cards which will override the default values set by *CONTROL_THERMAL_FORMING. The following examples shows how we use *CONTACT_THERMAL_FORMING to set HTC (Heat Transfer Coefficient) = 4.0 for all contact pairs except contact pair 2, which has HTC = 3.0.

```
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ID
$#      cid                               title
      1 Die
$#    surfa    surf fb   surfatyp   surf btyp   saboxid   sbboxid   sapr     sbpr
      4          1          2          2
$#    fs       fd       dc       vc       vdc      penchk     bt       dt
  0.400000  0.000    0.000    0.000  20.000000      0  0.0001.0000E+20
$#    sf sa   sf sb   sast     sb st   sfsat   sfsbt   fsf      vsf
  0.000    0.000    0.000    0.000  1.000000  1.000000  1.000000  1.000000
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_THERMAL_FRICTION_ID
$#      cid                               title
      2 Punch
$#    surfa    surf fb   surfatyp   surf btyp   saboxid   sbboxid   sapr     sbpr
      4          2          2          2
$#    fs       fd       dc       vc       vdc      penchk     bt       dt
  0.400000  0.000    0.000    0.000  20.000000      0  0.0001.0000E+20
$#    sf sa   sf sb   sast     sb st   sfsat   sfsbt   fsf      vsf
  0.000    0.000    0.000    0.000  1.000000  1.000000  1.000000  1.000000
$#    k       frad     h0       lmin     lmax     ftosa   bc_flg   algo
  0.000    0.000  3.000000  0.050000  0.050000
$#    lcfst   lcfdt   formula     a         b         c         d         lch
  115
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ID
$#      cid                               title
      3 Binder
$#    surfa    surf fb   surfatyp   surf btyp   saboxid   sbboxid   sapr     sbpr
      4          3          2          2
$#    fs       fd       dc       vc       vdc      penchk     bt       dt
  0.400000  0.000    0.000    0.000  20.000000      0  0.0001.0000E+20
$#    sf sa   sf sb   sast     sb st   sfsat   sfsbt   fsf      vsf
  0.000    0.000    0.000    0.000  1.000000  1.000000  1.000000  1.000000
*CONTROL_THERMAL_FORMING
$#    its     ptype     tsf     TSHEL   ITHOFF   solver   fwork
  1.0e-5      1
$#    k       frad     h0       lmin     lmax     ftosa   bc_flg   algo
  0.000    0.000  4.000000  0.050000  0.050000
$#    lcfst   lcfdt   formula     a         b         c         d         lch
  115
```

*CONTROL

*CONTROL_THERMAL_NONLINEAR

*CONTROL_THERMAL_NONLINEAR

Purpose: Set parameters for a nonlinear thermal or coupled structural/thermal analysis. The control card, *CONTROL_SOLUTION, is also required.

Card 1	1	2	3	4	5	6	7	8
Variable	REFMAX	TOL	DCP	LUMPBC	THLSTL	NLTHPR	PHCHPN	
Type	I	F	F	I	F	I	F	
Default	10	10^{-4}	1.0 or 0.5	0	0.	0	100.	

VARIABLE	DESCRIPTION
REFMAX	Maximum number of matrix reformations per time step. EQ.0: Set to 10 reformations.
TOL	Convergence tolerance for temperature. EQ.0.0: Set to $1000 \times$ machine roundoff.
DCP	Divergence control parameter: steady state problems $0.3 \leq DCP \leq 1.0$ default 1.0 transient problems $0.0 < DCP \leq 1.0$ default 0.5
LUMPBC	Lump thermal boundary conditions. LUMPBC = 1 activates a numerical method to damp out anomalous temperature oscillations resulting from very large step function boundary conditions. This option is not generally recommended. EQ.0: Off (default) EQ.1: On
THLSTL	Line search convergence tolerance: EQ.0.0: No line search GT.0.0: Line search convergence tolerance
NLTHPR	Thermal nonlinear print out level: EQ.0: No print out

VARIABLE	DESCRIPTION
	EQ.1: Print convergence parameters during solution of nonlinear system.
PHCHPN	Phase change penalty parameter: EQ.0.0: Set to default value 100. GT.0.0: Penalty to enforce constant phase change temperature

*CONTROL

*CONTROL_THERMAL_SOLVER

*CONTROL_THERMAL_SOLVER

Purpose: Set options for the thermal solution in a thermal only or coupled structural-thermal analysis. The control card, *CONTROL_SOLUTION, is also required.

Card Summary:

Card 1. This card is required.

ATYPE	PTYPE	SOLVER		GPT	EQHEAT	FWORK	SBC
-------	-------	--------	--	-----	--------	-------	-----

Card 2a. This card is optional. It is only read when SOLVER \neq 17.

MSG_LVL	MAX_ITR	ABSTOL	RELTOL	OMEGA			TSF
---------	---------	--------	--------	-------	--	--	-----

Card 2b. This card is optional. It is only read when SOLVER = 17.

MSG_LVL	NINNER	ABSTOL	RELTOL	NOUTER			
---------	--------	--------	--------	--------	--	--	--

Card 3. This card is optional.

MXDMP	DTVF	VARDEN		NCYCL			
-------	------	--------	--	-------	--	--	--

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	ATYPE	PTYPE	SOLVER		GPT	EQHEAT	FWORK	SBC
Type	I	I	I		I	F	F	F
Default	0	0	↓		8	1.	1.	0.
Remark				11				

VARIABLE

DESCRIPTION

ATYPE

Thermal analysis type:

EQ.0: Steady state analysis (see [Remark 4](#))

EQ.1: Transient analysis (see [Remark 3](#))

VARIABLE	DESCRIPTION
PTYPE	Thermal problem type: (see *CONTROL_THERMAL_NONLINEAR if nonzero) EQ.0: Linear problem, EQ.1: Nonlinear problem with material properties evaluated at gauss point temperature. EQ.2: Nonlinear problem with material properties evaluated at element average temperature.
SOLVER	Thermal analysis solver type (see Remarks 1 and 2): EQ.11: Direct solver EQ.12: Diagonal scaling (default for MPP) conjugate gradient iterative EQ.13: Symmetric Gauss-Seidel conjugate gradient iterative EQ.14: SSOR conjugate gradient iterative EQ.15: ILDLT0 (incomplete factorization) conjugate gradient iterative EQ.16: Modified ILDLT0 (incomplete factorization) conjugate gradient iterative EQ.17: GMRES solver for conjugate heat transfer problems EQ.18: ILDLT(T) (incomplete factorization with threshold pivoting) EQ.19: Preconditioned conjugate gradient with MUMPS (see Remark 2 in *CONTROL_IMPLICIT_SOLVER) EQ.30: Direct nonsymmetric factorization EQ.90: User-supplied linear equation (see Remark 3 of *CONTROL_IMPLICIT_SOLVER).
GPT	Number of Gauss points to be used in the solid elements: EQ.0: Use default value 8, EQ.1: One point quadrature is used.
EQHEAT	Mechanical equivalent of heat (see Remark 10). EQ.0.0: Default value 1.0, LT.0.0: Designates a load curve number for EQHEAT as a function of time.

*CONTROL

*CONTROL_THERMAL_SOLVER

VARIABLE	DESCRIPTION
FWORK	Fraction of mechanical work converted into heat. EQ.0.0: Use default value 1.0.
SBC	Stefan Boltzmann constant. Value is used with enclosure radiation surfaces; see *BOUNDARY_RADIATION_... LT.0.0: Use a smoothing algorithm when calculating view factors to force the row sum = 1.

Card 2a	1	2	3	4	5	6	7	8
Variable	MSG_LVL	MAX_ITR	ABSTOL	RELTOL	OMEGA			TSF
Type	I	I	F	F	F			F
Default	0	500	10^{-10}	10^{-6}	1.0 or 0.			1.

VARIABLE	DESCRIPTION
MSG_LVL	Output message level (For SOLVER > 10) EQ.0: No output (default), EQ.1: Summary information, EQ.2: Detailed information, use only for debugging.
MAXITR	Maximum number of iterations. For SOLVER > 11. EQ.0: Use default value 500,
ABSTOL	Absolute convergence tolerance. For SOLVER > 11. EQ.0.0: Use default value 10^{-10}
RELTOL	Relative convergence tolerance. For SOLVER > 11. EQ.0.0: Use default value 10^{-6}
OMEGA	Relaxation parameter omega for SOLVER 14 and 16. EQ.0.0: Use default value 1.0 for Solver 14, use default value 0.0 for Solver 16.

VARIABLE	DESCRIPTION							
TSF	Thermal Speedup Factor. This factor multiplies all thermal parameters with units of time in the denominator (e.g., thermal conductivity, convection heat transfer coefficients). It is used to artificially time scale the problem.							
	EQ.0.0: Default value 1.0,							
	LT.0.0: TSF is a load curve ID. Curve defines speedup factor as a function of time.							
	Its main use is in metal stamping. If the velocity of the stamping punch is artificially increased by 1000, then set TSF = 1000 to scale the thermal parameters.							

Card 2b	1	2	3	4	5	6	7	8
Variable	MSG_LVL	NINNER	ABSTOL	RELTOL	NOUTER			
Type	I	I	F	F	I			
Default	0	100	10^{-10}	10^{-6}	100			

VARIABLE	DESCRIPTION
MSG_LVL	Output message level (For SOLVER > 10) EQ.0: No output (default), EQ.1: Summary information, EQ.2: Detailed information, use only for debugging.
NINNER	Number of inner iterations for GMRES solve
ABSTOL	Absolute convergence tolerance. For SOLVER > 11. EQ.0.0: Use default value 10^{-10}
RELTOL	Relative convergence tolerance. For SOLVER > 11. EQ.0.0: Use default value 10^{-6}
NOUTER	Number of outer iterations for GMRES solve

CONTROL**CONTROL_THERMAL_SOLVER**

Card 3	1	2	3	4	5	6	7	8
Variable	MXDMP	DTVF	VARDEN		NCYCL			
Type	I	F	I		I			
Default	0	0.	0		1			.

VARIABLE	DESCRIPTION
MXDMP	Matrix Dumping for SOLVER > 11 EQ.0: No Dumping GT.0: Dump using ASCII format every MXDMP time steps. LT.0: Dump using binary format every MXDMP time steps.
DTVF	Time interval between view factor updates.
VARDEN	Variable thermal density for solid elements in a coupled thermal-structural analysis. Setting VARDEN to 1 or 2 will adjust the material density in the thermal solver to account for changes in element volume due to, for example, material compaction, thermal expansion, etc. In applications where volume changes are small, the default is recommended. EQ.0: Thermal density remains constant and equal to TRO as given in *MAT_THERMAL_OPTION (default). EQ.1: Thermal density varies to account for change in volume. If an equation of state (*EOS) is used, the initial internal energy specified therein is taken into account. EQ.2: Thermal density varies to account for change in volume. The initial internal energy is not considered.
NCYCL	Thermal matrix reassembly frequency. Default is at every thermal cycle.

Remarks:

1. **Recommended direct solver.** Solver 11 is the preferred direct solver. Solver type 30 is a nonsymmetric direct solver.

2. **Direct versus iterative solver.** Even though using a direct solver is usually less efficient than using an iterative solver (SOLVER = 12, 13, 14, 15, 16, 17, 18 and 19), we recommend it for thermal coupling analysis. If you are running a thermal only analysis, then consider using an iterative solver to decrease execution time (particularly for large models); otherwise we recommend a direct solver.
3. **Transient problems.** For transient problems, diagonal scaling conjugate gradient (SOLVER = 12) should be adequate.
4. **Steady state problems.** For steady-state problems, convergence may be slow or unacceptable, so consider using direct solver (SOLVER = 11) or a more powerful preconditioner (SOLVER = 13, 14, 15, 16, 18, and 19).
5. **Solvers 13 & 14.** Solver 13 (symmetric Gauss-Seidel) and solver 14 (SSOR) are related. When OMEGA = 1.0, solver 14 is equivalent to solver 13. The optimal omega value for SSOR is problem dependent but lies between 1.0 and 2.0.
6. **Solvers 15 & 16.** Solver 15 (incomplete LDLT0) and solver 16 (modified incomplete LDLT0) are related. Both are no-fill factorizations that require one extra n-vector of storage. The sparsity pattern of the preconditioner is exactly the same as that of the thermal stiffness matrix. Solver 16 uses the relaxation parameter OMEGA. The optimal OMEGA value is problem dependent, but lies between 0.0 and 1.0.
7. **Solver 17.** The GMRES solver has been developed as an alternative to the direct solvers in cases where the structural thermal problem is coupled with the fluid thermal problem in a monolithic approach using the ICFD solver. A significant gain of calculation time can be observed when the problem reaches 1M elements.
8. **Completion conditions for solvers 12 – 15.** Solvers 12, 13, 14, 15 and 16 terminate the iterative solution process when (1) the number of iterations exceeds MAXITR or (2) the 2-norm of the residual drops below:
$$\text{ABSTOL} + \text{RELTOL} \times \text{2-norm of the initial residual}$$
9. **Debug data.** Solvers 11 and up can dump the thermal conductance matrix and right-hand-side using the same formats as documented under *CONTROL_IMPLICIT_SOLVER. If this option is used, files beginning with T_ will be generated.
10. **Unit conversion factor.** EQHEAT is a unit conversion factor. EQHEAT converts the mechanical unit for work into the thermal unit for energy according to,

$$\text{EQHEAT} \times [\text{work}] = [\text{thermal energy}]$$

However, it is recommended that a consistent set of units be used with EQHEAT set to 1.0. For example, when using SI,

$$[\text{work}] = 1\text{Nm} = [\text{thermal energy}] = 1\text{J} \Rightarrow \text{EQHEAT} = 1.$$

11. **Obsolete variable CGTOL.** Starting with LS-DYNA R12, the variable CGTOL is obsolete; use RELTOL for non-default values of the relative convergence tolerance.

CONTROL_THERMAL_TIMESTEP**CONTROL*****CONTROL_THERMAL_TIMESTEP**

Purpose: Set time step controls for the thermal solution in a thermal only or coupled structural/thermal analysis. This card requires that the deck also include *CONTROL_SOLUTION, and, *CONTROL_THERMAL_SOLVER needed.

Card 1	1	2	3	4	5	6	7	8
Variable	TS	TIP	ITS	TMIN	TMAX	DTEMP	TSCP	LCTS
Type	I	F	F	F	F	F	F	I
Default	0	1.0	none	↓	↓	1.0	0.5	0

VARIABLE	DESCRIPTION
TS	Time step control: EQ.0: Fixed time step, EQ.1: Variable time step (may increase or decrease).
TIP	Time integration parameter. See Remark 1 for details. EQ.0.5: Crank-Nicolson scheme EQ.1.0: Fully implicit (default)
ITS	Initial thermal time step
TMIN	Minimum thermal time step: EQ.0.0: Set to structural explicit time step (default). LT.0.0: Curve ID = (-TMIN) gives minimum thermal time step size as a function of time. The load curve defines pairs (thermal time breakpoint, new minimum time step). See Remark 2 .
TMAX	Maximum thermal time step: EQ.0.0: Set to $100 \times$ structural explicit time step (default). LT.0.0: Curve ID = (-TMAX) gives maximum thermal time step size as a function of time. The load curve defines pairs (thermal time breakpoint, new minimum time step).

CONTROL**CONTROL_THERMAL_TIMESTEP**

VARIABLE	DESCRIPTION
	The time step will be adjusted to hit the time breakpoints exactly. See Remark 2 .
DTEMP	Maximum temperature change in each time step above which the thermal time step will be decreased: EQ.0.0: Set to a temperature change of 1.0. LT.0.0: Curve ID = (-DTEMP) gives maximum temperature change as a function of time. The load curve defines pairs (thermal time breakpoint, new temperature change). See Remark 2 .
TSCP	Time step control parameter ($0 < \text{TSCP} < 1.0$). The thermal time step is decreased by this factor, if the temperature change in the time step exceeds DTEMP or if the nonlinear solver fails to reach convergence within the given maximum number of matrix reformations. If the nonlinear solution algorithm is actually diverging, the thermal time step is decreased by DCP defined in *CONTROL_THERMAL_NONLINEAR. EQ.0.0: Set to a factor of 0.5.
LCTS	Load curve ID which defines data pairs of (thermal time breakpoint, new time step). The time step will be adjusted to hit the time breakpoints exactly. After the time breakpoint, the time step will be set to the 'new time step' ordinate value in the load curve.

Remarks:

1. **Time Integration Parameter.** TIP is relevant only to a transient analysis and is ignored for a steady state analysis. The only two accepted values that make mathematical sense are TIP = 0.5 or TIP = 1. TIP = 0.5 is the classical Crank-Nicolson method. It is mathematically more accurate than TIP = 1 and is unconditionally stable in time. However, a plot of nodal temperature as a function of time can be oscillatory, especially in a nonlinear problem. The oscillations decrease with time. TIP = 1.0 is the classical implicit method. It is also unconditionally stable. For this method, the plot of nodal temperature as a function of time does not have oscillations.
2. **Time Step Size Control.** If automatic time step size control is used with load curve definitions of the limits (dtmin, dtmax, dtemp), the time step is adjusted

to hit the breakpoints in the curves exactly. The time stepping algorithm ensures that the step size of the last time step before the breakpoint is not smaller than 50% of the previous one. If necessary, the sizes of the last two time steps before the breakpoint are averaged.

*CONTROL

*CONTROL_TIMESTEP

*CONTROL_TIMESTEP

Purpose: Set structural time step size control using different options.

Card 1	1	2	3	4	5	6	7	8
Variable	DTINIT	TSSFAC	ISDO	TSLIMT	DT2MS	LCTM	ERODE	MS1ST
Type	F	F	I	F	F	I	I	I
Default	↓	↓	0	0.0	0.0	0	0	0

This card is optional.

Card 2	1	2	3	4	5	6	7	8
Variable	DT2MSF	DT2MSLC	IMSCL			RMSCL	EMSCL	IHDO
Type	F	I	I			F	F	I
Default	not used	not used	0			0.0	0.0	0

This card is optional.

Card 3	1	2	3	4	5	6	7	8
Variable		IGADO	DTUSR	DTDYNV				
Type		I	F	I				
Default		0	0.0	0				

VARIABLE

DESCRIPTION

DTINIT

Initial time step size:

EQ.0.0: LS-DYNA determines initial step size.

VARIABLE	DESCRIPTION
TSSFAC	<p>Scale factor for computed time step (old name SCFT). See Remark 2 below.</p> <p>LT.0.0: $TSSFAC$ is the load curve or function ID defining the scale factor as a function of time.</p>
ISDO	<p>Basis of time size calculation for 4-node shell elements. 3-node shells use the shortest altitude for options 0,1 and the shortest side for option 2. This option has no relevance to solid elements, which use a length based on the element volume divided by the largest surface area.</p> <p>EQ.0: Characteristic length is given by</p> $\frac{\text{area}}{\min(\text{longest side}, \text{longest diagonal})}.$ <p>EQ.1: Characteristic length is given by</p> $\frac{\text{area}}{\text{longest diagonal}}.$ <p>EQ.2: Based on bar wave speed and,</p> $\max \left[\text{shortest side}, \frac{\text{area}}{\min(\text{longest side}, \text{longest diagonal})} \right].$ <div style="border: 1px solid black; padding: 10px; margin-top: 10px;"> <p>WARNING: Option 2 can give a much larger time step size that can lead to instabilities in some applications, especially when triangular elements are used.</p> </div> <p>EQ.3: This feature is currently unavailable. Time step size is based on the maximum eigenvalue. This option is okay for structural applications where the material sound speed changes slowly. The cost related to determining the maximum eigenvalue is significant, but the increase in the time step size often allows for significantly shorter run times without using mass scaling.</p>
TSLIMT	<p>Shell element minimum time step assignment, TSLIMT. When a shell controls the time step, element material properties (moduli <i>not</i> masses) will be modified such that the time step does not fall below the assigned step size. This option is applicable only to shell elements using material models:</p> <p style="padding-left: 40px;">*MAT_PLASTIC_KINEMATIC,</p>

*CONTROL

*CONTROL_TIMESTEP

VARIABLE	DESCRIPTION
	<p>*MAT_POWER_LAW_PLASTICITY, *MAT_STRAIN_RATE_DEPENDENT_PLASTICITY, *MAT_PIECE-WISE_LINEAR_PLASTICITY.</p>
	<p>WARNING: This so-called stiffness scaling option is NOT recommended. The DT2MS option below applies to all materials and element classes and is preferred.</p>
	<p>If both TSLIMT and DT2MS below are active and if TSLIMT is input as a positive number, then TSLIMT defaults to 10^{-18}, thereby disabling it.</p>
	<p>If TSLIMT is negative and less than $DT2MS$, then $TSLIMT$ is applied prior to the mass being scaled. If $DT2MS$ exceeds the magnitude of TSLIMT, then TSLIMT is set to 10^{-18}.</p>
DT2MS	<p>Time step size for mass scaled solutions. (Default = 0.0)</p> <p>GT.0.0: Positive values are for quasi-static analyses or time history analyses where the inertial effects are insignificant.</p> <p>LT.0.0: $TSSFAC \times DT2MS$ is the minimum time step size permitted and mass scaling is done if and only if it is necessary to meet the Courant time step size criterion. This option can be used in transient analyses if the mass increases remain insignificant. See also the variable MS1ST below and the *CONTROL_TERMINATION variable ENDMAS.</p>
	<p>WARNING: Superelements from, *ELEMENT_DIRECT_MATRIX_INPUT, are not mass scaled; consequently, DT2MS does not affect their time step size. In this case an error termination will occur, and DT2MS will need to be reset to a smaller value.</p>
LCTM	<p>Load curve ID that limits the maximum time step size (optional). This load curve defines the maximum time step size permitted as a function of time. If the solution time exceeds the final time value defined by the curve, the computed step size is used. If the time</p>

VARIABLE	DESCRIPTION
	step size from the load curve is exactly zero, the computed time step size is also used.
ERODE	Erosion flag for elements with small time step. See Remark 6 . EQ.0: Calculation will terminate if the solution time step drops to tsmin (see *CONTROL_TERMINATION). EQ.1: Solid elements or thick shell elements that cause the time step to drop to tsmin will erode; similarly, SPH particles that cause the time step to drop will be deactivated. EQ.10: Shell elements with time step below tsmin will erode. EQ.11: Same as ERODE = 1 but shell elements will also erode EQ.100: Beam elements with time step below tsmin will erode. EQ.101: Same as ERODE = 1 but beam elements will also erode EQ.110: Beam and shell elements will erode. EQ.111: Same as ERODE = 1 but beam and shell elements will also erode
MS1ST	Option for mass scaling that applies when DT2MS < 0. EQ.0: Mass scaling is considered throughout the analysis to ensure that the minimum time step cannot drop below TSSFAC × DT2MS . Added mass may increase with time, but it will never decrease. (default) EQ.1: Added mass is calculated at the first time step and remains unchanged thereafter. The initial time step will not be less than TSSFAC × DT2MS , but the time step may subsequently decrease, depending on how the mesh deforms and the element characteristic lengths change.
DT2MSF	Reduction (or scale) factor for initial time step size to determine the minimum time step size permitted. Mass scaling is done if it is necessary to meet the Courant time step size criterion. If this option is used, DT2MS effectively becomes -DT2MSF multiplied by the initial time step size, Δt , before scaling Δt by TSSFAC. This option is active only if DT2MS = 0 above.
DT2MSLC	Load curve for determining the magnitude of DT2MS as a function of time, $f_{DT2MS}(t)$, during the explicit solutions phase. Time zero must be in the abscissa range of this curve. At a given simulation

*CONTROL

*CONTROL_TIMESTEP

VARIABLE	DESCRIPTION
	time $t, f_{DT2MS}(t) \times \text{sign(DT2MS)}$ plays the role of DT2MS according to the description for DT2MS above. All negative ordinate values may be used in the curve, but the ordinate values <i>may not</i> change sign during the simulation. In the negative ordinate values case, $f_{DT2MS}(t)$ itself (sign and magnitude) determines how mass scaling is performed and DT2MS is neglected.
	GT.0: The magnitude of the added mass is the maximum of the current value and the value from the load curve. Hence, the amount of added mass will never decrease, although the time step size may be reduced as desired using DT2MSLC. LT.0: Uses load curve $ DT2MSLC $. Allows the amount of added mass to decrease with time.
IMSCL	Flag for selective mass scaling if and only if mass scaling active. Selective mass scaling does not scale the rigid body mass and is therefore more accurate. Since it is memory and CPU intensive, it should be applied only to small finely meshed parts. EQ.0: No selective mass scaling EQ.1: All parts undergo selective mass scaling. LT.0: Recommended; $ IMSCL $ is the part set ID of the parts that undergo selective mass scaling. All other parts are mass scaled the usual way.
RMSCL	Flag for using rotational option in selective mass scaling EQ.0.: Only translational inertia are selectively mass scaled. NE.0.: Both translational and rotational inertia are selectively mass scaled.
EMSCL	Fraction of added mass from mass scaling that contributes to gravity loads, in addition to the physical mass. See also *LOAD_BODY. This number should be between 0 and 1.
IHDO	Method for calculating solid element time steps (see Remark 7): EQ.0: Default method EQ.1: Modified method to improve time step continuity
IGADO	Method for calculating time steps for IGA elements: EQ.0: Default method (conservative)

VARIABLE	DESCRIPTION
	EQ.1: Account for interelement continuity (usually leads to larger time steps)
DTUSR	User-defined time step for explicit analysis. A nonzero value invokes a call to user subroutine <code>utimestep</code> in <code>dyn21.F</code> . This feature can be used to synchronize time steps between coupled codes.
DTDYNV	Flag to consider the effects of dynamic viscosity on the critical time step. Like bulk viscosity, dynamic viscosity affects the critical time step. This flag applies only to solid elements with <code>*MAT_NULL</code> . <ul style="list-style-type: none"> EQ.0: Do not account for the effect of dynamic viscosity. This setting can lead to instabilities. EQ.1: Account for the effect of dynamic viscosity

Remarks:

1. **Time Step Scale Factor TSSFAC.** During the solution we loop through the elements and determine a new time step size by taking the minimum value over all elements:

$$\Delta t^{n+1} = \text{TSSFAC} \times \min\{\Delta t_1, \Delta t_2, \dots, \Delta t_N\}$$

where N is the number of elements. The time step size roughly corresponds to the transient time of an acoustic wave through an element using the shortest characteristic distance. For stability reasons the scale factor TSSFAC should be set to a value less than 1.0. The default value of TSSFAC is as follows:

- a) If contacts are present, and SLSFAC is specified in CONTROL_CONTACT, then

$$\text{TSSFAC} = \begin{cases} 0.333 & \text{for } \text{SLSFAC} \geq 9. \\ 0.667 & \text{for } 1. \leq \text{SLSFAC} < 9. \end{cases}$$

- b) TSSFAC = 0.667 if either of the following cards is used:

`*INITIAL_DETONATION`

`*BOUNDARY_NON_REFLECTING`

- c) TSSFAC = 0.90 otherwise

Values larger than .90 will often lead to instabilities

2. **Sound Speed and Element Size.** The sound speed in steel and aluminum is approximately 5 mm per microsecond; therefore, if a steel structure is modeled

with element sizes of 5 mm, the computed time step size would be 1 microsecond. Elements made from materials with lower sound speeds, such as foams, will give larger time step sizes. Avoid excessively small elements and be aware of the effect of rotational inertia on the time step size in the Belytschko beam element. Sound speeds differ for each material, for example, consider:

Air	331 m/s
Water	1478
Steel	5240
Titanium	5220
Plexiglass	2598

3. **Use Rigid Bodies when Possible.** It is recommended that stiff components be modeled by using rigid bodies. Do not scale the Young's modulus, as that can substantially reduce the time step size.
4. **Triangular Elements.** The altitude of the triangular element should be used to compute the time step size. Using the shortest side is okay only if the calculation is closely examined for possible instabilities. This is controlled by parameter ISDO.
5. **Selective Mass Scaling.** In the explicit time integration context and in contrast to conventional mass scaling, selective mass scaling (SMS) is a well thought out scheme that not only reduces the number of simulation cycles but that also does not significantly affect the dynamic response of the system under consideration. The drawback is that a linear system of equations must be solved in each time step for the accelerations. In this implementation a preconditioned conjugate gradient method (PCG) is used.

An unfortunate consequence of this choice of solver is that the efficiency will worsen when attempting large time steps since the condition number of the assembled mass matrix increases with the added mass. Therefore, caution should be taken when choosing the desired time step size. For large models it is also recommended to only use SMS on critical parts since it is otherwise likely to slow down execution; the bottleneck being the solution step for the system of linear system of equations.

While some constraints and boundary conditions available in LS-DYNA are not supported for SMS, they can be implemented upon request from a user.

A partial list of constraints and boundary conditions supported with SMS:

- Pointwise nodal constraints in global and local directions
- Prescribed motion in global and local directions
- Adaptivity
- Rigid walls

- Deformable elements merged with rigid bodies
- Constraint contacts and spotwelds
- Beam release constraints

By default, only the translational dynamic properties are treated. This means that only rigid body translation will be unaffected by the mass scaling imposed. There is an option to also properly treat rigid body rotation in this way, this is invoked by flagging the parameter RMSCL. A penalty in computational expense is incurred, but the results could be improved if rotations are dominating the simulation.

Selective mass scaling is supported for most element formulations. For beam elements, only types 1 (Hughes-Liu) and 9 (Spotweld) are supported at the moment.

6. **ERODE.** If DTMIN > 0 on CONTROL_TERMINATION, then tsmin is calculated by $tsmin = dtstart \times DTMIN$ where dtstart is the initial step size determined by LS-DYNA.

If $tsmin > 0$ and $ERODE > 0$, and if the solution time step drops below $tsmin$, then those elements that cause the time step to drop will be deleted so that the solution can continue with a time step that stays at or above $tsmin$. Valid values for ERODE are 0, 1, 10, 11, 100, 101, 110, and 111. Each digit in ERODE switches on or off checking and deletion for a type of element where a one turns on checking and a zero turns it off. The one's place controls solid and thick shell elements as well as SPH particles. The tens place controls shell elements and the hundreds place controls beam elements. Set ERODE = 111 to check all element types.

In addition to maintaining a reasonable time step, setting ERODE = 1, 11, or 111 will also prevent error terminations due to negative volume of solid elements. However, the solid elements checked for negative volume can be limited to those contained in a part set by setting part set PSFAIL on *CONTROL_SOLID.

7. **IHDO.** The explicit time step calculation for solid elements is dependent not only on the highest element frequency but also on the volumetric strain rate and linear coefficient of bulk viscosity. The default method for including the viscous effects causes a discontinuity in the equation when the volumetric strain rate changes sign. As a result, some explicit dynamic solutions are observed to have a solution time step that fluctuates between two values that differ by a few percent. An alternative method which eliminates the discontinuity in the viscous effects is invoked by setting IHDO = 1.

*CONTROL

*CONTROL_UNITS

*CONTROL_UNITS

Purpose: Specify the user units for the current keyword input deck. This does not provide any mechanism for automatic conversion of units of any entry in the keyword input deck. It is intended to be used for several purposes, but currently only for the situation where an external database in another set of units will be loaded and used in the simulation. In this case, *CONTROL_UNITS provides the information necessary to convert the external data into internal units (see *CHEMISTRY_CONTROL for such external databases).

If the needed unit is not one of the predefined ones listed for use on the first card, then the second optional card is used to define that unit. Any non-zero scales that are entered on optional Card 2 override what is specified on the first card. These scales are given in terms of the default units on Card 1. For instance, if 3600.0 is given in the second 20 character field on the optional second card (TIME_SCALE), then 'hour' is the time unit (3600 seconds).

Card 1	1	2	3	4	5	6	7	8
Variable	LENGTH	TIME	MASS	TEMP				
Type	A	A	A	A				
Default	m	sec	kg	K				

User Defined Units Card. Optional card only used when a new unit needs to be defined.

Card 2	1	2	3	4	5	6	7	8
Variable	LENGTH_SCALE		TIME_SCALE		MASS_SCALE			
Type	F		F		F			
Default	1.0		1.0		1.0			

VARIABLE	DESCRIPTION
LENGTH	Length units: EQ.m: meter (default) EQ.mm: millimeter EQ.cm: centimeter EQ.in: inch EQ.ft: foot
TIME	Time units: EQ.sec: second (default) EQ.ms: millisecond EQ.micro_s: microsecond
MASS	Mass units: EQ.kg: kilogram (default) EQ.g: gram EQ.mg: milligram EQ.lb: pound EQ.slug: pound × second ² / foot EQ.slinch: pound × second ² / inch EQ.mtrc_ton: metric ton
TEMP	Temperature units: EQ.K: Kelvin (default) EQ.C: Celsius EQ.F: Fahrenheit EQ.R: Rankine
LENGTH_SCALE	Number of meters in the length unit for the input deck
TIME_SCALE	Number of seconds in the time unit for the input deck
MASS_SCALE	Number of kilograms in the mass unit for the input deck

*CONTROL

*CONTROL_2D_REMESHING_REGION

*CONTROL_2D_REMESHING_REGION

Purpose: Specify a region of a 2D mesh to remesh. The keyword should be used with ADPTYP = 8 on *CONTROL_ADAPTIVE.

Note that each definition of *CONTROL_2D_REMESHING_REGION specifies one region. To include more than one region, you need to include multiple *CONTROL_2D_REMESHING_REGION keywords in the input deck.

Include as many cards as necessary. This input ends at the next keyword ("**") card. Each line for LTYP = 1, 2, 3, 7, or 8 defines a region to remesh. If there are several of these lines, the intersection of these regions is remeshed.

Card 1	1	2	3	4	5	6	7	8
Variable	LTYP	PAR1	PAR2	PAR3	PAR4	PAR5	PAR6	PAR7
Type	I	F	F	F	F	F	F	F
Default	none	0.0	0.0	0.0	0.0	0.0	0.0	0.0

VARIABLE

DESCRIPTION

LTYP

Type of regions defined by the parameters PARi:

EQ.1: Box. PAR1 is the ID of *DEFINE_BOX that defines the region to remesh. The other parameters are not used.

EQ.2: Part set. PAR1 is the ID of *SET_PART that selects the parts to remesh. The other parameters are not used.

EQ.3: Region around elements in contact from a specified contact. To specify the desired contact, set PAR1 to its order of appearance in the input deck. For instance, PAR1 = 5 if the desired contact is the 5th contact keyword to appear in the input deck. A box is created around each element in contact for the remeshing. PAR2 through PAR4 add padding around the box to increase the remeshing region. PAR2 > 0 is the length subtracted from the x-coordinate of the box's lower corner. PAR3 > 0 is the length added to the x-coordinate of the box's upper corner. PAR4 > 0 is the length subtracted from the y-coordinate of the box's lower corner. PAR5 > 0 is the length added to the y-coordinate of the box's upper corner. The last 2 parameters are not used.

VARIABLE	DESCRIPTION
EQ.4:	PAR1 is the ID of *DEFINE_BOX that selects mesh boundaries (edges) of remeshing regions along which nodes added after remeshing (hanging nodes between edge ends) keep their initial parametric positions between the boundary corner nodes (edge-end constraining nodes). The other parameters are not used. By default, when nodes are added to the edges of elements in the region and the edges are not on the mesh boundary, the positions and velocities of the hanging nodes are interpolated from the positions and velocities of the original constraining nodes along these edges (constraining nodes are nodes that existed before remeshing). If the edge of a remeshed element is on the mesh boundaries, the positions and velocities of the hanging nodes on the edge are, by default, not interpolated because they are likely to be subject to boundary conditions. With LTYP = 4, the hanging nodes on the boundary of the mesh are interpolated.
EQ.5:	PAR1 is the ID of *SET_NODE that selects nodes along shell edges. After remeshing, the node set is recreated with nodes along the same shell edges. PAR2 > 0 is a thickness for the shell edges to select the new nodes after remeshing. The other parameters are not used.
EQ.6:	PAR1 is the ID of *SET_PART that selects the parts for which the total displacements are output to d3plot after remeshings.
EQ.7:	PAR1 is the ID of *SET_SHELL that selects the shells to remesh. The other parameters are not used.
EQ.8:	PAR1 is the ID of a 2D *SET_SEGMENT that selects the shells to remesh. A 2D *SET_SEGMENT is a list of segments with 2 nodes (the 2nd node is repeated for the 3rd and 4th node IDs). A 2D segment normal points to the right when going from the 1st to the 2nd node. A shell is in the region if nominally all the segment normal vectors point towards it (see Figure 12-90 for illustration). The other parameters are not used.
EQ.9:	PAR1 is the ID of a *SET_PART that selects the shell parts for which an area-weighted mapping of the element data is applied. The other parameters are not used. If PAR1 is 0, all parts are selected to be mapped.
EQ.10:	PAR1 is the ID of *DEFINE_CURVE that lists coordinate

*CONTROL

*CONTROL_2D_REMESHING_REGION

VARIABLE

DESCRIPTION

couples (x, y) to make a closed curve (the last point in this list is the same as the first one). The region of shells to remesh is on the right when moving along the curve from its 1st to last point (see [Figure 12-91](#)).

PAR*i*

Parameters defined by LTYP

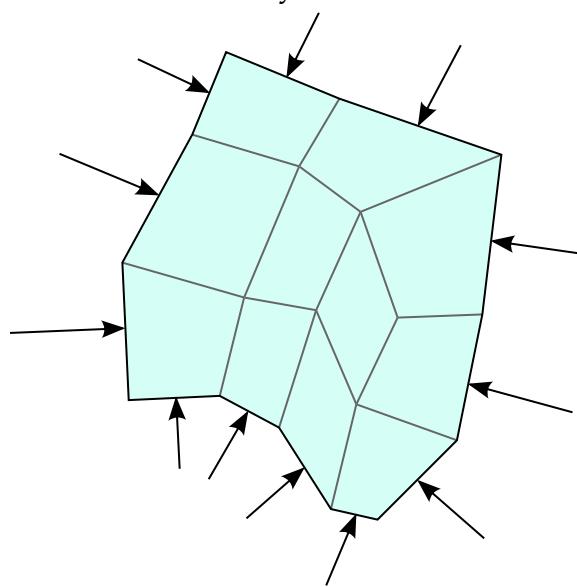


Figure 12-90. Example of a region specified with LTYP = 8. The arrows indicate the normal vectors for the segments.

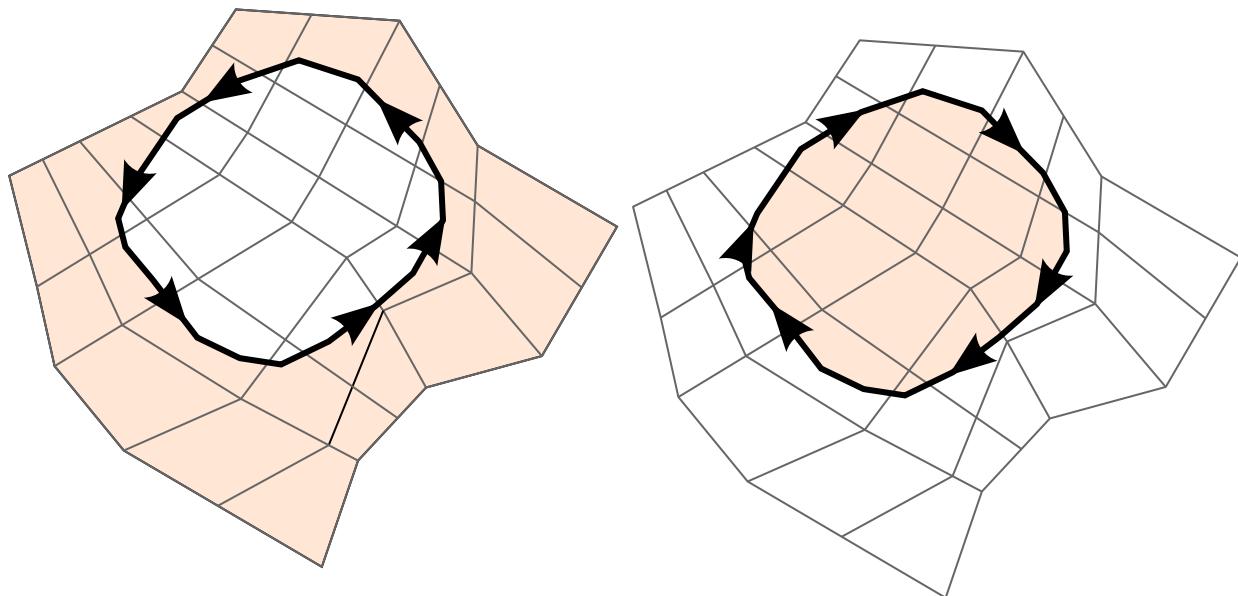


Figure 12-91. Example of regions specified with LTYP = 10. The arrows indicate the direction of the curve as input. The region depends on this direction.