

***DEFINE**

The keyword *DEFINE provides a way of defining boxes, coordinate systems, load curves, functions, tables, transformations, and orientation vectors. Some of the *DEFINE keywords also assist with particle methods, airbags, failure, metal forming, welds, generalized elements, and contact. The keyword cards in this section are defined in alphabetical order:

- *DEFINE_ADAPTIVE_SOLID_TO_DES
- *DEFINE_ADAPTIVE_SOLID_TO_SPH
- *DEFINE_BEAM_SOLID_COUPLING
- *DEFINE_BOX
- *DEFINE_BOX_ADAPTIVE
- *DEFINE_BOX_COARSEN
- *DEFINE_BOX_DRAWBEAD
- *DEFINE_BOX_NODES_ADAPTIVE
- *DEFINE_BOX_SPH
- *DEFINE_CABLE
- *DEFINE_CONNECTION_PROPERTIES
- *DEFINE_CONSTRUCTION_STAGES
- *DEFINE_CONTACT_EXCLUSION
- *DEFINE_CONTACT_VOLUME
- *DEFINE_CONTROL_VOLUME
- *DEFINE_CONTROL_VOLUME_FLOW_AREA
- *DEFINE_CONTROL_VOLUME_INTERACTION
- *DEFINE_COORDINATE_NODES
- *DEFINE_COORDINATE_SYSTEM
- *DEFINE_COORDINATE_VECTOR

*DEFINE

*DEFINE_CPG_GAS_PROPERTIES
*DEFINE_CPG_REGION
*DEFINE_CPM_BAG_INTERACTION
*DEFINE_CPM_CHAMBER
*DEFINE_CPM_GAS_PROPERTIES
*DEFINE_CPM_NPDATA
*DEFINE_CPM_SWITCH_REGION
*DEFINE_CPM_VENT
*DEFINE_CURVE
*DEFINE_CURVE_BOX_ADAPTIVITY
*DEFINE_CURVE_COMPENSATION_CONSTRAINT_OPTION
*DEFINE_CURVE_DRAWBEAD
*DEFINE_CURVE_DUPLICATE
*DEFINE_CURVE_ENTITY
*DEFINE_CURVE_FEEDBACK
*DEFINE_CURVE_FLC
*DEFINE_CURVE_FLD_FROM_TRIAXIAL_LIMIT
*DEFINE_CURVE_FUNCTION
*DEFINE_CURVE_SMOOTH
*DEFINE_CURVE_STRESS
*DEFINE_CURVE_TRIAXIAL_LIMIT_FROM_FLD
*DEFINE_CURVE_TRIM
*DEFINE_DE_ACTIVE_REGION
*DEFINE_DE_BOND
*DEFINE_DE_BOND_OVERRIDE
*DEFINE_DE_BY_PART

*DEFINE_DE_COHESIVE
*DEFINE_DE_FLOW_DRAG
*DEFINE_DE_HBOND
*DEFINE_DE_INJECT_BONDED
*DEFINE_DE_INJECT_SHAPE
*DEFINE_DE_INJECTION
*DEFINE_DE_INTERNAL_SKIP
*DEFINE_DE_MASSFLOW_PLANE
*DEFINE_DE_MESH_BEAM
*DEFINE_DE_MESH_SURFACE
*DEFINE_DE_MESH_VOLUME
*DEFINE_DE_PARTS_INTERACTION
*DEFINE_DE_PATTERN_OUTPUT
*DEFINE_DE_TEMP
*DEFINE_DE_TO_BEAM_COUPLING
*DEFINE_DE_TO_CPM_COUPLING
*DEFINE_DE_TO_SURFACE_COUPLING
*DEFINE_DE_TO_SURFACE_TIED
*DEFINE_DEATH_TIMES_OPTION
*DEFINE_DRIFT_REMOVE
*DEFINE_ELEMENT_DEATH_OPTION
*DEFINE_ELEMENT_EROSION_OPTION
*DEFINE_ELEMENT_GENERALIZED_SHELL
*DEFINE_ELEMENT_GENERALIZED_SOLID
*DEFINE_FABRIC_ASSEMBLIES
*DEFINE_FIBERS

*DEFINE

*DEFINE_FIELD
*DEFINE_FILTER
*DEFINE_FORMING_ADAPTIVE_PART_INITIAL
*DEFINE_FORMING_BLANKMESH
*DEFINE_FORMING_CLAMP
*DEFINE_FORMING_CONTACT
*DEFINE_FORMING_ONESTEP_PRIMARY
*DEFINE_FP_TO_SURFACE_COUPLING
*DEFINE_FRICTION
*DEFINE_FRICTION_ORIENTATION
*DEFINE_FRICTION_SCALING
*DEFINE_FUNCTION
*DEFINE_FUNCTION_TABULATED
*DEFINE_GROUND_MOTION
*DEFINE_HAZ_PROPERTIES
*DEFINE_HAZ_TAILOR_WELDED_BLANK
*DEFINE_HEX_SPOTWELD_ASSEMBLY
*DEFINE_ISPG_TO_SURFACE_COUPLING
*DEFINE_LANCE_SEED_POINT_COORDINATES
*DEFINE_MATERIAL_HISTORIES
*DEFINE_MULTI_DRAWBEADS_IGES
*DEFINE_MULTI_SHEET_CONNECTORS
*DEFINE_MULTISCALE
*DEFINE_NURBS_CURVE
*DEFINE_PART_FROM_LAYER
*DEFINE_PARTICLE_BLAST

*DEFINE_PBLAST_AIRGEO
*DEFINE_PBLAST_GEOMETRY
*DEFINE_PLANE
*DEFINE_POINT_CLOUD
*DEFINE_POROUS_OPTION
*DEFINE_PRESSURE_TUBE
*DEFINE_QUASAR_COUPLING
*DEFINE_REGION
*DEFINE_SALEAB_PARAMETERS
*DEFINE_SD_ORIENTATION
*DEFINE_SET_ADAPTIVE
*DEFINE_SPG_TO_SURFACE_COUPLING
*DEFINE_SPH_ACTIVE_REGION
*DEFINE_SPH_AMBIENT_DRAG
*DEFINE_SPH_DE_COUPLING
*DEFINE_SPH_INJECTION
*DEFINE_SPH_INJECTION_SIMPLIFIED
*DEFINE_SPH_MASSFLOW_PLANE
*DEFINE_SPH_MESH_BOX
*DEFINE_SPH_MESH_OBJ
*DEFINE_SPH_MESH_SURFACE
*DEFINE_SPH_TO_SPH_COUPLING
*DEFINE_SPH_VICINITY_SENSOR
*DEFINE_SPOTWELD_FAILURE
*DEFINE_SPOTWELD_FAILURE_RESULTANTS
*DEFINE_SPOTWELD_MULTISCALE

*DEFINE

```
*DEFINE_SPOTWELD_RUPTURE_PARAMETER  
*DEFINE_SPOTWELD_RUPTURE_STRESS  
*DEFINE_STAGED_CONSTRUCTION_PART  
*DEFINE_STOCHASTIC_ELEMENT_OPTION  
*DEFINE_STOCHASTIC_VARIATION  
*DEFINE_STOCHASTIC_VARIATION_PROPERTIES  
*DEFINE_TABLE  
*DEFINE_TABLE_2D  
*DEFINE_TABLE_3D  
*DEFINE_TABLE_{X}D  
*DEFINE_TABLE_MATRIX  
*DEFINE_TARGET_BOUNDARY  
*DEFINE_TRACER PARTICLES_2D  
*DEFINE_TRANSFORMATION  
*DEFINE_TRIM_SEED_POINT_COORDINATES  
*DEFINE_VECTOR  
*DEFINE_VECTOR_NODES
```

Unless noted otherwise, an additional keyword option TITLE may be appended to the *DEFINE keywords. If this option is used, then an addition line is read for each section in 80a format which can be used to describe the defined curve, table, etc. At present, the title serves no purpose other than to perhaps lend clarity to input decks.

Examples for the *DEFINE keyword can be found at the end of this section.

DEFINE_ADAPTIVE_SOLID_TO_DES**DEFINE*****DEFINE_ADAPTIVE_SOLID_TO_DES {OPTION}**

Purpose: Adaptively transform a Lagrangian solid part or part set to DES (Discrete Element Sphere) particles (elements) when the Lagrangian solid elements comprising those parts fail. One or more DES particles will be generated for each failed element as debris. The DES particles replacing the failed element inherit the properties of the failed solid element, including mass and kinematical state.

The available options include:

<BLANK>

ID

ID Card. Additional card for the ID keyword option.

Optional	1	2	3	4	5	6	7	8
Variable	DID			HEADING				
Type	I			A70				
Default	none			none				

Card 1	1	2	3	4	5	6	7	8
Variable	IPID	ITYPE	NQ	IPDES	ISDES	RSF	OUTDES	IBOND
Type	I	I	I	I	I	F	I	I
Default	none	none	none	none	none	1.0	0	0

DEFINE**DEFINE_ADAPTIVE_SOLID_TO_DES**

Bonds Card. This card is only included when IBOND = 1.

Card 2	1	2	3	4	5	6	7	8
Variable	PBN	PBS	PBN_S	PBS_S	SFA	ALPHA		
Type	F	F	F	F	F	F		
Default	none	none	none	none	1.0	0.0		

VARIABLE	DESCRIPTION
DID	Definition ID. This must be a unique number.
HEADING	Definition descriptor. It is suggested that unique descriptions be used.
IPID	ID of the solid part or part set to transform.
ITYPE	IPID type: EQ.0: Part ID NE.1: Part set ID
NQ	Adaptive option for hexahedral elements. For tetrahedral and pentahedral elements, see Remark 1 . EQ.1: Adapt one solid element to one discrete element EQ.2: Adapt one solid element to 8 discrete elements EQ.3: Adapt one solid element to 27 discrete elements
IPDES	Part ID for newly generated discrete elements. See Remark 2 .
ISDES	Section ID for discrete elements. See Remark 2 .
RSF	DES radius scale down factor, which is the ratio of the radius of the generated DES to the calculated radius based on volume consistency.
OUTDES	Allow user output generated discrete element nodes and DES properties to a keyword file. EQ.0: No output (default). EQ.1: Write data under filename desvfill.inc.

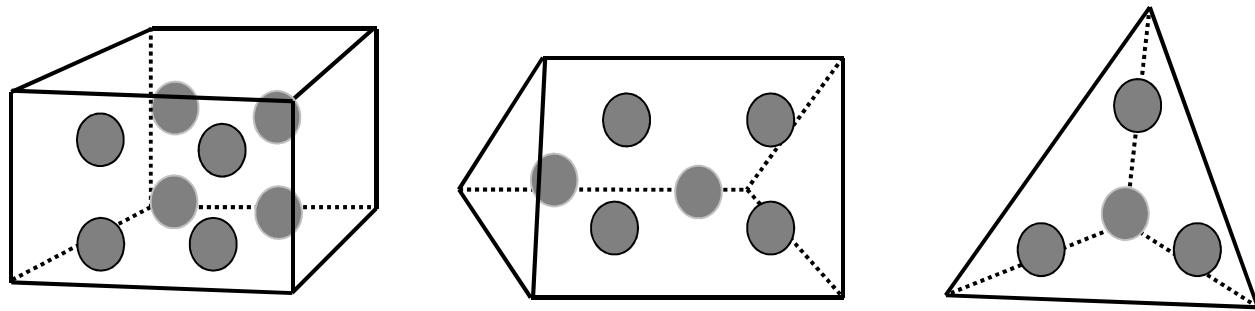


Figure 17-1. Left to right, illustration of conversion from solid to DES for NQ = 2 of hexahedron, pentahedron, and tetrahedron elements.

VARIABLE	DESCRIPTION
IBOND	Allow user define bonds between DES generated from the same solid element. EQ.0: No bonds (default). EQ.1: Bonds generated, need to define Card 2.
PBN	Parallel-bond modulus [Pa]. See Remark 3 .
PBS	Parallel-bond stiffness ratio. Shear stiffness/normal stiffness. See Remark 3 .
PBN_S	Parallel-bond maximum normal stress. A zero value defines an infinite maximum normal stress.
PBS_S	Parallel-bond maximum shear stress. A zero value defines an infinite maximum shear stress.
SFA	Bond radius multiplier. Default is 1.0.
ALPHA	Numerical damping

Remarks:

1. **DES Element to Solid Element Ratio.** The DES particles are evenly distributed within the solid element. For hexahedral elements the number of the generated DES particles is $NQ \times NQ \times NQ$. For pentahedral elements, the number of generated DES particles is 1, 6, and 18 for $NQ = 1, 2$, and 3, respectively. For tetrahedral elements, the number of generated DES particles is 1, 4, and 10 for $NQ = 1, 2$, and 3, respectively. See [Figure 17-1](#).
2. **Part ID.** The part ID for newly generated DES particles can be either a new part ID or the ID of an existing DES part.

3. **Bond Forces.** The normal force between two bonded discrete elements with radii r_1 and r_2 is calculated as

$$\Delta f_n = \frac{PBN}{(r_1 + r_2)} \times A \times \Delta u_n ,$$

where

$$A = \pi r_{\text{eff}}^2$$
$$r_{\text{eff}} = \min(r_1, r_2) \times SFA$$

The shear force is calculated as

$$\Delta f_s = PBS \times \frac{PBN}{(r_1 + r_2)} \times A \times \Delta u_s .$$

***DEFINE_ADAPTIVE_SOLID_TO_SPH {OPTION}**

Purpose: Create SPH particles to either replace or supplement solid Lagrangian elements.

Applications of this feature include adaptively transforming a Lagrangian solid Part or Part Set to SPH particles, when the Lagrangian solid elements comprising those parts fail. One or more SPH particles (elements) will be generated for each failed element. The SPH particles replacing the failed solid Lagrangian elements inherit all the Lagrange nodal quantities and all the Lagrange integration point quantities of these failed solid elements. Those properties are assigned to the newly activated SPH particles. The constitutive properties assigned to the new SPH part will correspond to the MID and EOSID referenced by the SPH *PART definition. This keyword with options of ICPL = 0, 1 has been extended to 2D cases too.

The available options include:

<BLANK>

ID

ID Card. Additional card for the ID keyword option.

Card ID	1	2	3	4	5	6	7	8
Variable	DID				HEADING			
Type	I				A70			
Default	none				none			

Card 1	1	2	3	4	5	6	7	8
Variable	IPID	ITYPE	NQ	IPSPH	ISSPH	ICPL	IOPT	CPCD
Type	I	I	I	I	I	I	I	F
Default	none	none	none	none	none	none	none	average

VARIABLE**DESCRIPTION**

DID

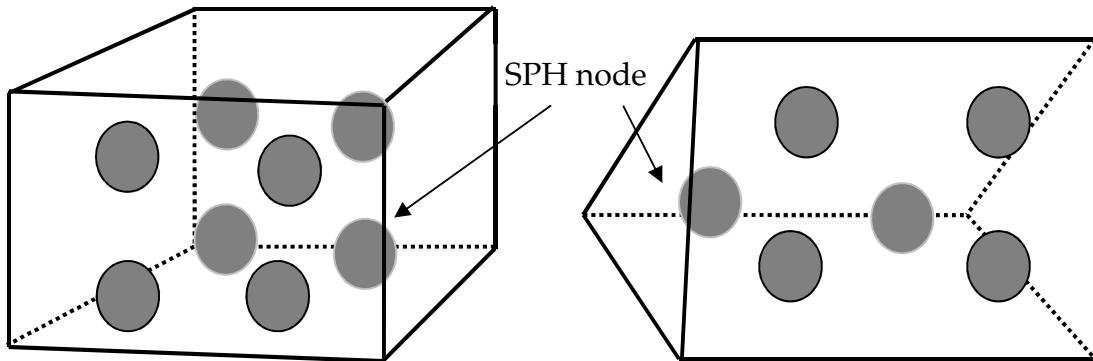
Definition ID. This must be a unique number.

DEFINE**DEFINE_ADAPTIVE_SOLID_TO_SPH**

VARIABLE	DESCRIPTION
HEADING	Definition descriptor. We suggest using unique descriptions.
IPID	ID of the solid part or part set to transform
ITYPE	IPID type: EQ.0: Part ID NE.0: Part set ID
NQ	Adaptive option for hexahedral elements. For tetrahedral and pentahedral elements, see Remark 1 : EQ.n: Adapt one 8-node solid element to $(n \times n \times n)$ SPH elements. The range of n is from 1 to 6.
IPSPH	Part ID for newly generated SPH elements. See Remark 2 .
ISSPH	Section ID for SPH elements. See Remark 2 .
ICPL	Coupling of newly generated SPH elements to the adjacent solid elements: EQ.0: Failure without coupling (debris simulation), EQ.1: Coupled to solid element, EQ.3: Provide only thermal coupling between SPH part and solid part (must be combined with IOPT = 0 option; see Remark 4).
IOPT	Coupling method: EQ.0: Coupling from beginning (used as constraint between SPH elements and solid elements), EQ.1: Coupling begins when Lagrangian solid element fails. See Remark 3 .
CPCD	Thermal coupling conductivity between SPH part and solid part for ICPL = 3 option. The default value is set as the average value of the conductivity from SPH part and the conductivity from solid part.

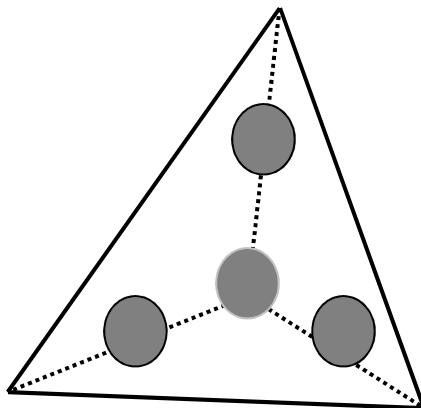
Remarks:

1. **Number of SPH Particles per Element.** The SPH particles are evenly distributed within the solid element. For hexahedral elements the number of the generated SPH particles is $NQ \times NQ \times NQ$. For pentahedral elements, the number



Example of SPH nodes for hexahedral element with NQ = 2

Example of SPH nodes for pentahedral element with NQ = 2



Example of SPH nodes for tetrahedral element with NQ = 2

Figure 17-2. Examples of distributions of SPH particles converted from solid Lagrangian elements.

of generated SPH particles is 1, 6, and 18 for NQ equal to 1, 2, and 3, respectively. For tetrahedral elements, the number of generated SPH particles is 1, 4, and 10 for NQ equal to 1, 2, and 3, respectively. See [Figure 17-2](#).

2. **SPH Part ID.** The Part ID for newly generated SPH particles can be either a new Part ID or the ID of an existing SPH Part. For constraint coupling, such as when ICPL = 1 and IOPT = 0, the newly generated SPH part ID should be different from the existing one.
3. **Mechanical Failure Coupling.** ICPL = 0 is used for debris simulation, so no coupling happens between newly generated SPH particles and solid elements; the user, therefore, needs to define node to surface contact for the interaction

between those two parts. When ICPL = 1 and IOPT = 1, the newly generated SPH particles are bonded with solid elements as one part through the coupling, and the new material ID with different failure criteria can be applied to the newly generated SPH particles.

4. **Thermal Coupling.** ICPL = 3, which must be combined with IOPT = 0, is used to thermally couple an SPH part and solid part(s). There is no structural coupling provided. A thermal conductivity value may be defined using the variable CPCD.

DEFINE_BEAM_SOLID_COUPLING**DEFINE*****DEFINE_BEAM_SOLID_COUPLING**

Purpose: To define a coupling interface between embedded structure and a block defined by solid(s) without sharing nodes.

Card 1	1	2	3	4	5	6	7	8
Variable	LSTRID	SOLID	LSTRTYPE	SOLTYPE	FORM	PSF		
Type	I	I	I	I	I	F		
Default	none	none	0	0	0	1.		

VARIABLE	DESCRIPTION
LSTRID	Part set ID or part ID of the Lagrangian structure. LSTRTYPE below indicates the ID type specified by LSTRTYPE.
SOLID	Part set ID or part ID of the solid block. SOLTYPE below indicates the ID type specified by SOLID.
LSTRTYPE	Type of Lagrangian structure set: EQ.0: Part set EQ.1: Part
TYPE	Type of solid set: EQ.0: Part set EQ.1: Part
Form	Coupling type: EQ.0: Constrained acceleration and velocity EQ.1: Penalty tied in all directions
PSF	Scale factor for penalty stiffness

*DEFINE

*DEFINE_BOX

*DEFINE_BOX_{OPTION}

Available options include:

<BLANK>

LOCAL

Purpose: Define a box-shaped volume. Two diagonally opposite corner points of a box are specified in global or local coordinates if the LOCAL option is active. The box volume is then used for various specifications for a variety of input options, such as velocities, contact, etc.

If the option, LOCAL, is active, a local coordinate system with two vectors (see [Figure 17-12](#)) is defined. The vector cross product, $\mathbf{z} = \mathbf{x} \times \mathbf{v}_{xy}$ where \mathbf{v}_{xy} is a vector in the xy -plane, determines the local z -axis. The local y -axis is then given by $\mathbf{y} = \mathbf{z} \times \mathbf{x}$.

A point, P , in the global coordinate system is considered to lie within the volume of the box if the coordinate $P - C$, where C is the global coordinate offset vector defined on Card 3, lies within the box after transformation into the local system, $P_{\text{local}} = T \times (P - C)$. The local coordinate, P_{local} , is checked against the minimum and maximum coordinates defined on Card 1 in the local system.

For the *INCLUDE_TRANSFORM options that include translations, rotations and mirrors, all box options are automatically converted from *DEFINE_BOX_xxxx to *DEFINE_BOX_xxxx_LOCAL in the DYNA.INC file. Here, xxxx represents the box options: ADAPTIVE, COARSEN, and SPH, which are defined below. If the transformation matrix of *INCLUDE_TRANSFORM involves using negative values for SCALE, see *DEFINE_TRANSFORMATION, to mirror/reflect an object, the box might not be correctly transformed. Instead of SCALE, it is advised to use the MIRROR option with A7 = 1 for the purpose of mirroring/reflecting.

Card 1	1	2	3	4	5	6	7	8
Variable	BOXID	XMN	XMX	YMN	YMX	ZMN	ZMX	
Type	I	F	F	F	F	F	F	
Default	0	0.0	0.0	0.0	0.0	0.0	0.0	

Local Card 1. First additional card for LOCAL keyword option.

Card 2	1	2	3	4	5	6	7	8
Variable	XX	YX	ZX	XV	YV	ZV		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	0.0	0.0	0.0		

Local Card 2. Second additional card for LOCAL keyword option.

Card 3	1	2	3	4	5	6	7	8
Variable	CX	CY	CZ					
Type	F	F	F					
Default	0.0	0.0	0.0					

VARIABLE	DESCRIPTION
BOXID	Box ID. Define unique numbers.
XMN	Minimum <i>x</i> -coordinate. Define in the local coordinate system if the option LOCAL is active.
XMX	Maximum <i>x</i> -coordinate. Define in the local coordinate system if the option LOCAL is active.
YMN	Minimum <i>y</i> -coordinate. Define in the local coordinate system if the option LOCAL is active.
YMX	Maximum <i>y</i> -coordinate. Define in the local coordinate system if the option LOCAL is active.
ZMN	Minimum <i>z</i> -coordinate. Define in the local coordinate system if the option LOCAL is active.
ZMX	Maximum <i>z</i> -coordinate. Define in the local coordinate system if the option LOCAL is active.

VARIABLE	DESCRIPTION
XX	X-coordinate on local x -axis. Origin lies at (0,0,0). Define if the LOCAL option is active.
YX	Y -coordinate on local x -axis. Define if the LOCAL option is active.
ZX	Z -coordinate on local x -axis. Define if the LOCAL option is active.
XV	X-coordinate of local xy -vector. Define if the LOCAL option is active.
YV	Y -coordinate of local xy -vector. Define if the LOCAL option is active.
ZV	Z -coordinate of local xy -vector. Define if the LOCAL option is active.
CX	X-global coordinate of offset vector to origin of local system. Define if the LOCAL option is active.
CY	Y -global coordinate of offset vector to origin of local system. Define if the LOCAL option is active.
CZ	Z -global coordinate of offset vector to origin of local system. Define if the LOCAL option is active.

DEFINE_BOX_ADAPTIVE**DEFINE*****DEFINE_BOX_ADAPTIVE_{OPTION}**

Available options include:

<BLANK>

LOCAL

Purpose: Define a box-shaped volume enclosing (1) the shells where the h -adaptive level is to be specified, or (2) the solids where the tetrahedral r -adaptive mesh size is to be specified. If the midpoint of the element falls within the box, the h -adaptive level is reset. The box can translate. It is also possible to define a fission box followed by a fusion box, in which case the mesh can refine when deformed and coarsen when flattened. Shells falling outside of this volume use the value, MAXLVL, on the *CONTROL_ADAPTIVE control cards. A related keyword is *DEFINE_CURVE_BOX_ADAPTIVITY. An alternative command in the case of shell h -adaptivity is *DEFINE_BOX_NODES_ADAPTIVE.

Card 1	1	2	3	4	5	6	7	8
Variable	BOXID	XMN	XMX	YMN	YMX	ZMN	ZMX	
Type	I	F	F	F	F	F	F	
Default	none	0.0	0.0	0.0	0.0	0.0	0.0	

Card 2	1	2	3	4	5	6	7	8
Variable	PID	LEVEL	LIDX/NDID	LIDY	LIDZ	BRMIN	BRMAX	
Type	I	I	I	I	I	F	F	
Default	0	1	0	0	0	0.0	0.0	

DEFINE**DEFINE_BOX_ADAPTIVE**

Local Card 1. First additional card for LOCAL keyword option. See *DEFINE_BOX for a description of the LOCAL option.

Card 3	1	2	3	4	5	6	7	8
Variable	XX	YX	ZX	XV	YV	ZV		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	0.0	0.0	0.0		

Local Card 2. Second additional card for LOCAL keyword option.

Card 4	1	2	3	4	5	6	7	8
Variable	CX	CY	CZ					
Type	F	F	F					
Default	0.0	0.0	0.0					

VARIABLE	DESCRIPTION
BOXID	Box ID. Define unique numbers.
XMN	Minimum <i>x</i> -coordinate. Define in the local coordinate system if the option LOCAL is active.
XMX	Maximum <i>x</i> -coordinate. Define in the local coordinate system if the option LOCAL is active.
YMN	Minimum <i>y</i> -coordinate. Define in the local coordinate system if the option LOCAL is active.
YMX	Maximum <i>y</i> -coordinate. Define in the local coordinate system if the option LOCAL is active.
ZMN	Minimum <i>z</i> -coordinate. Define in the local coordinate system if the option LOCAL is active.
ZMX	Maximum <i>z</i> -coordinate. Define in the local coordinate system if the option LOCAL is active.

VARIABLE	DESCRIPTION
PID	Deformable part ID. EQ.0: all active elements within box are considered.
LEVEL	Maximum number of refinement levels for elements that are contained in the box. Values of 1, 2, 3, 4, ... allow for a maximum of 1, 4, 16, 64, ... elements, respectively, to be created for each original element.
LIDX/NDID	Load curve ID / Node ID (see Remark 1). GT.0: load curve ID. Define adaptive box movement (displacement as a function of time) in global X-axis. LT.0: absolute value is a node ID, whose translation will be followed by the moving adaptive box. The node ID can be on a moving rigid body. EQ.0: no movement
LIDY	Load curve ID (see Remark 1). GT.0: load curve ID. Define adaptive box movement (displacement as a function of time) in global Y-axis. EQ.0: no movement
LIDZ	Load curve ID (see Remark 1). GT.0: load curve ID. Define adaptive box movement (displacement as a function of time) in global Z-axis. EQ.0: no movement
BRMIN	Minimum mesh size in 3D tetrahedron adaptivity
BRMAX	Maximum mesh size in 3D tetrahedron adaptivity
XX	X-coordinate on local x -axis. Origin lies at (0,0,0). Define if the LOCAL option is active.
YX	Y-coordinate on local x -axis. Define if the LOCAL option is active.
ZX	Z-coordinate on local x -axis. Define if the LOCAL option is active.
XV	X-coordinate of local xy -plane vector. Define if the LOCAL option is active.

*DEFINE

*DEFINE_BOX_ADAPTIVE

VARIABLE	DESCRIPTION
YV	Y-coordinate of local xy -plane vector. Define if the LOCAL option is active.
ZV	Z-coordinate of local xy -plane vector. Define if the LOCAL option is active.
CX	X-global coordinate of offset vector to origin of local system. Define if the LOCAL option is active.
CY	Y-global coordinate of offset vector to origin of local system. Define if the LOCAL option is active.
CZ	Z-global coordinate of offset vector to origin of local system. Define if the LOCAL option is active.

Remarks:

1. **Moving Box.** The moving adaptive box is very useful and efficient in situations where deformations happens locally, such as roller hemming and incremental forming simulations. With the moving box feature, elements entering one box can be refined and fused together when they enter another box. Mesh fission outside of the moving box envelope is controlled by MAXLVL and other parameters under *CONTROL_ADAPTIVE. The fusion controls (NCFREQ, IADPCL) can be defined using *CONTROL_ADAPTIVE. Currently, only IADPCL = 1 is supported.

Only when one of the LCIDX/NDID, LICDY, or LCIDZ is defined, the adaptive box will be moving; otherwise it will be stationary.

2. **3D r -Adaptivity.** For 3D tetrahedron r -adaptivity, the current implementation does not support the LOCAL option. In Card 2, LEVEL is not supported in 3D r -adaptivity.

Example:

Referring to a partial input deck below, and [Figure 17-3](#), a strip of sheet metal is being roller hemmed. The process consists of pre- and final hemming. Each pre- and final roller is defined with a moving adaptive box with IDs 2 and 3, respectively. [Figure 17-3](#) shows the box shapes. The first box, a fission box, has a LEVEL = 3 refinement, while the second box, a fusion box, was set at LEVEL = 1 refinement. Elements outside of the volume envelope made by the moving boxes undergo no fission and fusion (MAXLVL = 1). These settings cause mesh fission when the elements enter moving box 2 (LEVEL = 3) and mesh fusion only when the elements enter moving box 3 (LEVEL = 1). No fission/fusion

(MAXLVL = 1) occurs outside of the volume envelope created by the moving boxes. In the example, the boxes 2 and 3 move in the global X-direction for a distance of 398 mm as defined by load curve 11 and 450 mm as defined by load curve 12, respectively.

```

*CONTROL_TERMINATION
0.252
*CONTROL_ADAPTIVE
$ ADPFREQ ADPTOL ADPTYP MAXLVL TBIRTH TDEATH LCADP IOFLAG
  8.05E-4 0.200000 2 1 0.0001.0000E+20 0 1
$ ADPSIZE ADPASS IREFLG ADPENE ADPTH MEMORY ORIENT MAXEL
  0.300000 1 0 5.0
$ IADPN90 NCFREQ IADPCL ADPCTL CBIRTH CDEATH
  -1 0 1 1 10.0 0.000 10.30
*DEFINE_BOX_ADAPTIVE
$# BOXID XMN XMX YMN YMXX ZMN ZMX
  2 -10.00000 36.000000 -15.03000 3.991000 1.00E+00 48.758000
$# PID LEVEL LIDX/NDID LIDY LIDZ
  6 3 11
*DEFINE_BOX_ADAPTIVE
$# BOXID XMN XMX YMN YMXX ZMN ZMX
  3 -100.0000 -60.0000 -15.03000 3.991000 1.00E+00 48.758000
$# PID LEVEL LIDX/NDID LIDY LIDZ
  6 1 12
*DEFINE_CURVE
11
  0.000 0.0
  0.00100000 1.0
  0.19900000 397.0
  0.20000000 398.0
  1.000 398.0
  :
*DEFINE_CURVE
12
  0.0 0.0
  0.05 0.0
  0.051 1.0
  0.251 401.0
  0.252 450.0
  :

```

A moving box can also follow the movement of a node, which can be on a moving rigid body. In this case, NDIDs for the motion of the boxes are defined instead of load curves. For example, in [Figure 17-3](#) and a partial keyword example below, box 2 follows a node (ID: 33865) on the pre-roller, and box 3 follows another node (ID: 38265) on the final roller.

```

*DEFINE_BOX_ADAPTIVE
$# BOXID XMN XMX YMN YMXX ZMN ZMX
  2 -10.00000 36.000000 -15.03000 3.991000 1.00E+00 48.758000
$# PID LEVEL LIDX/NDID LIDY LID
  6 3 -33865
*DEFINE_BOX_ADAPTIVE
$# BOXID XMN XMX YMN YMXX ZMN ZMX
  3 -100.0000 -60.0000 -15.03000 3.991000 1.00E+00 48.758000
$# PID LEVEL LIDX/NDID LIDY LID
  6 3 -38265

```

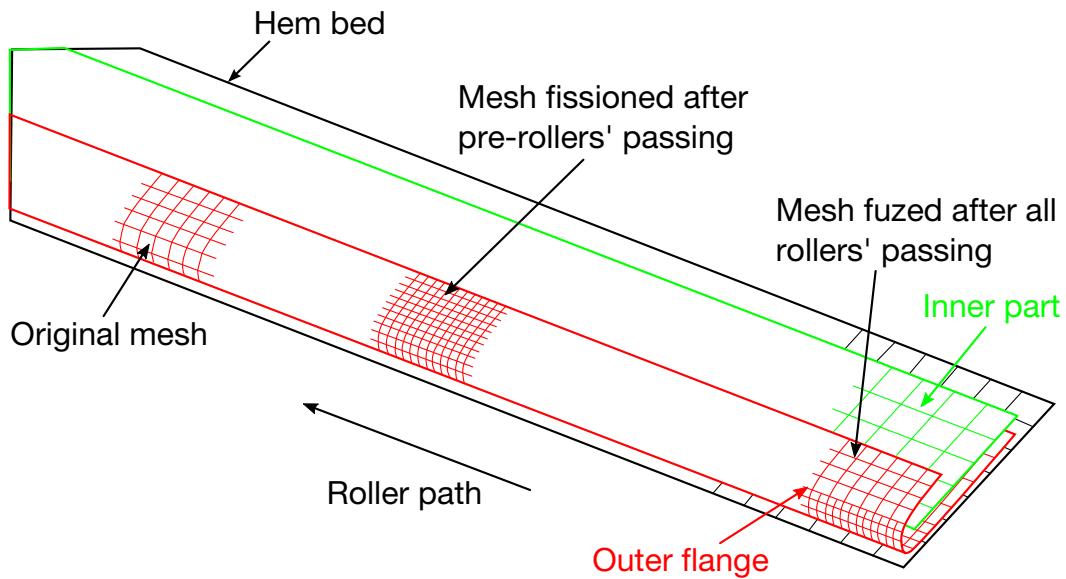
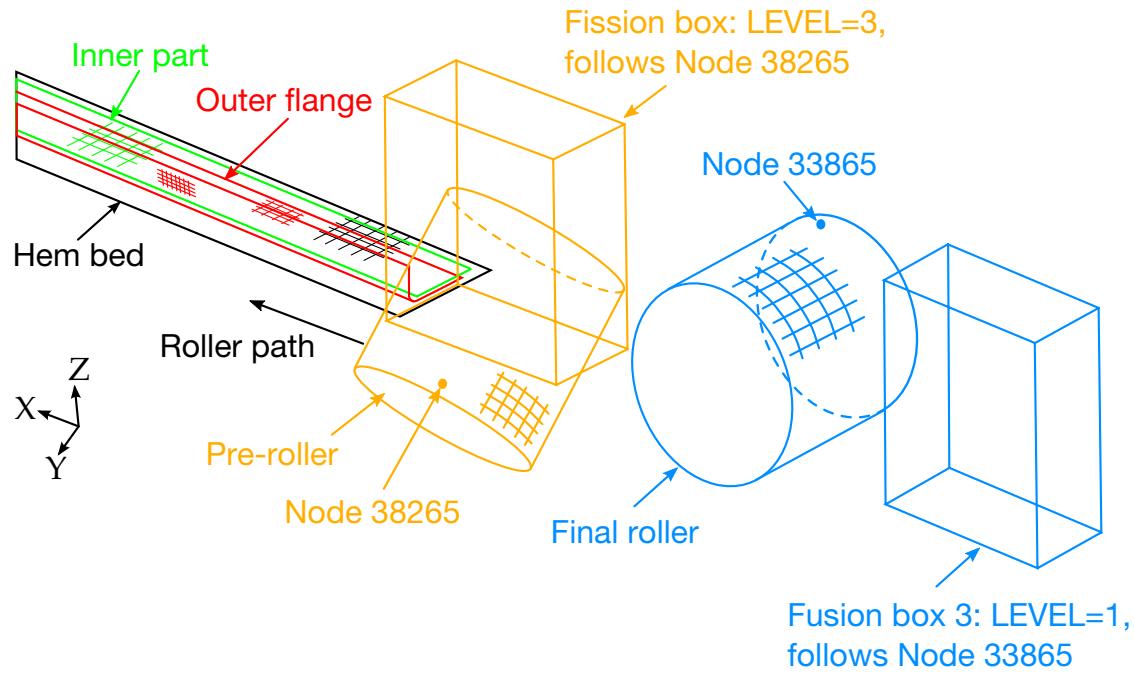


Figure 17-3. Defining mesh fission and fusion.

***DEFINE_BOX_COARSEN_{OPTION}**

Available options include:

<BLANK>

LOCAL

Purpose: Define a specific box-shaped volume indicating elements which are protected from mesh coarsening. See also *CONTROL_COARSEN.

Card	1	2	3	4	5	6	7	8
Variable	BOXID	XMN	XMX	YMN	YMX	ZMN	ZMX	IFLAG
Type	I	F	F	F	F	F	F	I
Default	none	0.0	0.0	0.0	0.0	0.0	0.0	0

Local Card 1. First additional card for LOCAL keyword option. See *DEFINE_BOX for a description of the LOCAL option.

Card 2	1	2	3	4	5	6	7	8
Variable	XX	YX	ZX	XV	YV	ZV		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	0.0	0.0	0.0		

Local Card 2. Second additional card for LOCAL keyword option.

Card 3	1	2	3	4	5	6	7	8
Variable	CX	CY	CZ					
Type	F	F	F					
Default	0.0	0.0	0.0					

DEFINE**DEFINE_BOX_COARSEN**

VARIABLE	DESCRIPTION
BOXID	Box ID. Define unique numbers.
XMN	Minimum <i>x</i> -coordinate. Define in the local coordinate system if the option LOCAL is active.
XMX	Maximum <i>x</i> -coordinate. Define in the local coordinate system if the option LOCAL is active.
YMN	Minimum <i>y</i> -coordinate. Define in the local coordinate system if the option LOCAL is active.
YMX	Maximum <i>y</i> -coordinate. Define in the local coordinate system if the option LOCAL is active.
ZMN	Minimum <i>z</i> -coordinate. Define in the local coordinate system if the option LOCAL is active.
ZMX	Maximum <i>z</i> -coordinate. Define in the local coordinate system if the option LOCAL is active.
IFLAG	Flag for protecting elements inside or outside of box: EQ.0: elements inside the box cannot be coarsened EQ.1: elements outside the box cannot be coarsened
XX	X-coordinate on local <i>x</i> -axis. Origin lies at (0,0,0). Define if the LOCAL option is active.
YX	<i>Y</i> -coordinate on local <i>x</i> -axis. Define if the LOCAL option is active.
ZX	<i>Z</i> -coordinate on local <i>x</i> -axis. Define if the LOCAL option is active.
XV	X-coordinate of local <i>xy</i> -vector. Define if the LOCAL option is active.
YV	<i>Y</i> -coordinate of local <i>xy</i> -vector. Define if the LOCAL option is active.
ZV	<i>Z</i> -coordinate of local <i>xy</i> -vector. Define if the LOCAL option is active.
CX	X-global coordinate of offset vector to origin of local system. Define if the LOCAL option is active.

VARIABLE	DESCRIPTION
CY	Y-global coordinate of offset vector to origin of local system. Define if the LOCAL option is active.
CZ	Z-global coordinate of offset vector to origin of local system. Define if the LOCAL option is active.

Remarks:

1. **Multiple Boxes.** Many boxes may be defined. If an element is protected by any box then it may not be coarsened.

*DEFINE

*DEFINE_BOX_DRAWBEAD

*DEFINE_BOX_DRAWBEAD

Purpose: Define a specific box or tube shaped volume around a draw bead. This option is useful for the draw bead contact. If box shaped, the volume will contain the draw bead nodes and elements between the bead and the outer edge of the blank. If tubular, the tube is centered around the draw bead. All elements within the tubular volume are included in the contact definition.

Card	1	2	3	4	5	6	7	8
Variable	BOXID	PID	SID	IDIR	STYPE	RADIUS	CID	
Type	I	F	F	F	I	F	I	
Default	0	0.0	0.0	0.0	4	0.0	0	

VARIABLE	DESCRIPTION
BOXID	Box ID. Define unique numbers.
PID	Part ID of blank.
SID	Set ID that defines the nodal points that lie along the draw bead. If a node set is defined, the nodes in the set must be consecutive along the draw bead. If a part or part set is defined, the set must consist of beam or truss elements. Within the part set, no ordering of the elements is assumed, but the number of nodes must equal the number of beam elements plus 1.
IDIR	Direction of tooling movement. The movement is in the global coordinate direction unless the tubular box option is active and CID is nonzero. In this latter case, the movement is in the local coordinate direction. EQ.1: tooling moves in <i>x</i> -direction, EQ.2: tooling moves in <i>y</i> -direction, EQ.3: tooling moves in <i>z</i> -direction.
STYPE	Set type: EQ.2: part set ID, EQ.3: part ID,

VARIABLE	DESCRIPTION
	EQ.4: node set ID.
RADIUS	The radius of the tube, which is centered around the draw bead. Elements of part ID, PID, that lie within the tube will be included in the contact. If the radius is not defined, a rectangular box is used instead. This option is recommended for curved draw beads and for draw beads that are not aligned with the global axes.
CID	Optional coordinate system ID. This option is only available for the tubular drawbead.

*DEFINE

*DEFINE_BOX_NODES_ADAPTIVE

*DEFINE_BOX_NODES_ADAPTIVE

Purpose: Define mesh fission and fusion at the beginning of an adaptive step according to the path of a moving tool and a set of parameters. This method applies only to shell h-adaptivity and is sometimes referred to as “tube” adaptivity for its adaptive boundary shape. (This tube is more accurately described as a torus.) It can help reduce the computational time for incremental forming or roller hemming simulations while maintaining accuracy in the area of interest. This method is a major improvement over the keyword *DEFINE_BOX_ADAPTIVE (see [Remarks](#) and [Figure 17-4](#)). Related keywords include *DEFINE_CURVE_BOX_ADAPTIVITY and *CONTROL_ADAPTIVE_CURVE.

Card 1	1	2	3	4	5	6	7	8
Variable	BOXID	NODE	LCX	LCY	LCZ	ITYPE	RADIUS	NPIECE
Type	I	F	F	F	F	F	F	
Default	none	none	none	none	none	none	0.0	0

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

VARIABLE	DESCRIPTION
BOXID	Box ID. Define unique numbers.
NODE	A reference node ID from which the tube will form in front of it, following the tool path described by LCX, LCY and LCZ.
LCX, LCY, LCZ	Load curve IDs (see *DEFINE_CURVE) that define the path of the tool in the global X, Y, and Z directions, respectively.

VARIABLE	DESCRIPTION
ITYPE	Type of curves LCX, LCY and LCZ. Currently only time as a function of displacement load curves are supported. EQ.2: LCX, LCY and LCZ are defined as time as a function of displacement.
RADIUS	The radius of the tube that defines the fission/fusion boundary.
NPIECE	Number of segments used to approximate the tool path in one adaptive step. Note that the tool's path is divided into several linear segments for approximation.
PID	The deformable part or part set ID on which the tube adaptivity is to be applied (see *PART). GT.0: Part ID LT.0: PID is a part set ID. A part set ID can be useful for simulating the forming of tailor welded blanks.
LEVEL	Desired mesh refinement level. Level set to a value of 1, 2, 3, ... allows a maximum of 1, 4, 16, ... elements to be created for each original element in the "tube region".

Remarks:

Metal forming simulations for applications similar to incremental forming and roller hemming are computationally intensive, mostly because of long tool paths and localized deformation. Several methods have been developed to increase the efficiency of these applications.

The keyword *DEFINE_BOX_ADAPTIVE was one of those methods. While it offered significant CPU time reduction, the adaptive fission box does not stay in front of the tool when the tool path turns around 180 degrees (see [Figure 17-4](#)). Also, the fusion box, which is supposed to stay behind the tool to coarsen the mesh, does not always stay behind the tool. This drawback is addressed by this keyword.

This keyword creates a tube with a user defined radius (RADIUS) that encloses the tool path ahead and moves along with a reference node (NODE) located on the tool. The mesh is refined in the area in front of the tube (look-forward adaptivity). The tube moves with the tool. Formed mesh behind the tube is fused according to user set criterion, thus saving CPU time (see [Figure 17-5](#)).

From a parameter study of this keyword [Zhu et. al.], the following was determined:

- In comparison to a convergence study without this keyword, the simulation time was reduced by 46%.
- The optimum tube radius is 10 times the final element size.
- The optimum ADPFREQ (see *CONTROL_ADAPTIVITY) should be about 1.6% of the total simulation time. Note the ADPFREQ indirectly controls how far the tube with extend in front of the tool.

Example:

To activate this feature, in addition to using this keyword, *CONTROL_ADAPTIVE also needs to be included in the input deck. See [Figure 17-5](#) and the partial input below.

```
*CONTROL_ADAPTERIVE
$ ADPFREQ      ADPTOL      ADPTYP      MAXLVL      TBIRTH      TDEATH      LCADP      IOFLAG
    0.08        1.0          2            2           0.0         13.6          0           1
$ ADPSIZE      ADPASS      IREFLG      ADPENE      ADPTH       MEMORY      ORIENT      MAXEL
                1            0            0.0          0.0
$ IADPN90     IADPGH      NCFREQ      IADPCL      ADPCTL      CBIRTH      CDEATH      LCLVL
    -1          0            4            0           0.5         0.0         13.6
*I DEFINE _BOX _NODES _ADAPTIVE
$ ID          NODE        LCX          LCY          LCZ          ITYPE      RADIUS      NPIECE
    2          114116      1            2            3            2          10.0          8
$ PID         LEVEL
    7            2
```

Reference:

Zhu, X.H., Fan, H.F., Zhang, L., and Xiao, Y.Z., *Tube adaptivity for mesh fission/fusion in LS-DYNA, 2017 3rd China LS-DYNA Users' Conference, Shanghai, China.*

Revision information:

This keyword is available in both SMP and MPP starting in Revision 120597 and is jointly developed with Nissan Motor Corporation.

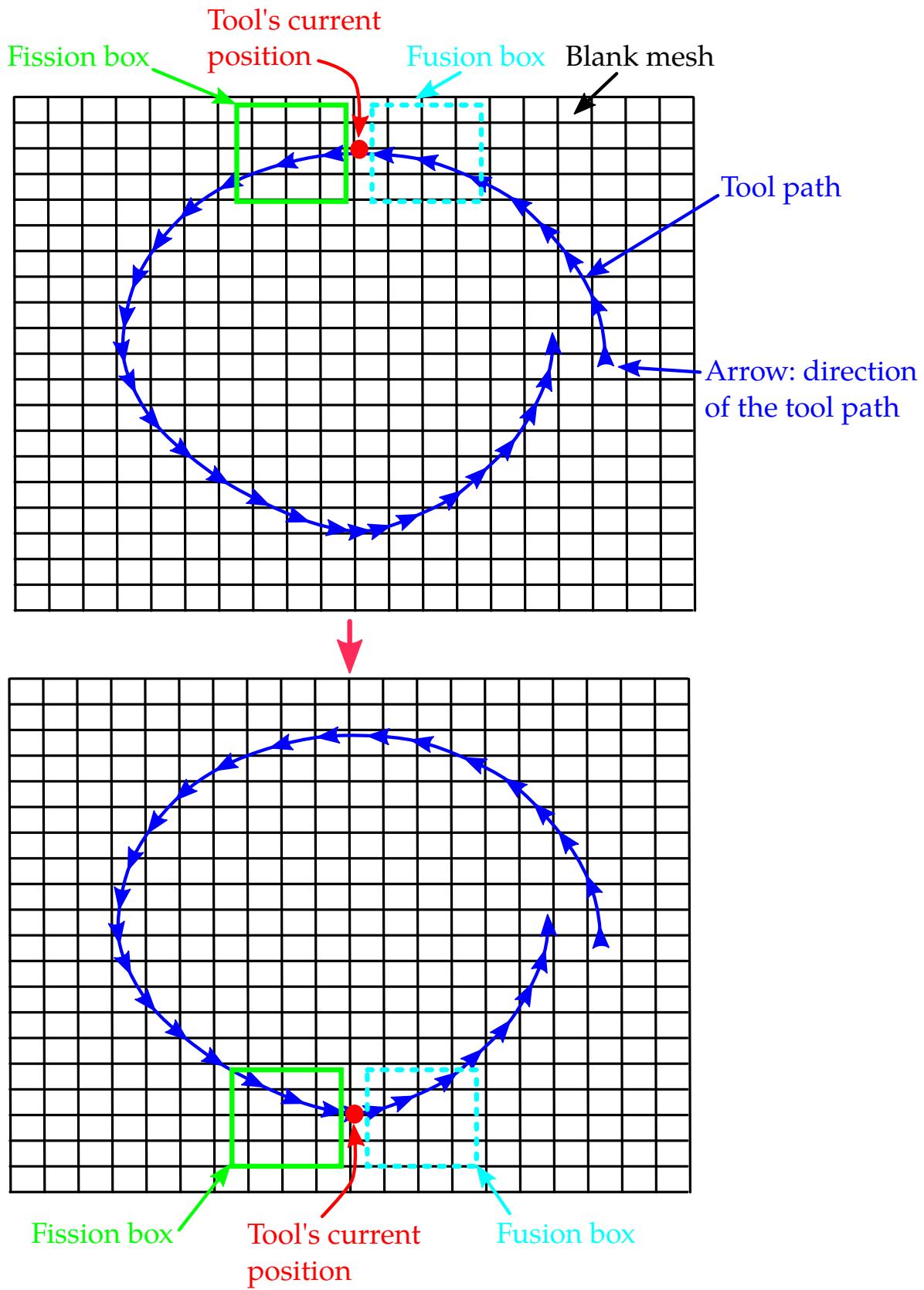


Figure 17-4. Limitation of the existing capability by ***DEFINE_BOX_NODES_ADAPTIVE**.

***DEFINE**

***DEFINE_BOX_NODES_ADAPTIVE**

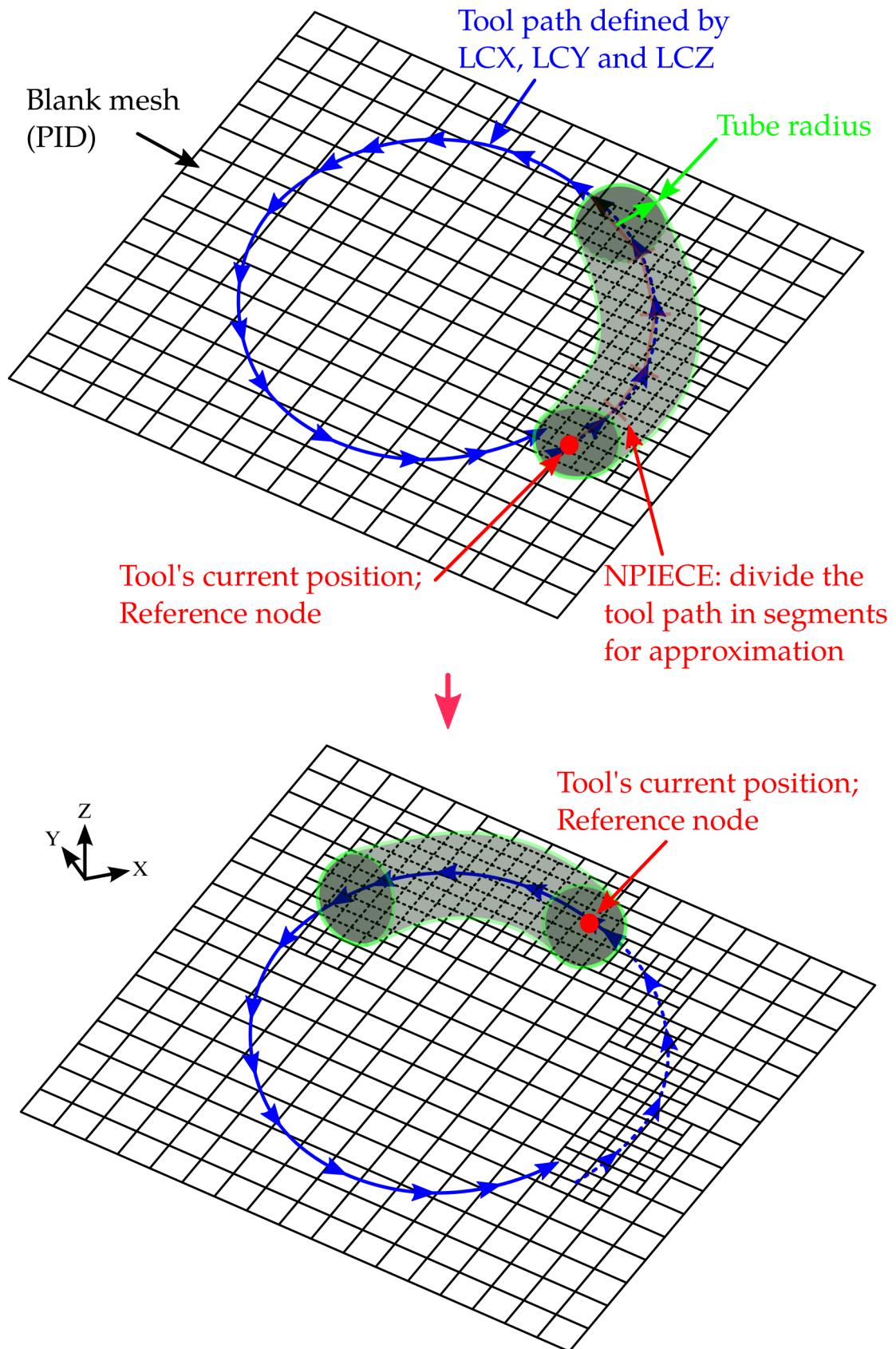


Figure 17-5. “Tube” adaptivity.

***DEFINE_BOX_SPH_{OPTION}**

Available options include:

<BLANK>

LOCAL

Purpose: Define a box-shaped volume for activating and deactivating SPH particles. To define the box, you specify two diagonally opposite corner points. The corner points are assumed to be in the global coordinate system unless the LOCAL keyword option is active. See *DEFINE_BOX for more details about the LOCAL option. When particles are in the box, the SPH particles are active and LS-DYNA computes the SPH approximation (see FORM in *CONTROL_SPH) for the particles. SPH particles are deactivated as they leave the box but reactivate if they re-enter the box. Unlike a box specified with *DEFINE_BOX, the box may move either with a node or in the direction defined by a vector with a motion determined by a load curve.

WARNING: The box specified for deactivating/activating particles by *CONTROL_SPH (see BOXID) must be a *DEFINE_BOX, not a *DEFINE_BOX_SPH. If BOXID references a *DEFINE_BOX_SPH, LS-DYNA will give an error message.

Card 1	1	2	3	4	5	6	7	8
Variable	BOXID	XMN	XMX	YMN	YMX	ZMN	ZMX	VID
Type	I	F	F	F	F	F	F	I
Default	none	0.0	0.0	0.0	0.0	0.0	0.0	0

Card 2	1	2	3	4	5	6	7	8
Variable	LCID	VD	NID	IREACT	IBUFF	ISHOW	PID	
Type	I	I	I	I	I	I	I	
Default	0	0	0	0	0	0	0	

DEFINE**DEFINE_BOX_SPH**

Local Card 1. First additional card for the LOCAL keyword option. See *DEFINE_BOX for a description of the LOCAL option.

Card 3	1	2	3	4	5	6	7	8
Variable	XX	YX	ZX	XV	YV	ZV		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	0.0	0.0	0.0		

Local Card 2. Second additional card for the LOCAL keyword option.

Card 4	1	2	3	4	5	6	7	8
Variable	CX	CY	CZ					
Type	F	F	F					
Default	0.0	0.0	0.0					

VARIABLE	DESCRIPTION
BOXID	Box ID. Define unique numbers.
XMN	Minimum <i>x</i> -coordinate. Define in the local coordinate system if the option LOCAL is active.
XMX	Maximum <i>x</i> -coordinate. Define in the local coordinate system if the option LOCAL is active.
YMN	Minimum <i>y</i> -coordinate. Define in the local coordinate system if the option LOCAL is active.
YMX	Maximum <i>y</i> -coordinate. Define in the local coordinate system if the option LOCAL is active.
ZMN	Minimum <i>z</i> -coordinate. Define in the local coordinate system if the option LOCAL is active.
ZMX	Maximum <i>z</i> -coordinate. Define in the local coordinate system if the option LOCAL is active.

VARIABLE	DESCRIPTION
VID	Vector ID to describe direction of box motion when VD = 0 or 1; see *DEFINE_VECTOR.
LCID	Load curve ID to describe value of box motion as a function of time; see *DEFINE_CURVE
VD	Velocity/Displacement flag: EQ.0: Velocity, EQ.1: Displacement, EQ.2: Referential node
NID	Referential nodal ID for VD = 2 (SPH box will move with this node).
IREACT	Reactivation flag: EQ.0: Particles outside of the box are permanently deactivated. EQ.1: Deactivated particles get reactivated when they enter the box.
IBUFF	Buffer zone flag: EQ.0: Particles on the edge of the box do not get any special treatment. EQ.1: Particles on the edge of the box are frozen in space and act as neighbors for active particles inside the box. This option is mainly used for fluid simulations to prevent the fluid from spilling out of the activation box.
ISHOW	Create dummy part for visualizing the position of the activation box during post-processing: EQ.0: No part is created. EQ.1: A dummy part is added for visualization.
PID	Part ID used for visualization if ISHOW = 1: EQ.0: A unique part ID is automatically created. GT.0: PID is the part ID for the created part. This should be a unique part ID.
XX	X-coordinate on local x-axis. Origin lies at (0,0,0). Define if the LOCAL option is active.

DEFINE**DEFINE_BOX_SPH**

VARIABLE	DESCRIPTION
YX	Y -coordinate on local x -axis. Define if the LOCAL option is active.
ZX	Z -coordinate on local x -axis. Define if the LOCAL option is active.
XV	X-coordinate of local xy -vector. Define if the LOCAL option is active.
YV	Y -coordinate of local xy -vector. Define if the LOCAL option is active.
ZV	Z -coordinate of local xy -vector. Define if the LOCAL option is active.
CX	X-global coordinate of offset vector to origin of local system. Define if the LOCAL option is active.
CY	Y -global coordinate of offset vector to origin of local system. Define if the LOCAL option is active.
CZ	Z -global coordinate of offset vector to origin of local system. Define if the LOCAL option is active.

***DEFINE_CABLE**

Purpose: Create a cable consisting of a beam element core surrounded by solid elements. This setup is commonly used for modeling electric cables, where the solid elements represent the insulator and copper core, and the beam elements give additional bending stiffness. This keyword causes the automatic generation of the solid elements from the beam element core. The output file `define_cable.k` contains the generated elements, nodes, and connectivity for visualization and further pre-processing. Running an input deck containing this keyword with `define_cable.k` included causes using the solids, nodes, and connectivity from the file instead. Cross-sectional data, such as contact force, stress, and area, are calculated from the solid elements and saved for each corresponding beam element. The automatically output ascii file `cablestats` contains summary statistics for all the cables present. [*DATABASE_CABLE](#) activates outputting element-wise data to bin-out and/or the ascii file `cableout`.

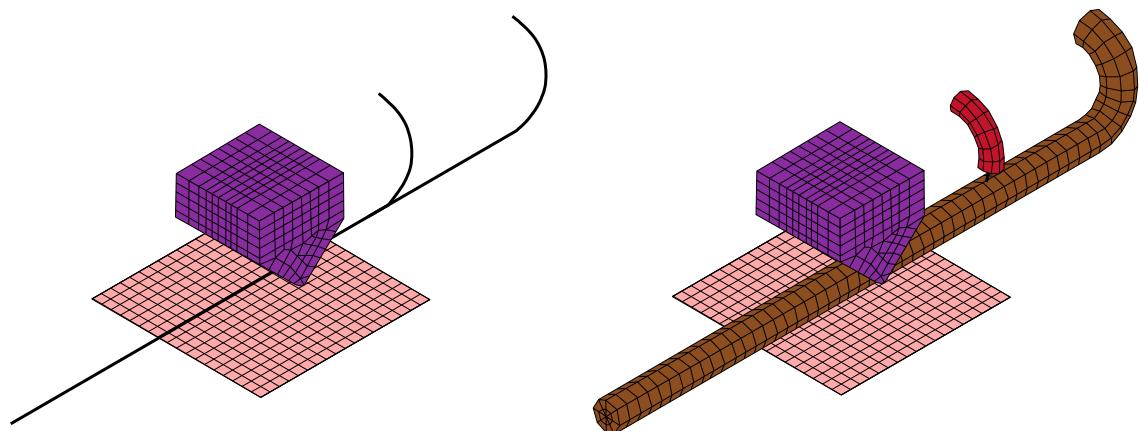


Figure 17-6. Example of initial beam geometry and resulting solid/beam geometry

Card Summary:

Card 1. This card is required.

BPID	NLAYR	NCIRC	NCYCL	NLPID	ELNR	NDNR	TRIM

Card 2. This card is required.

SPID	STHK						

DEFINE**DEFINE_CABLE****Data Card Definitions:**

Card 1	1	2	3	4	5	6	7	8
Variable	BPID	NLAYR	NCIRC	NCYCL	SHLPID	ELNR	NDNR	TRIM
Type	I	I	I	I	I	I	I	I
Default	0	0	0	1	0	0	0	0

VARIABLE	DESCRIPTION
BPID	Beam part ID for the cable core. All connected beam elements in the part model a cable. For MPP, all elements in the beam part are put on a single processor.
NLAYR	Number of solid layers in thickness
NCIRC	Number of solid elements on the circumference
NCYCL	Collect data every NCYCL cycle
SHLPID	Optional PID to create shells on the solid faces. This is commonly used for extra null shells that may increase contact stability. This part ID needs to be for an existing part without any elements associated with it. SHLPID on this card cannot be used by any other *DEFINE_CABLE instantiations.
ELNR	Optional starting element number for generated solids. If several cable keywords have an ELNR that causes collisions, ELNR will be automatically adjusted.
NDNR	Optional starting node number for generated nodes. If several cable keywords have an NDNR that causes collisions, NDNR will be automatically adjusted.
TRIM	Set to 1 if adjoint cables need to be trimmed, meaning a cable connected to the interior of another cable will not create new solids such that the connection has overlapping solid elements.

DEFINE_CABLE**DEFINE**

Card 2	1	2	3	4	5	6	7	8
Variable	SLDPID	SLDTHK						
Type	I	F						
Default	0	0.0						

VARIABLE**DESCRIPTION**

SLDPID Solid part ID for the automatically generated solids. This part ID needs to be an existing part without any elements associated with it. SLDPID on this card cannot be used by any other *DEFINE_CABLE instantiations.

SLDTHK Thickness for each solid layer

*DEFINE

*DEFINE_CONNECTION_PROPERTIES

*DEFINE_CONNECTION_PROPERTIES_{OPTION}

Available options include:

<BLANK>

ADD

Purpose: Define failure related parameters for solid element spot weld failure by *MAT_SPOTWELD_DAIMLERCHRYSLER. For each connection identifier, CON_ID, a separate *DEFINE_CONNECTION_PROPERTIES section must be included. The ADD keyword option allows material specific properties to be added to an existing connection ID. See [Remark 2](#).

Card Summary:

Card 1. This card is required.

CON_ID	PRUL	AREAEQ		DGTYP	MOARFL		
--------	------	--------	--	-------	--------	--	--

Card 2. This card is omitted if the ADD keyword option is used. See [Remark 2](#).

	DSIGY	DETAN	DDGPR	DRANK	DSN	DSB	DSS
--	-------	-------	-------	-------	-----	-----	-----

Card 3. This card is omitted if the ADD keyword option is used. See [Remark 2](#).

DEXSN	DEXSB	DEXSS	DLCSN	DLCNB	DLCSS	DGFAD	DSCLMRR
-------	-------	-------	-------	-------	-------	-------	---------

Card 4. This card is included if he ADD keyword option is used. For each shell material with material specific data, define for this CON_ID Cards 4 and 5. Add as many pairs of these cards as necessary. This input is terminated by the next keyword ("*") card. See [Remark 2](#).

MID	SGIY	ETAN	DGPR	RANK	SN	SB	SS
-----	------	------	------	------	----	----	----

Card 5. This card is included if the ADD keyword option is used. For each shell material with material specific data, define for this CON_ID Cards 4 and 5. Add as many pairs of these cards as necessary. This input is terminated by the next keyword ("*") card. See [Remark 2](#).

EXSN	EXSB	EXSS	LCSN	LCSB	LCSS	GFAD	SCLMRR
------	------	------	------	------	------	------	--------

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	CON_ID	PRUL	AREAEQ		DGTYP	MOARFL		
Type	F	I	I		I	I		
Default	0	0	0		0	0		

VARIABLE	DESCRIPTION
CON_ID	Connection ID, referenced on *MAT_SPOTWELD_DAIMLER-CHRYSLER. Multiple sets of connection data may be used by assigning different connection IDs.
PRUL	The failure rule number for this connection: EQ.1: Use data of weld partner with lower RANK (default). GE.2: Use *DEFINE_FUNCTION expressions to determine weld data depending on several values of both weld partners. Variables DSIGY, DETAN, DDGPR, DSN, DSB, DSS, DEXSN, DEXSB, DEXSS, and DGFAD must be defined as function IDs, see Remark 5 .
AREAEQ	Area equation number for the connection area calculation: EQ.0: (default) $\text{Area}_{\text{true}} = \text{Area}_{\text{modelled}}$ EQ.1: Millimeter form; see Remark 4 EQ.-1: Meter form; see Remark 4
DGTYP	Damage type: EQ.0: No damage function is used. EQ.1: Strain based damage EQ.2: Failure function based damage EQ.3 or 4: Fading energy based damage (see Remark 4) EQ.5: Improved version of DGTYP = 4 (see Remark 4)
MOARFL	Modeled area flag: EQ.0: $\text{Area}_{\text{modelled}}$ goes down with shear (default).

DEFINE**DEFINE_CONNECTION_PROPERTIES**

VARIABLE		DESCRIPTION						
EQ.1: Area _{modelled} stays constant.								
Card 2	1	2	3	4	5	6	7	8
Variable		DSIGY	DETAN	DDGPR	DRANK	DSN	DSB	DSS
Type		F	F	F	F	F	F	F
Default		none	none	10 ¹⁰	none	none	none	none

VARIABLE		DESCRIPTION						
DSIGY		Default yield stress for the spot weld element: GT.0: Constant value LT.0: DSIGY references a yield curve or table; see Remark 6 .						
DETAN		Default tangent modulus for the spot weld element						
DDGPR		Default damage parameter for hyperbolic based damage function						
DRANK		Default rank value						
DSN		Default normal strength						
DSB		Default bending strength						
DSS		Default shear strength						

Card 3	1	2	3	4	5	6	7	8
Variable	DEXSN	DEXSB	DEXSS	DLCSN	DLCSB	DLCSS	DGFAD	DSCLMRR
Type	F	F	F	I	I	I	F	F
Default	1.0	1.0	1.0	0	0	0	none	1.0

DEFINE_CONNECTION_PROPERTIES**DEFINE**

VARIABLE	DESCRIPTION
DEXSN	Default exponent on normal stress term
DEXSB	Default exponent on bending stress term
DEXSS	Default exponent on shear stress term
DLCSN	Default curve ID for normal strength scale factor as a function of strain rate. If the first strain rate value in the curve is negative, it is assumed that all strain rate values are given as a natural logarithm of the strain rate.
DLCSB	Default curve ID for bending strength scale factor as a function of strain rate. If the first strain rate value in the curve is negative, it is assumed that all strain rate values are given as a natural logarithm of the strain rate.
DLCSS	Default curve ID for shear strength scale factor as a function of strain rate. If the first strain rate value in the curve is negative, it is assumed that all strain rate values are given as a natural logarithm of the strain rate.
DGFAD	Default fading energy for damage type 3 and type 4
DSCLMRR	Default scaling factor for torsional moment in failure function

Material Data Card 1.

Card 4	1	2	3	4	5	6	7	8
Variable	MID	SGIY	ETAN	DGPR	RANK	SN	SB	SS
Type	F	F	F	F	F	F	F	F
Default	none	none	none	10^{10}	none	none	none	none

VARIABLE	DESCRIPTION
MID	Material ID of the shell material for which properties are defined
SIGY	Yield stress to be used in the spot weld element calculation: GT.0: Constant value LT.0: SIGY references a yield curve or table; see Remark 6 .

DEFINE**DEFINE_CONNECTION_PROPERTIES**

VARIABLE	DESCRIPTION
ETAN	Tangent modulus to be used in the spot weld element calculation
DGPR	Damage parameter for hyperbolic based damage function
RANK	Rank value. See Remark 4 .
SN	Normal strength
SB	Bending strength
SS	Shear strength

Material Data Card 2.

Card 5	1	2	3	4	5	6	7	8
Variable	EXSN	EXSB	EXSS	LCSN	LCSB	LCSS	GFAD	SCLMRR
Type	F	F	F	I	I	I	F	F
Default	none	1.0						

VARIABLE	DESCRIPTION
EXSN	Exponent on normal stress term
EXSB	Exponent on bending stress term
EXSS	Exponent on shear stress term
LCSN	Curve ID for normal strength scale factor as a function of strain rate. If the first strain rate value in the curve is negative, it is assumed that all strain rate values are given as a natural logarithm of the strain rate.
LCSB	Curve ID for bending strength scale factor as a function of strain rate. If the first strain rate value in the curve is negative, it is assumed that all strain rate values are given as a natural logarithm of the strain rate.
LCSS	Curve ID for shear strength scale factor as a function of strain rate. If the first strain rate value in the curve is negative, it is assumed

VARIABLE	DESCRIPTION
	that all strain rate values are given as a natural logarithm of the strain rate.
GFAD	Fading energy for damage type 3 and 4
SCLMRR	Scaling factor for torsional moment in failure function

Remarks:

1. **Restriction to *MAT_SPOTWELD_DAIMLERCHRYSLER.** This keyword is used only with *MAT_SPOTWELD_DAIMLERCHRYSLER. The data input is used in a 3 parameter failure model. Each solid spot weld element connects shell elements that may have the same or different materials. The failure model assumes that failure of the spot weld depends on the properties of the welded materials, so this keyword allows shell material specific data to be input for the connection. The default data will be used for any spot weld connected to a shell material that does not have material specific data defined, so it is not necessary to define material specific data for all welded shell materials.
2. **ADD Option.** To simplify data input, the ADD keyword option allows material specific data to be added to an existing *DEFINE_CONNECTION_PROPERTIES table. To use the ADD option, omit Cards 2 and 3, and input only CON_ID on Card 1. Then use Cards 4 and 5 to input material specific data. For each unique CON_ID, control parameters and default values must be input in one set of *DEFINE_CONNECTION_PROPERTIES data. The same CON_ID may be used for any number of sets of material specific data input with the ADD option.
3. **The Three Parameter Failure Function.** The three parameter failure function is

$$f = \left(\frac{\sigma_n}{\sigma_n^F} \right)^{m_n} + \left(\frac{\sigma_b}{\sigma_b^F} \right)^{m_b} + \left(\frac{\tau}{\tau^F} \right)^{m_\tau} - 1 ,$$

where the three strength terms are SN, SB, and SS, and the three exponents are EXSN, EXSB, and EXSS. The strengths may be a function of strain rate by using the load curves, LCSN, LCSB, and LCSS. The peak stresses in the numerators are calculated from force resultants and simple beam theory.

$$\sigma_n = \frac{N_{rr}}{A}, \quad \sigma_b = \frac{\sqrt{M_{rs}^2 + M_{rt}^2}}{Z}, \quad \tau = SCLMRR \times \frac{M_{rr}}{2Z} + \frac{\sqrt{N_{rs}^2 + N_{rt}^2}}{A} ,$$

where the area is the cross section area of the weld element and Z is given by:

$$Z = \pi \frac{d^3}{32} ,$$

where d is the equivalent diameter of the solid spot weld element assuming a circular cross section.

4. **Control Parameters PRUL, AREAQ, And DGTYP.** There are three control parameters that define how the table data will be used for the connection, PRUL, AREAEQ, and DGTYP.

- PRUL.** PRUL determines how the parameters will be used. Because each weld connects two shell surfaces, one weld can have two sets of failure data as well as two values for ETAN and SIGY. For PRUL = 1 (default), a simple rule is implemented and the data with the lower RANK will be used. For PRUL = 2 or 3, function expressions can be used to determine the data based on several input values from both weld partners (see [Remark 5](#) for details).
- AREAEQ.** The second control parameter is AREAEQ which specifies a rule for calculating a true weld cross section area, A_{true} to be used in the failure function in place of the modeled solid element area, A . For AREAEQ = 1, A_{true} is calculated by

$$A_{\text{true}} = \frac{\pi}{4} (5\sqrt{t_{\min \text{ shell}}})^2 ,$$

where $t_{\min \text{ shell}}$ is the thickness of the welded shell surface that has the smaller thickness. For AREAEQ = -1, A_{true} is calculated by

$$A_{\text{true}} = \frac{\pi}{4} \left(\frac{5}{1000} \sqrt{1000 \times t_{\min \text{ shell}}} \right)^2 ,$$

The equation for AREAEQ = 1 is valid only for a length unit of millimeters, and AREAEQ = -1 is valid only for a length unit of meters.

- DGTYP.** The third control parameter, DGTYP, chooses from two available damage types. For DGTYP = 0, damage is turned off and the weld fails immediately when $f \geq 0$. For DGTYP > 0, damage is initiated when $f \geq 0$ and complete failure occurs when $\omega \geq 1$. For DGTYP = 1, damage growth is a function of plastic strain:

$$\omega = \frac{\varepsilon_{\text{eff}}^p - \varepsilon_{\text{failure}}^p}{\varepsilon_{\text{rupture}}^p - \varepsilon_{\text{failure}}^p} , \quad \varepsilon_{\text{failure}}^p \leq \varepsilon_{\text{eff}}^p \leq \varepsilon_{\text{rupture}}^p ,$$

where $\varepsilon_{\text{eff}}^p$ is the effective plastic strain in the weld material. When the value of the failure function first exceeds zero, the plastic strain at failure, $\varepsilon_{\text{failure}}^p$, is set to the current plastic strain, and the rupture strain is offset from the plastic strain at failure by

$$\varepsilon_{\text{rupture}}^p = \varepsilon_{\text{failure}}^p + \text{RS} - \text{EFAIL} ,$$

where RS and EFAIL are the rupture strain and plastic strain at failure which are input on the *MAT_SPOTWELD_DAIMLERCHRYSLER card. If failure occurs when the plastic strain is zero, the weld material yield stress is reduced to the current effective stress such that damage can progress.

For DGTYP = 2, damage is a function of the failure function, f :

$$f \geq 0 \Rightarrow \omega = \frac{f}{f_{\text{rupture}}} ,$$

where f_{rupture} is the value of the failure function at rupture which is defined by

$$f_{\text{rupture}} = \text{RS} - \text{EFAIL} ,$$

and RS and EFAIL are input on the *MAT_SPOTWELD_DAIMLERCHRYSLER card.

Because the DGTYP = 1 damage function is scaled by plastic strain, it will monotonically increase in time. The DGTYP = 2 damage function is forced to be a monotonically increasing function in time by using the maximum of the current value and the maximum previous value. For both DGTYP = 1 and DGTYP = 2, the stress scale factor is then calculated by

$$\hat{\sigma} = \frac{\text{DGPR} \times (1 - \omega)}{\omega \left(\frac{1}{2} + \sqrt{\frac{1}{4} + \text{DGPR}} \right) + \text{DGPR}} \sigma ,$$

This equation becomes nearly linear at the default value of DGPR which is 10^{10} .

For DGTYP = 3, damage is a function of total strain:

$$\omega = \frac{\Delta \varepsilon_n}{\Delta \varepsilon_{\text{fading}}} .$$

$\Delta \varepsilon_n$ is the accumulated total strain increment between moment of damage initiation (failure) and current time step t_n

$$\Delta \varepsilon_n = \Delta \varepsilon_{n-1} + \Delta t_n \sqrt{\frac{2}{3} \text{tr}(\dot{\varepsilon}_n \dot{\varepsilon}_n^T)} , \quad \Delta \varepsilon|_{t_{\text{failure}}} = 0 ,$$

and $\Delta \varepsilon_{\text{fading}}$ is the total strain increment for fading (reduction of stresses to zero)

$$\Delta \varepsilon_{\text{fading}} = \frac{2 \times \text{GFAD}}{\sigma_{\text{failure}}} ,$$

where GFAD is the fading energy from input and σ_{failure} is the effective stress at failure. The stress scale factor is then calculated by a linear equation

$$\widehat{\sigma} = (1 - \omega)\sigma ,$$

where σ is the Cauchy stress tensor at failure and ω is the actual damage value. Problems can occur, if the loading direction changes after the onset of failure, since during the damage process, the components of the stress tensor are kept constant and hence represent the stress state at failure.

DGTYP = 4 should be used when describing the damage behavior of the spotweld in a more realistic way. For DGTYP = 4, damage is a function of the internal work done by the spotweld after failure,

$$\widehat{\sigma} = (1 - \omega)\sigma^{\text{ep}}, \quad \omega = \frac{G_{\text{used}}}{2 \times \text{GFAD}}, \quad G_{\text{used}} = G_{\text{used}}^{n-1} + \det(\mathbf{F}) \sigma_{ij}^{\text{ep}} \Delta \varepsilon_{ij} .$$

In the above, \mathbf{F} is the deformation gradient. σ^{ep} is a scaled Cauchy stress tensor based on the undamaged Cauchy stress tensor σ^{wd} and scaled in such a way that the same internal work is done in the current time step as in the time step before (equipotential):

$$\sigma^{\text{ep}} = \alpha \sigma^{\text{wd}}, \quad \alpha = \frac{\sigma_{ij}^{n-1,\text{ep}} \Delta \varepsilon_{ij}}{\sigma_{ij}^{\text{wd}} \Delta \varepsilon_{ij}} .$$

DGTYP = 4 has two serious disadvantages that may result in unstable behavior after failure:

- i) If the spot weld is unloaded during damage, the undamaged Cauchy stress tensor tends to zero and therefore α to infinity.
- ii) The factor α is a very instable variable because it depends directly on the total strain increment.

DGTYP = 5 was developed to overcome these disadvantages. In contrast to DGTYP = 4, the undamaged Cauchy stress tensor is only calculated with an elastoplastic material model (not scaled by α). This limitation of the stress tensor makes it more stable during damage. At failure the yield stress is set to the current von Mises stress, and the plastic modulus is set to zero. Calculating the current damage value is the same as for DGTYP = 4. Note that, to achieve realistic behavior for the axial loading situation, you should use the UNIAXIAL keyword option of *MAT_SPOTWELD_DAIMLER. Additionally, with this material model, the spotweld can be deleted based on the current plastic strain with the input field RS.

5. **Failure Rule from *DEFINE_FUNCTION.** The failure rule number PRUL = 2 or 3, is available starting with Release R7. To use this new option, 11 fields must be defined as function IDs: DSIGY, DETAN, DDGPR, DSN, DSB, DSS, DEXSN, DEXSB, DEXSS, DGFAD, and DSCLMRR.

These functions depend on:

(t1, t2) = thicknesses of both weld partners
(sy1, sy2) = initial yield stresses at plastic strain
(sm1, sm2) = maximum engineering yield stresses
r = strain rate
a = spot weld area
fn = normal term in failure function
fb = bending term in failure function
fs = shear term in failure function
(ym1, ym2) = Young's moduli of both weld partners

For DSIGY = 100, such a function could look like:

```
*DEFINE_FUNCTION
 100
func(t1,t2,sy1,sy2,sm1,sm2,r,a,fn,fb,fs,ym1,ym2)=0.5*(sy1+sy2)
```

All the listed arguments in their correct order must be included in the argument list. For PRUL = 2, the thinner part is the first weld partner. For PRUL = 3, the bottom part (nodes 1-2-3-4) is the first weld partner. Since material parameters must be identified from both weld partners during initialization, this feature is only available for a subset of material models at the moment, namely material types 3, 24, 36, 81, 120, 123, 124, 133, 187, 224, 243, 251, and 258. This new option eliminates the need for the ADD option.

6. **Yield Curve or Table for SIGY.** When using this option, a simplified plasticity algorithm is used, assuming a linear behavior within one time increment. Thus, no iterative return mapping has to be performed.

*DEFINE

*DEFINE_CONSTRUCTION_STAGES

*DEFINE_CONSTRUCTION_STAGES

Purpose: Define times and durations of construction stages.

Card	1	2	3	4	5	6	7	8
Variable	ISTAGE	ATS	ATE	ATR	RTS	RTE	blank	IDYNAIN
Type	I	F	F	F	F	F		I
Default	none	0.0	0.0	none	ATS	ATE		0

VARIABLE	DESCRIPTION
ISTAGE	Stage ID
ATS	Analysis time at start of stage
ATE	Analysis time at end of stage
ATR	Analysis time duration of ramp
RTS	Real time at start of stage
RTE	Real time at end of stage
IDYNAIN	Flag to control output of dynain file at the end of the stage (see Remark 5): EQ.0: default to setting of IDYNAIN on *CONTROL-STAGED_CONSTRUCTION EQ.1: do not write dynain file.

Remarks:

1. **Related Keywords.** See also *CONTROL_STAGED_CONSTRUCTION and *DEFINE_STAGED_CONSTRUCTION_PART.
2. **Stage Start and End Times.** The first stage should start at time zero. There must be no gaps between stages, that is, ATS for each stage must be the same as ATE for the previous stage.
3. **Ramp Time.** The ramp time allows gravity loading and part removal to be applied gradually during the first time period ATR of the construction stage.

4. **Analysis and Real Time.** The analysis always runs in “analysis time” – typically measured in seconds. The “real time” is used only as a number to appear on output plots and graphs and is completely arbitrary. See ITIME on *CONTROL_STAGED_CONSTRUCTION.
5. **IDYNAIN.** If IDYNAIN is zero or blank, it will default to the same setting as IDYNAIN on *CONTROL_STAGED_CONSTRUCTION. The default of *CONTROL_STAGED_CONSTRUCTION is to write a dynain file at the end of each stage. This can be suppressed for individual stages by setting IDYNAIN to 1 on this keyword.

DEFINE**DEFINE_CONTACT_EXCLUSION*****DEFINE_CONTACT_EXCLUSION**

Purpose: Exclude tied nodes from being treated in specific contact interfaces. This keyword is currently only available in the MPP version.

Card 1	1	2	3	4	5	6	7	8
Variable	EID				TITLE			
Type	I				A70			

ID Card 1. This card sets the contact interface IDs of up to 7 tied interfaces.

Card 2	1	2	3	4	5	6	7	8
Variable	TARGET	C1	C2	C3	C4	C5	C6	C7
Type	I	I	I	I	I	I	I	I

Optional ID Cards. More tied interfaces. Include as many cards as necessary. This input ends at the next keyword (*) card.

Card 3	1	2	3	4	5	6	7	8
Variable	C8	C9	C10	C11	C12	C13	C14	C15
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
EID	Exclusion ID
TITLE	Exclusion Title
TARGET	Contact interface from which tied nodes are to be excluded. This must be the ID of a SINGLE_SURFACE, NODE_TO_SURFACE, or SURFACE_TO_SURFACE contact with SOFT ≠ 2.
C _i	The IDs of TIED contacts: 7 on the first card and 8 per additional card for as many cards as necessary. Any node which is a tracked node in one of these interfaces, and is in fact tied, will not be processed (as a tracked node) in the TARGET interface.

Remarks:

1. **Contact Forces on Excluded Nodes.** If a node is excluded from the Target by this mechanism, contact forces may still be applied to the node due to any tracked or reference nodes impacting the contact segments of which it is a part (no contact SEGMENTS are deleted, only contact NODES).
2. **SURFACE_TO_SURFACE and SINGLE_SURFACE Targets.** If you intend to exclude tied tracked nodes completely from other contacts in the model, this keyword can be used to do so for NODES_TO_SURFACE and ONE_WAY_SURFACE_TO_SURFACE contacts only. For (TWO_WAY) SURFACE_TO_SURFACE and SINGLE_SURFACE contacts, such nodes should be excluded from the contact definitions if so desired.

*DEFINE

*DEFINE_CONTACT_VOLUME

*DEFINE_CONTACT_VOLUME

Purpose: Define a rectangular, a cylindrical, or a spherical volume in a local coordinate system. The volume can be referenced by *SET_NODE_GENERAL for the purpose of defining a node set consisting of nodes inside the volume, or by *CONTACT_... for the purpose of defining nodes or segments on the SURFA side or the SURFB side of the contact (see SABOXID and SBBOXID on Card 1 of *CONTACT_...).

Card Summary:

Card 1. This card is required.

CVID	CID	TYPE	XC	YC	ZC		
------	-----	------	----	----	----	--	--

Card 2a. This card is included if and only TYPE = 0.

XMN	XMX	YMN	YMX	ZMN	ZMX		
-----	-----	-----	-----	-----	-----	--	--

Card 2b. This card is included if and only if TYPE = 1.

LENGTH	RINNER	ROUTER	D_ANGC				
--------	--------	--------	--------	--	--	--	--

Card 2c. This card is included if and only if TYPE = 2.

RINNER	ROUTER	D_ANGS					
--------	--------	--------	--	--	--	--	--

Data Cards:

Card 1	1	2	3	4	5	6	7	8
Variable	CVID	CID	TYPE	XC	YC	ZC		
Type	I	I	I	F	F	F		
Default	0	0	0	0.	0.	0.		

VARIABLE	DESCRIPTION
CVID	Contact volume ID
CID	Coordinate system ID. Required for rectangular and cylindrical volumes.

VARIABLE	DESCRIPTION
TYPE	Volume type: EQ.0: Rectangle EQ.1: Cylinder EQ.2: Sphere
XC	x -coordinate which defines the origin of coordinate system or the center of the sphere (TYPE= 3) in the global coordinate system.
YC	y -coordinate which defines the origin of coordinate system or the center of the sphere (TYPE = 3) in the global coordinate system.
ZC	z -coordinate which defines the origin of coordinate system or the center of the sphere (TYPE = 3) in the global coordinate system.

Rectangular Prism. Use when TYPE = 0.

Card 2a	1	2	3	4	5	6	7	8
Variable	XMN	XMX	YMN	YMX	ZMN	ZMX		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	0.0	0.0	0.0		

VARIABLE	DESCRIPTION
XMN	Minimum x -coordinate in local coordinate system
XMX	Maximum x -coordinate in local coordinate system
YMN	Minimum y -coordinate in local coordinate system
YMX	Maximum y -coordinate in local coordinate system
ZMN	Minimum z -coordinate in local coordinate system
ZMX	Maximum z -coordinate in local coordinate system

DEFINE**DEFINE_CONTACT_VOLUME**

Cylinder. Use when TYPE = 1.

Card 2b	1	2	3	4	5	6	7	8
Variable	LENGTH	RINNER	ROUTER	D_ANGC				
Type	F	F	F	F				
Default	0.0	0.0	0.0	0.0				

VARIABLE	DESCRIPTION
LENGTH	Length of cylinder originating at (XC,YC,ZC) and revolving around the local <i>x</i> -axis.
RINNER	Inner radius of cylinder
ROUTER	Outer radius of cylinder
D_ANGC	If the included angle between the axis of the cylinder and the normal vector to the contact segment is <i>less</i> than this angle, the segment is deleted.

Sphere. Use when TYPE = 2.

Card 2c	1	2	3	4	5	6	7	8
Variable	RINNER	ROUTER	D_ANGS					
Type	F	F	F					
Default	0.0	0.0	0.0					

VARIABLE	DESCRIPTION
RINNER	Inner radius of sphere
ROUTER	Outer radius of sphere
D_ANGS	If the included angle between a line draw from the center of the sphere to the centroid of the segment and the normal vector to the contact segment is <i>greater</i> than this angle, the segment is deleted.

***DEFINE_CONTROL_VOLUME**

Purpose: Specify an incompressible control volume. This control volume only works with the implicit solver.

Card 1	1	2	3	4	5	6	7	8
Variable	CVID	SSID	RHO	LCSRC				
Type	I	I	F	I				
Default	none	none	0.0	0				

VARIABLE	DESCRIPTION
CVID	Control volume ID
SSID	Segment set ID giving the control volume geometry. See Remark 2 .
RHO	Density of fluid contained in the domain (and entering the domain through the source curve). See Remarks 1 and 5 .
LCSRC	Load curve specifying the mass flow rate (mass/time) into the domain. Note that RHO must be set to a nonzero number for this to work. See Remarks 1 and 5 .

Remarks:

- Definition of a control volume.** A *control volume* is a domain in space that contains an incompressible fluid. The volume of the fluid contained within the domain exclusively determines the domain's volume. Thus, the domain's volume can *only* change if fluid flows into or out of the domain. The flow can be between control volumes or between a control volume and the ambient environment. In both cases, fluid flows through *flow areas*. The flow into a domain can also be defined by a source term determined by combining RHO and LCSRC. [Figure 17-7](#) illustrates two control volumes with one flow area between the control volumes and two flow areas to the ambient environment.
- Defining the control volume and flow areas.** For the volume calculation of a control volume to be correct, *the domain must be closed*. In other words, the segment set defining the volume must not contain any holes. So, either the segment set SSID in *DEFINE_CONTROL_VOLUME defines a closed volume, *or* this keyword is complemented by *DEFINE_CONTROL_VOLUME_FLOW_AREA

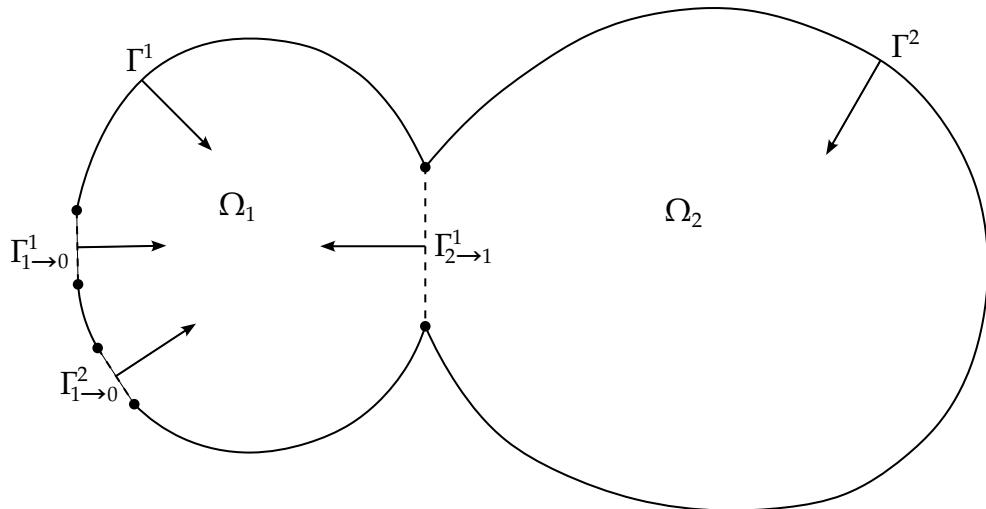


Figure 17-7. Two control volume with one sharing flow areas with the ambient environment. Control volume 1, Ω_1 , has boundary Γ_1 . It shares two flow areas, $\Gamma_{1 \rightarrow 0}^1$ and $\Gamma_{1 \rightarrow 0}^2$, with the ambient environment, and one flow area $\Gamma_{2 \rightarrow 1}^1$, with control volume 2, Ω_2 . Ω_2 has boundary Γ_2 and only shares flow areas with Ω_1 . See Remarks 1, 2, and 5.

keywords with *node sets* (`FASTYP = 1`) defining the perimeters of the holes that the defined segment set `SSID` doesn't cover. For the latter case, the volume is automatically closed by the segments that are generated by these complementary keyword definitions. If the flow area is defined with a segment set (`FASTYP = 2`) in [*DEFINE_CONTROL_VOLUME_FLOW_AREA](#), these segments should already be in the definition of `SSID`. They are *not* added to the set defining the control volume.

The segment set `SSID` *must* be oriented so that the normal of each segment points towards the interior of the domain as illustrated in [Figure 17-7](#). Furthermore, the segment sets of each flow area must be oriented so that their normal vectors point towards the interior of the *first* domain, CVID1, in the associated [*DEFINE_CONTROL_VOLUME_INTERACTION](#) keyword. If the flow area is defined by a node set (`FASTYP = 1`), the order of the nodes in the set must be clockwise when seen from *outside* the *first* control volume, CVID1. To interface with the ambient environment, leave the second domain, CVID2, blank. Whenever a flow area is defined, the pressure difference between the domains (or domain and ambient) does not result in nodal forces on the segment nodes in the flow area. Thus, the stiffness matrix is nonsymmetric whenever a flow area is defined, so using `LCPACK = 3` on [*CONTROL_IMPLICIT_SOLVER](#) may improve convergence.

3. **Flow between domains.** [*DEFINE_CONTROL_VOLUME_INTERACTION](#) determines the flow between domains or between a domain and the ambient environment. The function defined by `LCID` specifies the flow rate (volume/time) as a function of the time, pressure drop, and area of the flow area

opening. The curve may be left blank, causing no flow through the opening. The pressure difference used in functions is the one between the first and second domain, meaning $\Delta p = p_{\text{CVID1}} - p_{\text{CVID2}}$. The flow rate is positive into domain CVID2 and negative into domain CVID1.

4. **Output data.** Fluid cavity data, such as the time history of the volume and pressure, is output to the LSDA binout file.
5. **Mathematical formulation.** We define a control volume as Ω_i with boundary Γ_i for $i \in [1, n]$. The volume of a domain, $V_i = V_i(\mathbf{x})$, is obtained by Green's formula, which is a surface integral over Γ_i . Here, \mathbf{x} is the nodal coordinate vector, and it is implicitly understood that the volume calculation depends only on a subset S_i of all the nodes. We also assume that the fluid inside control volume Ω_i has a uniform pressure, p_i , that is assumed given for now. This pressure is relative to the ambient environment, not absolute.

We define flow areas as $\Gamma_{i \rightarrow j}^k$ with $i \in [1, n]$, $j \in [0, n - 1]$ where $j < i$, and $k \in [1, m_{ij}]$. i and j indicate the two domains or the domain and the ambient environment that share a flow area. $j = 0$ is the ambient environment. m_{ij} is the number of flow surfaces between two given domains or between a domain and the ambient environment. The area of a flow surface is $A_{i \rightarrow j}^k = A_{i \rightarrow j}^k(\mathbf{x})$ and depends only on a subset $S_{i \rightarrow j}^k$ of all the nodes. We have $\Gamma_{i \rightarrow j}^k \subset \Gamma_i \cap \Gamma_j$ for $j > 0$, meaning that this flow surface is shared between domains Ω_i and Ω_j .

The flow rate into domain Ω_i is defined as:

$$R_i = R_i^+ - \sum_{j=0}^{i-1} \sum_{k=1}^{m_{ij}} R_{i \rightarrow j}^k A_{i \rightarrow j}^k + \sum_{j=i+1}^n \sum_{k=1}^{m_{ji}} R_{j \rightarrow i}^k A_{j \rightarrow i}^k ,$$

where R_i^+ is a source term and $R_{i \rightarrow j}^k$ is the flow rate density from Ω_i to Ω_j through $\Gamma_{i \rightarrow j}^k$. We assume that the individual flow rate densities depend on pressure differences meaning $R_{i \rightarrow j}^k = R_{i \rightarrow j}^k(p_i - p_j)$. The volume increase of Ω_i can therefore be calculated as

$$\Delta V_i = \int_0^t R_i dt .$$

We now introduce the volume constraint, which in vector notation is written as:

$$\mathbf{c} = \mathbf{V} - \mathbf{V}_0 - \Delta \mathbf{V} = \mathbf{0} ,$$

where \mathbf{V}_0 are the volumes at time zero. A standard Lagrangian multiplier approach results in:

$$\mathbf{r}(\mathbf{x}) + \left(\frac{\partial \mathbf{c}}{\partial \mathbf{x}} \right)^T \mathbf{q} = \mathbf{0} ,$$

where \mathbf{q} is the vector of Lagrangian multipliers representing the pressure inside the domains. We may now set $\mathbf{p} = \mathbf{q}$ to complete the set of equations in unknowns \mathbf{x} and \mathbf{p} .

As discussed in [Remark 2](#), the actual implementation is slightly different in that the nodal force contributions do not involve the pressures exerted on flow areas. So, the last equilibrium equation is changed to

$$\mathbf{r}(\mathbf{x}) + \mathbf{N}^T \mathbf{p} = \mathbf{0},$$

where \mathbf{N} is a vector that accomplishes just that. Deriving the stiffness matrix, while neglecting second-order contributions, we end up with

$$\mathbf{K} = \begin{pmatrix} \frac{\partial \mathbf{r}}{\partial \mathbf{x}} & \mathbf{N}^T \\ \frac{\partial \mathbf{c}}{\partial \mathbf{x}} & \frac{\partial \mathbf{c}}{\partial \mathbf{p}} \end{pmatrix}.$$

It is nonsymmetric in the general case (flow areas exist).

***DEFINE_CONTROL_VOLUME_FLOW_AREA**

Purpose: Specify the flow area between two interacting control volumes (see [*DEFINE_CONTROL_VOLUME](#)).

Card 1	1	2	3	4	5	6	7	8
Variable	FAID	FCIID	FASID	FASTYP				
Type	I	I	I	I				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
FAID	Flow area ID
FCIID	Fluid cavity interaction ID referencing the *DEFINE_CONTROL_VOLUME_INTERACTION for which this area is used
FASID	Set ID giving the flow area
FASTYP	Type of set specifying flow area (see Remark 2): EQ.1: Node set giving the perimeter of the area. The flow area will be automatically meshed. EQ.2: Segment set covering the flow area

Remarks:

- Volume calculation of the control volumes.** The flow area segments are added to both control volumes automatically to close them for the volume calculation.
- Specifying flow area.** The flow area can be defined with either a segment set or a node set. For the segment set, the nodes forming the segments must be on the boundary of the flow area. Nodes placed inside the flow area will not move as the control volume deforms which may lead to poor accuracy in the flow area calculation. See [Remark 4](#).

Defining the area with a node set requires ordering the nodes giving the perimeter such that the normal of the area points inward toward the first control volume listed in the interaction and, thus, outward from the second. The

numbering, therefore, proceeds clockwise around a hole in the first control volume when viewed from the outside.

3. **Forces in flow area.** The segments or null shells defining the flow areas on the control volumes do not exert any force.
4. **Calculation of the flow area.** The flow area is calculated as the surface area of the segments defining it, and not as a projected area in the direction of some mean normal direction. For example, if the flow area is meshed about a node in the center for the flow area and the perimeter nodes translate relative to the motionless center node, the resulting mesh will define a cone. The flow area will be the surface area of the cone and not the circular cross-section area.

***DEFINE_CONTROL_VOLUME_INTERACTION**

Purpose: Specify the interaction between two control volumes (see [*DEFINE_CONTROL_VOLUME](#)).

Card 1	1	2	3	4	5	6	7	8
Variable	FCIID	CVID1	CVID2	LCID	AREA			
Type	I	I	I	I	F			
Default	none	none	none	none	optional			

VARIABLE	DESCRIPTION
FCIID	Fluid cavity interaction ID
CVID1	First control volume ID (see *DEFINE_CONTROL_VOLUME)
CVID2	Second control volume ID (see *DEFINE_CONTROL_VOLUME)
LCID	Load curve ID for a *DEFINE_FUNCTION or user subroutine specifying the pressure gradient-driven interaction between the connected cavities (see Remarks 1 and 2): GT.0: ID for *DEFINE_FUNCTION . The functions's input arguments are time, pressure difference, the area between the control volumes, the current time step size, and a flag for the start of a new time step. Currently, LS-DYNA passes the current time and cavity pressures as input arguments. LT.0: User subroutine <code>usr_icvflow</code> in <code>dyn21umat.F</code> determines the pressure gradient.
AREA	Constant area for the case when a flow area is not specified with *DEFINE_CONTROL_VOLUME_FLOW_AREA

Remarks:

1. **Function / user subroutine.** Specifying the load curve as a function with [*DEFINE_FUNCTION](#) or as a user subroutine gives the freedom to define various (nonlinear/time-dependent) lumped parameter models, that is, boundary conditions between the cavities.

2. **Area of valve opening.** We assume that the area/size of the valve opening plays no role in the defined interaction for the curve.

***DEFINE_COORDINATE_NODES**

Purpose: Define a local coordinate system with three node numbers. The local cartesian coordinate system is defined in the following steps. If the primary direction is along the x -axis, then the z -axis is computed from the cross product of x and \bar{y} , $z = x \times \bar{y}$ (see [Figure 17-8](#)); the y -axis is, then, computed as $y = z \times x$. A similar procedure applies if the local axis is along the y or z axes.

Card	1	2	3	4	5	6	7	8
Variable	CID	N1	N2	N3	FLAG	DIR		
Type	I	I	I	I	I	A		
Default	none	none	none	none	0	X		

VARIABLE	DESCRIPTION
CID	Coordinate system ID. A unique number has to be defined.
N1	ID of node located at local origin.
N2	ID of node located along local x -axis if DIR = X, the y -axis if DIR = Y, and along the z -axis if DIR = Z.
N3	ID of node located in local x - y plane if DIR = X, the local y - z plane if DIR = Y, and the local z - x plane if DIR = Z.
FLAG	Set to unity, 1, if the local system is to be updated each time step. Generally, this option when used with nodal SPC's is <i>not recommended</i> since it can cause excursions in the energy balance because the constraint forces at the node may go through a displacement if the node is partially constrained.
DIR	Axis defined by node N2 moving from the origin node N1. The default direction is the x -axis.

Remarks:

The nodes N1, N2, and N3 must be separated by a reasonable distance and not collinear to avoid numerical inaccuracies.

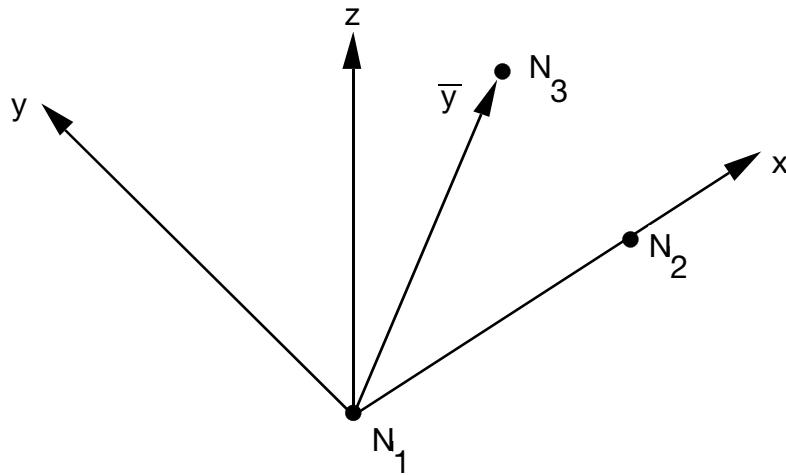


Figure 17-8. Definition of local coordinate system using three nodes when the node N₂ lies along the x -axis.

***DEFINE_COORDINATE_SYSTEM_{OPTION}**

Available options include:

<BLANK>

IGES

Purpose: Define a local coordinate system.

This card implements the same method as *DEFINE_COORDINATE_NODES; but, instead of reading coordinate positions from nodal IDs, it directly reads the three coordinates from its data cards as Cartesian triples.

When the IGES option is active, LS-DYNA will generate the coordinate system from an IGES file containing three straight curves representing the x -, y -, and z -axes. See [Remark 4](#).

Card Summary:

Card 1. This card is included if and only if the keyword option is unset (<BLANK>).

CID	X0	Y0	Z0	XL	YL	ZL	CIDL

Card 2. This card is included if and only if the keyword option is unset (<BLANK>).

XP	YP	ZP					

Card 3. This card is included if and only if the IGES keyword option is used.

FILENAME

Data Card Definitions:**Card 1 for <BLANK> Keyword Option.**

Card 1	1	2	3	4	5	6	7	8
Variable	CID	X0	Y0	Z0	XL	YL	ZL	CIDL
Type	I	F	F	F	F	F	F	I
Default	0	0.0	0.0	0.0	0.0	0.0	0.0	0

DEFINE**DEFINE_COORDINATE_SYSTEM**

VARIABLE	DESCRIPTION
CID	Coordinate system ID. A unique number must be defined.
XO	X-coordinate of origin.
YO	Y-coordinate of origin.
ZO	Z-coordinate of origin.
XL	X-coordinate of point on local <i>x</i> -axis.
YL	Y-coordinate of point on local <i>x</i> -axis.
ZL	Z-coordinate of point on local <i>x</i> -axis.
CIDL	Coordinate system ID applied to the coordinates used to define the current system. The coordinates X0, Y0, Z0, XL, YL, ZL, XP, YP, and ZP are defined with respect to the coordinate system CIDL.

Card 2 for <BLANK> Keyword Option.

Card 2	1	2	3	4	5	6	7	8
Variable	XP	YP	ZP					
Type	F	F	F					
Default	0.0	0.0	0.0					

VARIABLE	DESCRIPTION
XP	X-coordinate of point in local <i>xy</i> -plane.
YP	Y-coordinate of point in local <i>xy</i> -plane.
ZP	Z-coordinate of point in local <i>xy</i> -plane.

Card 1 for IGES Keyword Option.

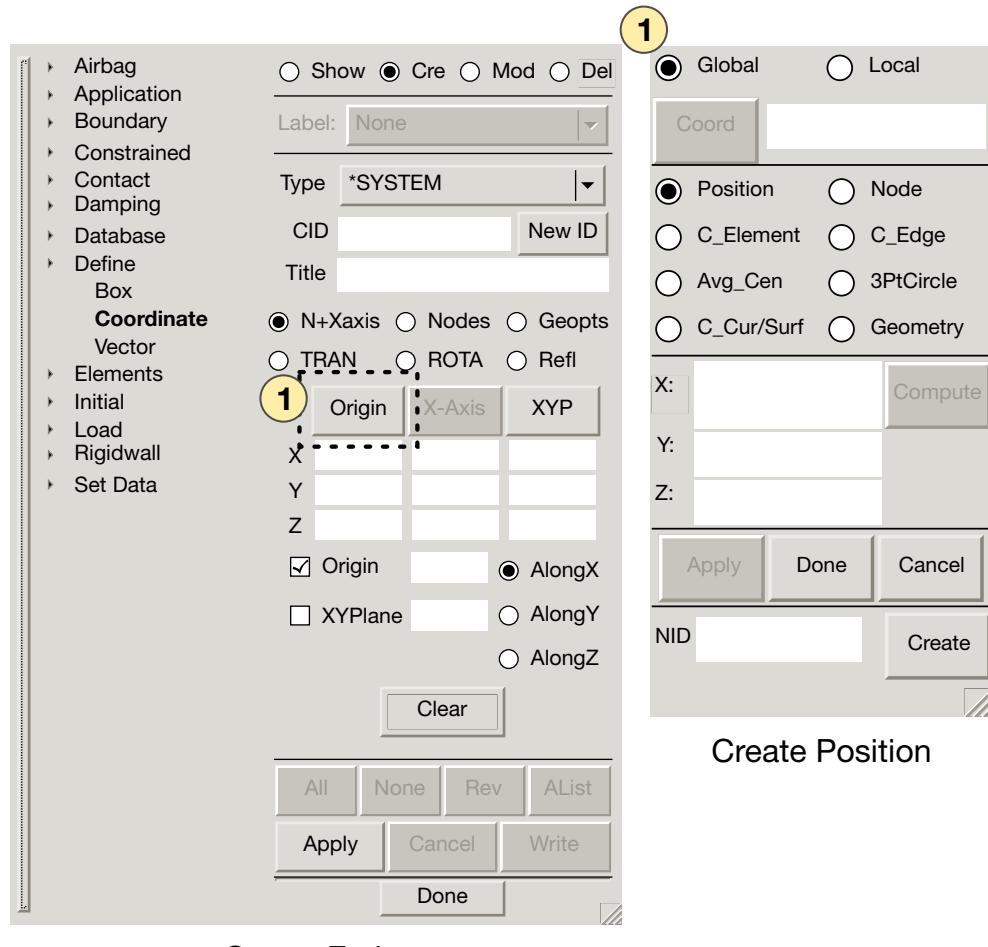
Card 3	1	2	3	4	5	6	7	8
Variable					FILENAME			
Type					C			
Default					none			

VARIABLE	DESCRIPTION
FILENAME	Name of the IGES file containing three curves (see Remark 4 below).

Remarks:

1. **Avoiding Numerical Inaccuracies.** The coordinates of the points must be separated by a reasonable distance and not co-linear to avoid numerical inaccuracies.
2. **Chains of Coordinate Transformations.** Care must be taken to avoid chains of coordinate transformations because there is no guarantee that they will be executed in the correct order.
3. **LS-PrePost.** A coordinate system can be created using the dialog box located at *Model* (main window) → *CreEnt* → *Define* (see the left pane) → *Coordinate*. This will activate a *Define Coordinate* dialog in the right pane. Select the *Cre* radio button at the top of the right pane and set the *type* dropdown to **SYSTEM*. The next set of radio buttons (below the title input box) sets the method used to define the coordinate system. See [Figure 17-9](#).
 - a) The *N+Xaxis* method generates a coordinate system from based on:
 - i) a user specified origin,
 - ii) one of the three global axes (this is a *severe* restriction), and
 - iii) a 3rd point.

The 3rd point, together with the specified global axis defines the new system's *xy*-plane. The remaining axes are derived using orthogonality and right-handedness. This method requires the user to pick two points which



Create Entity

Create Position

Figure 17-9. *LS-PrePost4.0* Dialog for defining a coordinate system.

involves the *Create Position* dialog box, as shown in the left frame of Figure 17-9.

NOTE: After defining each point in the *Create Position* dialog, it is *very important* to use the *done* button. The *Create Entity* dialog stays up and remains interactive while the *Create Position* dialog is also up and interactive. This can be confusing. Returning to the *Create Entity* dialog without choosing *done* is a common mistake.

- b) The *node* method generates a coordinate system from three points:
 - i) The first point specifies the origin.
 - ii) The first and second points together specify the *x*-axis.
 - iii) The three points together specify the *xy*-plane of the new coordinate system. The *y*- and *z*-axis are derived from orthogonality and right-handedness.

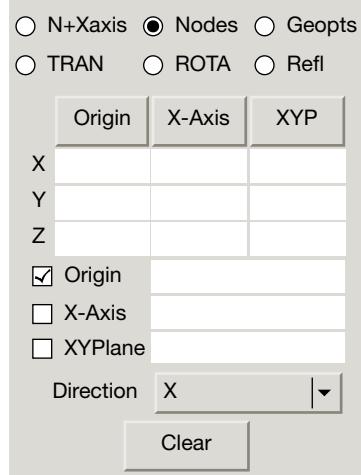


Figure 17-10. Subset of *Create Entity* dialog for both *Nodes* and *Geopts* methods.

- c) The *Geopts* option generates the new coordinate system from a global axis and two points. With this method the new system's z-axis is set from the *Direction* drop-down. This new system's *xy*-plane is, then, orthogonal to the chosen direction. The remaining two points serve to define the origin and the *x*-axis (by projecting the second point). This option is useful for metal forming application, since, often times, only the *z*-axis is important while the *x*- and *y*-axes are not.
- 4. **IGES.** When option, IGES, is used, three curves in the IGES format will be used to define a local coordinate system. IGES curve entity types 126, 110 and 106 are currently supported. Among the three curves, the longest length will be made as local *Z*-axis, the mid-length will be *Y*-axis and the shortest length *X*-axis. Suggested *X*-, *Y*- and *Z*-axis lengths are 100 mm, 200 mm and 300 mm, respectively.

All three curves must have one identical point and will be used for the origin of the new local coordinate system. The coordinate system ID for the local system will be based on the IGES file name. The IGES file name must start with a number, followed by an underscore "_", or by a dot. The number preceding the file name will be used as the new local coordinate system ID, which can then be referenced in *MAT_20 cards, for example.

After the LS-DYNA run, three beam elements of a new PID will be created in place of the three curves representing the local *X*-, *Y*-, and *Z*-axes in the d3plot file for viewing in LS-PrePost. See [Figure 17-11](#).

The following partial input contains an example in which the keyword is used to create a local coordinate system (CID = 25) from IGES input. The IGES file named, 25_iges, contains three intersecting curves in one of the three supported IGES entity types. The example demonstrates using the IGES coordinate system (ID = 25) to specify the local coordinate system for a rigid body (PID = 2,

*DEFINE

*DEFINE_COORDINATE_SYSTEM

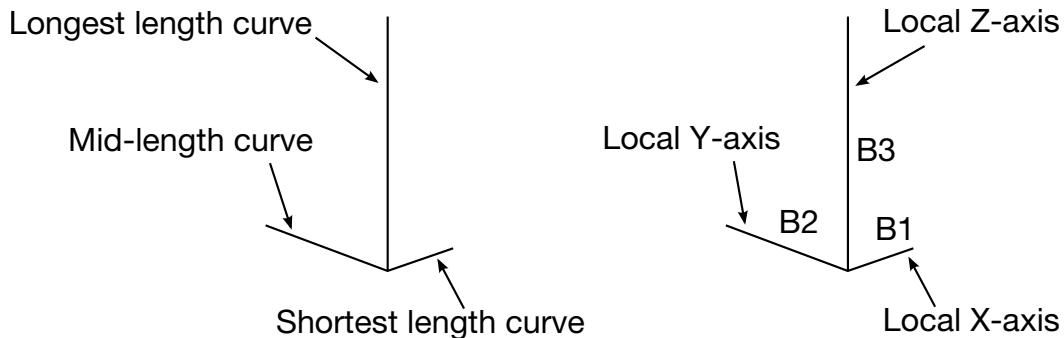


Figure 17-11. Input curves (left). The generated local coordinate system is written to the d3plot file as a part consisting of three beams (right).

MID = 2). The keyword, *BOUNDARY_PRESCRIBED_MOTION_RIGID_LOCAL, then uses this local coordinate system to assign velocities from load curves 3 and 5 for the rigid body motion in the local *x*-direction.

```
*KEYWORD
*DEFINE_COORDINATE_SYSTEM_IGES_TITLE
Flanging OP25
25_iges
$-----1-----2-----3-----4-----5-----6-----7-----8
*PART
punch
      2          2          2
*MAT_RIGID
$     MID       RO       E       PR           N   COUPLE      M   ALIAS
$     2 7.830E-09 2.070E+05      0.28
$     CMO      CON1      CON2
      -1        25    011111
$LCO or A1      A2      A3      V1      V2      V3
25
$-----1-----2-----3-----4-----5-----6-----7-----8
*BOUNDARY_PRESCRIBED_MOTION_RIGID_LOCAL
$   typeID      DOF      VAD      LCID      SF      VID      DEATH      BIRTH
      2          1          0          3      -1.0      0    0.00241      0.0
      2          1          0          5      -1.0      0    0.0115243  0.00241
```

The keyword can be repeated for each new coordinate system if multiple coordinate systems are needed.

Revision information:

This option is available starting in LS-DYNA Revision 62798.

***DEFINE_COORDINATE_VECTOR**

Purpose: Define a local coordinate system with two vectors, see [Figure 17-12](#). The vector cross product, $\mathbf{z} = \mathbf{x} \times \mathbf{v}_{xy}$ where \mathbf{v}_{xy} is a vector in the xy -plane, determines the z -axis. The y -axis is then given by $\mathbf{y} = \mathbf{z} \times \mathbf{x}$. If this coordinate system is assigned to a nodal point, then at each time step during the calculation, the coordinate system is incrementally rotated using the angular velocity of the nodal point to which it is assigned.

Card	1	2	3	4	5	6	7	8
Variable	CID	XX	YX	ZX	XV	YV	ZV	NID
Type	I	F	F	F	F	F	F	I
Default	0	0.0	0.0	0.0	0.0	0.0	0.0	0.

VARIABLE	DESCRIPTION
CID	Coordinate system ID. A unique number has to be defined.
XX	X-coordinate on local x -axis. Origin lies at (0,0,0).
YX	Y -coordinate on local x -axis
ZX	Z -coordinate on local x -axis
XV	X-coordinate of local xy -vector
YV	Y -coordinate of local xy -vector
ZV	Z -coordinate of local xy -vector
NID	Optional nodal point ID. The coordinate system rotates with the rotation of this node. If the node is not defined, the coordinate system is stationary. See Remark 2 .

Remarks:

- Numerical Inaccuracies.** These vectors should be separated by a reasonable included angle to avoid numerical inaccuracies.
- Rotation.** Ideally, this nodal point should be attached to a rigid body or a structural part where the nodal point angular velocities are meaningful. It should be

noted that angular velocities of nodes may not be meaningful if the nodal point is attached only to solid elements and even to shell elements where the drilling degree of freedom may be singular, which is likely in flat geometries.

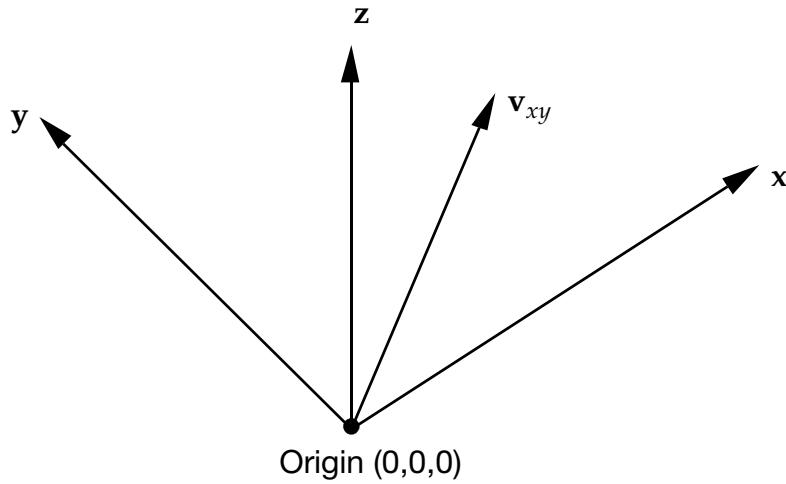


Figure 17-12. Definition of the coordinate system with two vectors.

DEFINE_CPG_GAS_PROPERTIES**DEFINE*****DEFINE_CPG_GAS_PROPERTIES**

Purpose: Define extended gas thermodynamic properties for the CPG solver.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	XMM	CP0	CP1	CP2	CP3	CP4	
Type	I	F	F	F	F	F	F	
Default	none	none	0.	0.	0.	0.	0.	

This card is optional.

Card 2	1	2	3	4	5	6	7	8
Variable	MU							
Type	F							
Default	0.							

This card is optional. If using this card, Card 2 must also be included, but it can be a blank line.

Card 3	1	2	3	4	5	6	7	8
Variable	TC							
Type	F							
Default	0.							

VARIABLE**DESCRIPTION**

ID Unique ID for this card

XMM Molar mass

DEFINE**DEFINE_CPG_GAS_PROPERTIES**

VARIABLE	DESCRIPTION
CP0, ..., CP4	Coefficients of temperature-dependent specific heat with constant pressure $C_p(T) = C_{p0} + C_{p1}T + C_{p2}T^2 + C_{p3}T^3 + C_{p4}T^4$
MU	Dynamic viscosity associated with the gas. See Remark 1 .
TC	Thermal conductivity of the gas. See Remark 2 .

Remarks:

1. **Mixture viscosity.** We use Carr's law to estimate gas-mixture viscosity composition:

$$\mu_{gm} = \frac{\sum_{i=1}^N y_i \mu_i \sqrt{M_i}}{\sum_{i=1}^N y_i}$$

with y_i the mole fraction of the i^{th} gas mixture component, μ_i the viscosity of the i^{th} component and M_i the molar mass of the i^{th} component.

To obtain a constant viscosity in the entire domain, define the same value for each gas.

Defining a viscosity is mandatory for the calculation of a friction force at the wall. See PFRIC in [*AIRBAG_CPG](#).

2. **Mixture thermal conductivity.** A simple mass fraction weighted average is done to estimate gas-mixture viscosity. In airbag applications, thermal diffusive effects are expected to be small.

DEFINE_CPG_REGION**DEFINE*****DEFINE_CPG_REGION**

Purpose: Define a geometrical region in the CPG domain for applying properties such as initial conditions (see [*INITIAL_CPG](#)).

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

Card 2	1	2	3	4	5	6	7	8
Variable	P1X	P1Y	P1Z	P2X	P2Y	P2Z	RAD	
Type	F	F	F	F	F	F	F	
Default	0.	0.	0.	0.	0.	0.	0.	

VARIABLE**DESCRIPTION**

DRID

Unique ID for this region

DRTYP

Region geometry type:

EQ.1: Domain divided by a section plane that acts as a cut-off.
The plane's normal vector points outwards from the region.

EQ.2: Box

EQ.3: Sphere

EQ.4: Cylinder

P1X/P1Y/P1Z/
P2X/P2Y/P2Z

Points with meaning that depends on DRTYP:

DRTYP.EQ.1: Plane normal and origin point

DRTYP.EQ.2: Minimum and maximum coordinates of the box

DEFINE**DEFINE_CPG_REGION**

VARIABLE	DESCRIPTION
	DRTYP.EQ.3: Origin of the sphere (second point not used) DRTYP.EQ.4: Points defining the center axis
RAD	Sphere or cylinder radius if DRTYP = 3 or 4, respectively

***DEFINE_CPM_BAG_INTERACTION**

Purpose: To model energy flow from a one airbag (the source) to another airbag (the sink). The source must be an active particle airbag and the sink a control volume (CV) airbag converted from a particle bag.

To track the flow of energy, LS-DYNA automatically determines which vent parts are common to both airbags. At each time step the energy that is vented through the common vents is subtracted from the source and added to the sink. In turn, the sink bag's pressure provides the downstream pressure value for the source bag's venting equation. While this model accounts for energy flow from source to sink, it ignores flow from sink to source.

If CHAMBER is used for the sink CV bag, see [Remark 1](#).

Card 1	1	2	3	4	5	6	7	8
Variable	BAGID1	BAGID2	NSPEC					
Type	I	I	I					
Default	none	none	0					

VARIABLE	DESCRIPTION
BAGID1	Airbag ID of source CPM particle bag
BAGID2	Airbag ID of sink CV bag switched from CPM bag
NSPEC	The location of the 1 st gas component from the CPM bag to be filled in the CV bag. See Remark 2 .

Remarks:

1. **Chambers.** Due to the complexity of the bookkeeping, the sink bag may have several chambers, but only one of the chambers may interact with the source bag. LS-DYNA finds this chamber automatically by looking at the commonly shared parts.
2. **NSPEC.** If NSPEC equals zero, the mass and energy vented out from the CPM bag will be directly fed into the CV bag. In this case the two bags are assumed to have the same gas species.

If the CPM bag and the CV bag contain different gas species initially, the gas species of the CPM bag must be listed after the inflator gas species of the CV bag (see NGAS on *AIRBAG_PARTICLE). The order of the gas components should be the same. NSPEC gives the first location of the CPM gas in the CV gas species list.

DEFINE_CPM_CHAMBER**DEFINE*****DEFINE_CPM_CHAMBER**

Purpose: To define airbag chambers for air particle initialization or chamber inter-action.

NOTE: The part set that specifies the airbag, SID1, for *AIRBAG_PARTICLE must include the parts that define the chambers, meaning SID1 for this keyword.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	NCHM						
Type	I	I						
Default	none	0						

Chamber Definition Card Sets:

Add NCHM chamber definition card sets. Each chamber definition card set consists of a Chamber Definition Card followed by NINTER Interaction Cards.

Chamber Definition Card.

Card 2	1	2	3	4	5	6	7	8
Variable	SID1	SID2	NINTER	CHM_ID				
Type	I	I	I	I				
Default	none	0	0	0				

*DEFINE

*DEFINE_CPM_CHAMBER

LS-DYNA keyword deck by LS-PrePost

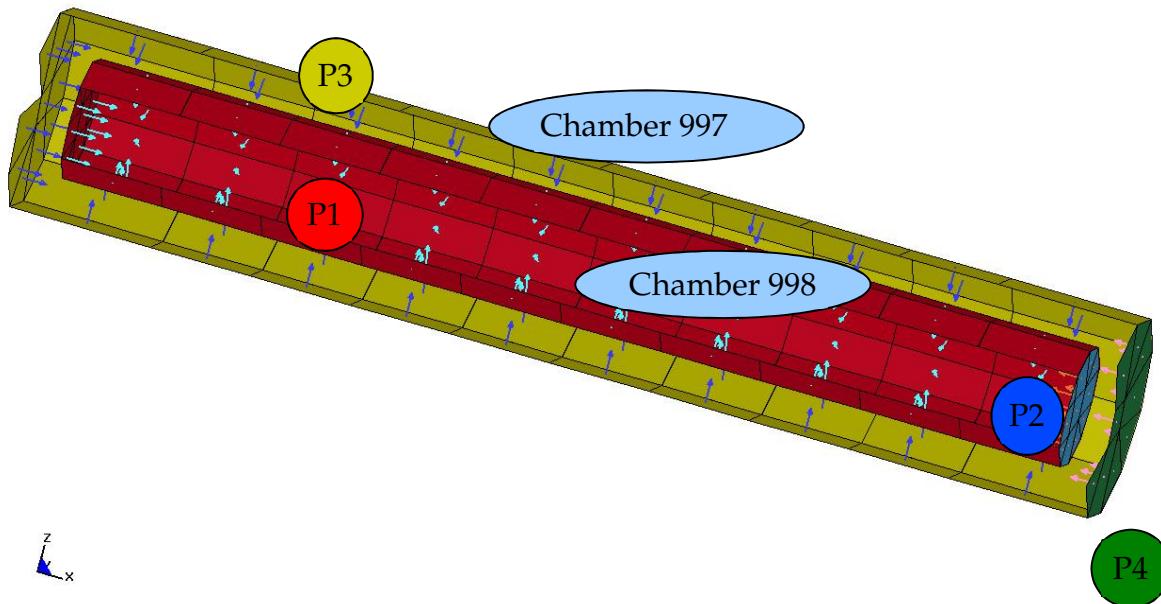


Figure 17-13. Illustration of the example in [Remark 1](#)

Interaction Cards. Add NINTER of these. If NINTER = 0, skip this card.

Card 3	1	2	3	4	5	6	7	8
Variable	SID3	ITYPE3	TOCHM					
Type	I	I	I					
Default	none	none	none					

VARIABLE	DESCRIPTION
ID	Unique ID for this card
NCHM	Number of chambers defined in this card
SID1	Part set defining all parts that constitute the chamber volume. See Remark 1 .
SID2	Part set defining the parts whose shell normal vectors need to be flipped, such as separation walls between chambers. See Remark 1 .
NINTER	Number of vent hole definition for chamber interaction.

VARIABLE	DESCRIPTION
CHM_ID	Chamber ID (see Remark 2)
SID3	Set defining interaction between chambers
ITYPE3	Set type: EQ.0: Part EQ.1: Part set
TOCHM	Chamber ID of the connected chamber

Remarks:

1. **Normal Vectors.** Each chamber's volume is calculated based on the part normals pointed inwards. So SID1 would normally have parts with their shell normals pointing inwards. But in some cases, parts may be shared by more than one chamber. In this case, the shell orientation of certain part(s) may need to be flipped for the other chambers in question. In such cases, SID2 can be used to flip the shell normals for specific parts. An example of this is given below and illustrated in [Figure 17-13](#).

```
*SET_PART_LIST
$#      sid
        1
$#      pid1      pid2      pid3      pid4
        1          2          3          4
*SET_PART_LIST
$#      sid
        20
$#      pid1      pid2
        1          2
*DEFINE_CPM_CHAMBER
$#      id      nchm
        1234      2
$#      sid1      sid2      ninter      chm_id
        20          0          1      998
$#      sid3      itype3      tochm
        2          0      997
$#      sid1      sid2      ninter      chm_id
        1          20          1      997
$#      sid3      itype3      tochm
        2          0      998
```

2. **Particle Collisions.** Particles with different chamber ID will not interact in particle to particle collision. This feature will allow program to distinguish particles separated by a thin wall.
3. **Output Files.** All chambers data are output to lsda binout database. The utility "l2a" can convert it into abstat_chamber ASCII file and process with LS-PrePost under abstat format.

DEFINE**DEFINE_CPM_GAS_PROPERTIES*****DEFINE_CPM_GAS_PROPERTIES**

Purpose: Define extended gas thermodynamic properties.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	XMM	CP0	CP1	CP2	CP3	CP4	
Type	I	F	F	F	F	F	F	
Default	none	none	0.	0.	0.	0.	0.	

This card is optional.

Card 2	1	2	3	4	5	6	7	8
Variable	MUTO	MUT1	MUT2	MUT3	MUT4	CHM_ID	VINI	
Type	F	F	F	F	F	I	F	
Default	0.	0.	0.	0.	0.	0	0.	

This card is optional. If using this card, Card 2 must also be included, but it can be a blank line.

Card 3	1	2	3	4	5	6	7	8
Variable	PID	PCUT	MCUT	TTIME				
Type	I	F	F	F				
Default	0	1e+20	-1e+20	0.				

VARIABLE**DESCRIPTION**

ID Unique ID for this card

XMM Molar mass

VARIABLE	DESCRIPTION
CP0, ..., CP4	Coefficients of temperature-dependent specific heat with constant pressure $C_p(T) = C_{p0} + C_{p1}T + C_{p2}T^2 + C_{p3}T^3 + C_{p4}T^4$
MUT0, ..., MUT4	Coefficients of temperature-dependent Joule-Thomson effect $\mu_t(T) = \mu_{t0} + \mu_{t1}T + \mu_{t2}T^2 + \mu_{t3}T^3 + \mu_{t4}T^4$
CHM_ID	Chamber ID (see Remark 1)
VINI	Initial volume for user-defined inflator (see Remark 1): EQ.0.0: User-defined inflator disabled GT.0.0: Initial volume LT.0.0: Calculate volume based on chamber geometry
PID	Part ID to get part pressure
PCUT	Cutoff pressure. If the part pressure found for PID is greater than PCUT, the mass flow rate of this gas is restricted.
MCUT	Cutoff mass flow rate. If the total mass flow rate excluding this gas is declining and below MCUT, the mass flow rate of this gas is restricted.
TTIME	Taper down duration. After satisfying one of the above cutoff conditions (MCUT or PCUT), a linear factor from 1.0 to 0.0 is applied to this mass flow rate. If TTIME equals zero, the flow from this gas is stopped after the cutoff.

Remarks:

1. **User-defined routines.** If you define CHM_ID and VINI, the `user_inflator` routine uses this gas property. We provide this routine in the user-defined subroutine Fortran files of the general `usermat` package. LS-DYNA gives the current chamber volume, pressure, temperature, and time step to the subroutine and expects a return value of change of chamber, burned gas temperature, and mass flow rate to feedback for releasing particles. LS-DYNA outputs all state data for this chamber to the `abstat_chamber` section of binout.

*DEFINE

*DEFINE_CPM_GAS_PROPERTIES

Example:

```
*AIRBAG_PARTICLE
$====1=====2=====3=====4=====5=====6=====7=====8=====
 1010          1      1011          1          0        0.0       0.0        1
 100000        0        1      300.0     1.0e-04        1
 1            1        1
 61           0        1.0        0        0        1        0.0
 1.0E-04      300.0    -9900
 651          653    -9910
 3000001      1.0
$=====
*DEFINE_CPM_GAS_PROPERTIES
$====1=====2=====3=====4=====5=====6=====7=====8=====
 9900  2.897E-02  2.671E+01  7.466E-03-1.323E-06
 9910  4.0E-03      20.79
-610.63    -0.0926
```

DEFINE_CPM_NPDATA**DEFINE*****DEFINE_CPM_NPDATA**

Purpose: To define extended CPM's NPDATA parameters (see *AIRBAG_PARTICLE).

Card 1	1	2	3	4	5	6	7	8
Variable	ID	HCONV	PFRIC	SDFBLK	KP	INIP	CP	PSFDCF
Type	I	F	F	F	F	I	F	F
Default	none	none	none	1.0	none	none	none	1.0

This card is optional.

Card 2	1	2	3	4	5	6	7	8
Variable	HCINT	CDEXT						
Type	F	F						
Default	optional	0.0						

VARIABLE	DESCRIPTION
ID	Unique ID for this card (referred to by HCONV on *AIRBAG_PARTICLE)
HCONV	Convective heat transfer coefficient used to calculate heat loss from the airbag's external surface to the ambient. See *AIRBAG_HYBRID developments (Resp. P.O. Marklund). LT.0.0: HCONV is a load curve ID defining the heat convection coefficient as a function of time.
PFRIC	Friction factor, F_r , if $-1.0 < \text{PFRIC} \leq 1.0$. Defaults to FRIC from Card 1 on *AIRBAG_PARTICLE if undefined. Otherwise, LE.-1.0: PFRIC is the curve ID which defines F_r as a function of the part pressure. GT.1.0: PFRIC is the *DEFINE_FUNCTION ID that defines F_r .

DEFINE**DEFINE_CPM_NPDATA**

VARIABLE	DESCRIPTION
SDFBLK	Scaling down factor for blockage factor (default = 1.0, no scaling down). The valid factor will be (0.0,1.0]. If 0.0, it is set to 1.0.
KP	Thermal conductivity of the part
INIP	Place initial air particles on the surface: EQ.0: Yes (default) EQ.1: No This feature excludes surfaces from initial particle placement. This option helps prevent particles from being trapped between adjacent fabric layers.
CP	Specific heat
PSFDCF	Additional scale factor for force decay constant. See Remark 1 .
HCINT	Convective heat transfer coefficient between CPM particles and the part of interest. This feature is intended for analyses that couple the CPM and thermal solvers.
CDEXT	Drag coefficient of the external air for the part or part set. The CD_EXT field in *AIRBAG_PARTICLE must be nonzero. GT.0.0: Drag coefficient LT.0.0: Curve ID for the time-dependent drag coefficient

Remarks:

1. **Scale factor for force decay constant.** The particle impact force is gradually applied to an airbag segment by a particular smoothing function with the following form:

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

where τ is the force decay constant. PSFDCF scales τ :

$$\tau = \tau \times \text{PSFDCF} .$$

***DEFINE_CPM_SWITCH_REGION**

Purpose: Define spherical or cylindrical regions of interest from a set of segments where the particle-to-particle collision will be omitted over specified intervals.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	BAGID	SSID	NFREQ	OFF	R	HEIGHT	
Type	I	I	I	I	F	F	F	
Default	none	none	none	1.0	none	none	0.0	

VARIABLE	DESCRIPTION
ID	Unique ID for this option
BAGID	*AIRBAG_PARTICLE ID to apply this option
SSID	Segment set ID
NFREQ	Number of cycles to skip the particle-to-particle collision: GT.0: Number of cycles LT.0: Time-dependent load curve with ID NFREQ
OFF	Offset distance of the region's CG from the segment center along the normal direction
R	Radius of the region
HEIGHT	Height of the cylinder for cylindrical domains: EQ.0.0: Spherical domain GT.0.0: Height of the cylindrical domain

DEFINE**DEFINE_CPM_VENT*****DEFINE_CPM_VENT**

Purpose: To define extended vent hole options.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	C23	LCTC23	LCPC23	ENH_V	PPOP	C23UP	IOPT
Type	I	F	I	I	I	F	F	I
Default	none	1.0	none	none	↓	none	none	none

Card 2	1	2	3	4	5	6	7	8
Variable	JT	IDS1	IDS2	IOPT1	PID1	PID2	VANG	LCRED
Type	I	I	I	I	I	I	F	I
Default	0	none	none	none	none	none	0.	none

This card is optional.

Card 3	1	2	3	4	5	6	7	8
Variable	NID1	NID2	NID3	LCAC23	PSETPV	SFPV	LPATM	IBLKOFF
Type	I	I	I	I	I	F	I	I
Default	0	0	0	0	0	0.0	0	0

DEFINE_CPM_VENT**DEFINE**

This card is optional.

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

VARIABLE	DESCRIPTION
ID	Unique ID for this card
C23	Vent hole coefficient. This is the Wang-Nefske leakage parameter.
LCTC23	Load curve defining the vent hole coefficient as a function of time
LCPC23	Load curve or function defining the vent hole coefficient as a function of pressure: GT.0: Load curve ID for *DEFINE_CURVE_FUNCTION or *DEFINE_CURVE LT.0: *DEFINE_FUNCTION ID
ENH_V	Enhance venting option. The default is 0; however, if the Joule-Thomson effect is enabled, this field will be set to 1 automatically. EQ.0: Disable EQ.1: Enable
PPOP	Pressure difference between interior and ambient pressure to open the vent hole. Once the vent is open, it will stay open.
C23UP	Scale factor of C23 while switching from CPM to uniform pressure calculation.
IOPT	Directional venting: EQ.1: In shell normal EQ.2: Against shell normal One-way venting: EQ.10: In shell normal

DEFINE**DEFINE_CPM_VENT**

VARIABLE	DESCRIPTION
	EQ.20: Against shell normal Special vent option: EQ.100: Enable compression seal vent. Vent area is adjusted according to the formula below (see Remark 1). $A_{\text{vent}} = \max(A_{\text{current}} - A_0, 0)$
	EQ.200: Enable push-out vent. Particle remains active while going through this external vent within the range of 2 times of its characteristic length, l_{vent} . $l_{\text{vent}} = \sqrt{A_{\text{vent}}}$
JT	Include the Joule-Thomson effect. When the Joule-Thomson effect is enabled, ENH_V is automatically set to 1 (enable). EQ.0: Disable EQ.1: Use part pressure EQ.2: Use chamber pressure
IDS1	JT's up stream condition part ID/chamber ID
IDS2	JT's downstream condition part ID/chamber ID
IOPT1	Upstream chamber ID for one-way vent hole. This will help the code to determine the probability function.
PID1, PID2	PID1 and PID2 indicate parts for determining the local part pressures that can be used to evaluate the vent probability function. Depending on if a chamber is defined, how the local part pressures are evaluated changes (see *DEFINE_CPM_CHAMBER). PID1 and PID2 are optional if a chamber is defined, otherwise they are required input. When a chamber is defined, specifying PID1 and PID2 causes the vent probability function to be evaluated from the difference of local part pressures between PID1 and PID2. Otherwise, the calculation involves the chamber pressure. This option is usually used for vents near a long sleeve which causes unrealistic venting using chamber pressure alone.
	When a chamber is not defined, the vent probability function is evaluated from the difference of local part pressures between PID1 and PID2, using the location of the part centers to help determine

VARIABLE	DESCRIPTION
	vent direction. If the part is an external part, the part pressure will be used. If the part is an internal part, the pressure on the shell's positive normal side will be used. If the vent is an external vent, PID1 should be the same as PID2 to avoid input error.
VANG	Cone angle in degrees. Particle going through this vent will be redirected based on this angle. This option is only valid for an internal vent. GT.0.0: Cone angle (maximum 270°) EQ.0.0: Disabled (default) EQ.-1.0: Direction follows the vent normal (see Remark 2) EQ.-2.0: Direction follows local coordinates system defined by three nodes (see Remark 2)
LCRED	Time dependent probability curve to control CPM particle through the internal vent with VANG option (see Remark 3).
NID1-NID3	Three nodes define a moving coordinate system for the direction of flow through the vent when VANG = -2. (see Remark 4)
LCAC23	Load curve defining vent hole coefficient as a function of current vent area
PSETPV	PSETPV is a part set ID for internal airbag parts that interact with the push-out vent (IOPt = 200). The sign determines where the ambient pressure is applied: GT.0: Ambient pressure is applied to elements in these parts that are at least a distance of SFPV × CL away from the vent. LT.0: Ambient pressure is applied to elements in these parts that are within a distance of SFPV × CL of the vent. CL is defined as the $\sqrt{\text{element area}}$.
SFPV	Scale factor for element's characteristic length
LPATM	Load curve for ambient pressure of the external vent. This option only works for the CPM mode.
IBLKOFF	Flag for turning off blockage treatment. This flag only applies when IBLOCK is nonzero on *AIRBAG_PARTICLE. EQ.0: Use IBLOCK from *AIRBAG_PARTICLE (default). EQ.1: Turn off blockage treatment.

*DEFINE

*DEFINE_CPM_VENT

VARIABLE	DESCRIPTION
JTND	Node / node set for applying vent reaction force: GT.0: Node ID LT.0: Node Set ID. The average force is evenly applied among the nodes in the node set.

Remarks:

1. **Compression Seal Vent Model.** In order to evaluate bag state variables correctly, the CPM domain needs to be a closed surface for the volume to be well-defined. If the model contains a flap vent which is free to open and close, this option will correctly maintain the bag's integrity.
2. **VANG < 0.** When VANG < 0, all the particles passing through vent are in a collinear stream following the venting direction with no spread, effectively a zero degree angle. The particles disperse after the vent due to particle collisions.
3. **LCRED.** The application of the VANG option upon a particle passing through the vent at a given time depends on the time dependent probability function, LCRED, if defined. This option can be used to strengthen or weaken the directional vent stream.
4. **NID1-NID3.** If NID1, NID2, and NID3 are given, vectors \mathbf{v}_1 and \mathbf{v}_2 , given by NID2 – NID1 and NID3 – NID1, respectively, will form the local XY-plane. The jet direction is determined by the vector \mathbf{v}_3 which is given as the cross product of \mathbf{v}_1 with \mathbf{v}_2 , that is, $\mathbf{v}_1 \times \mathbf{v}_2$. If NID3 is zero, the jet direction is determined by the vector \mathbf{v}_3 which is given by NID1 – NID2.

Example:

```
*AIRBAG_PARTICLE
$====1=====2=====3=====4=====5=====6=====7=====8=====
   1010          1      1011          1          0        0.0       0.0       1
 100000          0          1      300.0     1.0e-04
           1          1          1
          61          0      -9910
 1.0E-04      300.0    2.897E-2   2.671E+1   7.466E-3  -1.323E-6
    1000      1001    4.0E-3      20.79
 3000001      1.0
$=====
*DEFINE_CPM_VENT
$====1=====2=====3=====4=====5=====6=====7=====8=====
 9910          1.0          0          0          1        0.0
           1          51          2
```

***DEFINE_CURVE_{OPTION}**

Purpose: Define a curve [for example, load (ordinate value) as a function of time (abscissa value)], often loosely referred to as a load curve. The ordinate may represent something other than a load however, as in the case of curves for constitutive models.

For most constitutive models, *DEFINE_CURVE curves are rediscretized internally with equal intervals along the abscissa for fast evaluation. Rediscretization is *not* used when evaluating loading conditions such as pressures, concentrated forces, or displacement boundary conditions (see [Remark 1](#) for more details).

The curve rediscretization algorithm was enhanced for the 2005 release of version 970. In certain cases the new load-curve routines changed the final results enough to disrupt benchmarks. For validated models, such as barriers and occupants, requiring numerical consistency, there are keyword options for reverting to the older algorithms.

Available options include:

<OPTION>

3858

5434a

which correspond to the first releases of version 970 and the 2005 release, respectively.

Since input errors and wrong results are sometimes related to load curve usage, a “*Load curve usage*” table is printed in the d3hsp file after all the input is read. This table should be checked to ensure that each curve ID is referenced by the option for which the curve is intended.

Card 1	1	2	3	4	5	6	7	8
Variable	LCID	SIDR	SFA	SFO	OFFA	OFFO	DATTYP	LCINT
Type	A	I	F	F	F	F	I	I
Default	none	0	1.	1.	0.	0.	0	0

*DEFINE

*DEFINE_CURVE

Point Cards. Put one pair of points per card (2E20.0). Input is terminated at the next keyword ("**") card.

Card 2	1	2	3	4	5	6	7	8
Variable	A1		01					
Type	E20.0		E20.0					
Default	0.0		0.0					

VARIABLE	DESCRIPTION
LCID	Load curve identification. A unique number or label not containing “.” must be specified. When defined by a number, tables (see *DEFINE_TABLE) and load curves may not share common IDs. If LCID is non-numeric, the LS-DYNA execution line should include “plabel = y” for proper (but unfortunately slower) input processing.
SIDR	Flag controlling use of curve during dynamic relaxation. SIDR set to 1 or 2 will activate a dynamic relaxation phase unless IDR-FLG = -999 in *CONTROL_DYNAMIC_RELAXATION . EQ.0: Load curve used in the normal analysis phase only or for other applications, EQ.1: Load curve used in the dynamic relaxation phase but not the normal analysis phase, EQ.2: Load curve applies to both the dynamic relaxation phase and normal analysis phase.
SFA	Scale factor for abscissa value. This is useful for simple modifications. EQ.0.0: Default set to 1.0.
SFO	Scale factor for ordinate value (function). This is useful for simple modifications. EQ.0.0: Default set to 1.0.
OFFA	Offset for abscissa values, see Remark 3 .
OFFO	Offset for ordinate values (function), see Remark 3 .

VARIABLE	DESCRIPTION
DATTYP	Data type. This affects how offsets are applied (see Remark 4). EQ.-100: For defining the proxy, α , from experiments for the chemical shrinkage coefficient as a function of temperature (see *MAT_ADD_CHEM_SHRINKAGE for details) EQ.-2: For fabric stress as a function of strain curves (*MAT_FABRIC) as described below. EQ.0: General case for time-dependent curves, force as a function displacement curves and stress-strain curves EQ.1: For general (x, y) data curves whose abscissa values do not increase monotonically EQ.6: For general (r, s) data (coordinates in a 2D parametric space) whose values do not increase monotonically. Use for definition of trimming polygons for trimmed NURBS (*ELEMENT_SHELL_NURBS_PATCH , NL > 0).
LCINT	The number of discretization points to use for this curve. If a model contains multiple curves, then the maximum value of all LCINTs is used to determine the memory requirement of each individual curve. Extremely large values should therefore be avoided if many curves are used. EQ.0: Use value of LCINT from *CONTROL SOLUTION .
A1, A2, ...	Abscissa values. See Remark 3 .
O1, O2, ...	Ordinate (function) values. See Remark 3 .

Remarks:

1. **Warning concerning rediscretization.** For constitutive models, unless noted otherwise in the material description, LS-DYNA internally rediscretizes the curve with uniform spacing to bypass searching during evaluations. The major drawback of this algorithm is that any detail in the curve on a scale finer than the uniform rediscretization grid will be smoothed-out and lost. *It is, therefore, important to avoid placing a single point off at some value approaching infinity.* The lone point at infinity will cause the resolution of the uniform grid to be coarse relative to the other points, causing the rediscretized curve to be, possibly, featureless.

Therefore, when defining curves for constitutive models, points should generally be spaced as uniformly as possible. Also, since the constitutive model curves are extrapolated, it is important to ensure that extrapolation does not lead to physically meaningless values, such as a negative flow stress. Conversely, extrapolation can be exploited to control the results of evaluations at points far from the input data.

2. The number of points in each rediscretized curve is controlled by the parameter LCINT in [*CONTROL_SOLUTION](#). By changing LCINT to a value greater than the default of 100, the rediscretized curves may better resemble the input curves. The data points of the rediscretized curves are written to messag and d3hsp if the parameter IPCURV is set to 1 in [*CONTROL_OUTPUT](#).
3. **Scaling.** The load curve values are scaled after the offsets are applied, that is:
$$\text{Abscissa value} = \text{SFA} \times (\text{Defined value} + \text{OFFA})$$
$$\text{Ordinate value} = \text{SFO} \times (\text{Defined value} + \text{OFFO})$$
4. **DATTYP.** The DATTYP field controls how the curve is processed during the calculation.
 - a) For DATTYP = 0 positive offsets may be used when the abscissa represents time, since two additional points are generated automatically at time zero and at time $0.999 \times \text{OFFA}$ with the function values set to zero.
 - b) If DATTYP = 1, then the offsets do not create these additional points. Negative offsets for the abscissa simply shift the abscissa values without creating additional points.
 - c) For material *MAT_FABRIC with FORM = 4, 14, -14, or 24, set DATYP = -2 to define stress as a function of strain curves using engineering stress and strain instead of 2nd Piola-Kirchhoff stress and Green strain.
 - d) For adding chemical shrinkage effects with *MAT_ADD_CHEM_SHRINKAGE, setting DATYP = -100 signals to LS-DYNA that the load curve defines the proxy for the chemical shrinkage coefficient as a function of temperature. LS-DYNA then internally converts this proxy to the chemical shrinkage coefficient.
5. **Context dependent extrapolation.** Load curves are not extrapolated by LS-DYNA for applied loads such as pressures, concentrated forces, displacement boundary conditions, etc. Function values are set to zero if the time, etc., goes off scale. Therefore, extreme care must be observed when defining load curves. In the constitutive models, extrapolation is employed if the values on the abscissa go off scale.

6. **Restart.** The curve offsets and scale factors are ignored during restarts if the curve is redefined. See *CHANGE_CURVE_DEFINITION in the restart section.

DEFINE**DEFINE_CURVE_BOX_ADAPTIVITY*****DEFINE_CURVE_BOX_ADAPTIVITY**

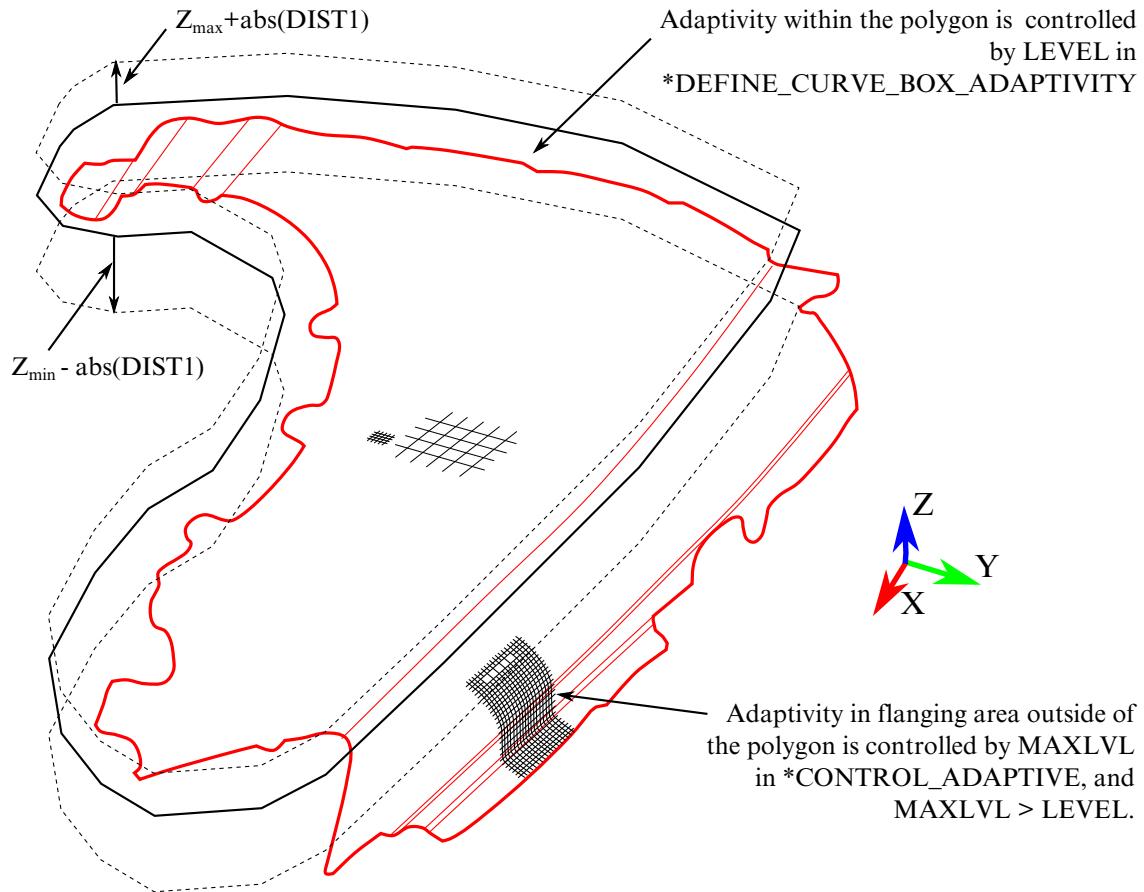
Purpose: To define a polygon adaptive box for sheet metal forming. This keyword is applicable to shell elements. This keyword is used with *CONTROL_ADAPTIVE. This keyword is similar to *DEFINE_BOX_ADAPTIVE. It is only available for SMP.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	PID	LEVEL	DIST1				
Type	I	I	I	F				
Default	none	none	none	none				

Point Cards. Include as many as necessary. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	X		Y		Z			
Type	F		F		F			
Default	none		none		none			

VARIABLE	DESCRIPTION
ID	Curve ID; must be unique. The curve must be closed: its first and last point <i>must</i> coincide. See Examples .
PID	Sheet blank Part ID, as in *PART.
LEVEL	Adaptive refinement levels, similar to the field MAXLVL in *CONTROL_ADAPTIVE. See Remark 1 .
DIST1	Depth in the Z-direction that the curve defined with Card 2 will be extruded. Currently this variable must be input as a negative value. The box depth in the Z-direction will be extended in the -Z-direction by $Z_{\min} - \text{DIST1} $ and in +Z-direction by $Z_{\max} + \text{DIST1} $, where Z_{\min} and Z_{\max} are the minimum and maximum Z coordinates in all the (X, Y, Z) data points input with Card 2. See Remark

**Figure 17-14.** Defining an adaptive polygon box

VARIABLE	DESCRIPTION
	3 and Figure 17-14 .
X	X coordinate of a point on the curve. See Remark 3 .
Y	Y coordinate of a point on the curve. See Remark 3 .
Z	Z coordinate of a point on the curve. See Remark 3 .

Remarks:

1. **Mesh Refinement Definition.** Within the polygon, the variable LEVEL has priority over MAXLVL in ***CONTROL_ADAPTIVE**, but is limited by the minimum element size controlled by ADPSIZE. A larger LEVEL than MAXLVL value will enable more mesh refinement within the polygon, up to the size defined by ADPSIZE, than outside of the box. However, mesh refinement when LEVEL > MAXLVL is not recommended. This keyword (and ***DEFINE_BOX_ADAPTIVE**) is intended for the MAXLVL > LEVEL case in which the polygon box excludes the local areas of interest. For this case, refinement inside the local

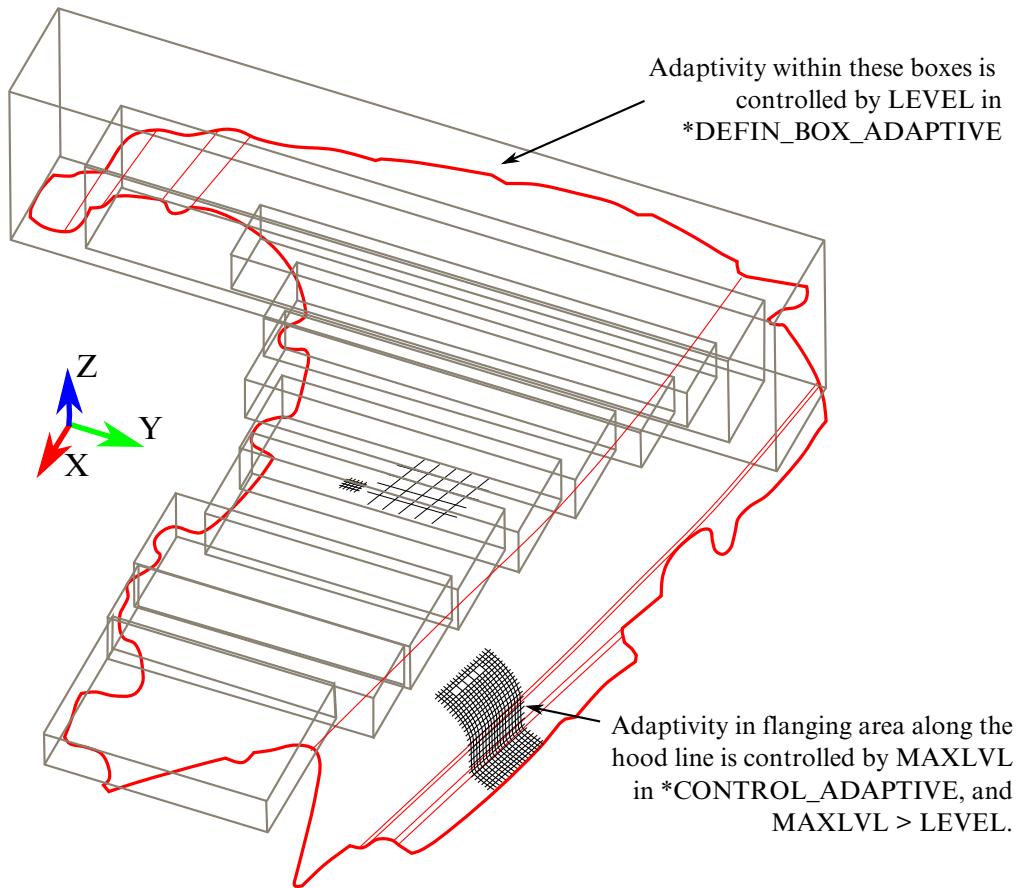


Figure 17-15. Defining adaptive boxes

areas will be controlled by MAXLVL while outside of the area will be controlled by LEVEL as shown in [Figure 17-14](#).

2. **Advantages.** With this keyword only one box needs to be defined to specify the selected region while *DEFINE_ADAPTIVE_BOX requires multiple boxes to be defined for the same region as shown in [Figure 17-15](#).
3. **Curve Location.** The 3D curve (closed polygon) defined by (X, Y, Z) data pairs should be near the sheet blank after the blank is auto-positioned in the beginning of a simulation. Similar to *DEFINE_BOX_ADAPTIVE, only the elements on the sheet blank initially within the polygon will be considered for use with this keyword. Local coordinate system is not supported at the moment.
4. **IGES Format.** The 3D curve can be converted from IGES format to the format required here following the procedure outlined in keyword *INTERFACE_BLANKSIZE.

Examples:

A partial keyword example is provided below, where inside the polygon mesh has no

refinement (LEVEL = 1), while outside of the box, the mesh is refined 5 levels (MAXLVL = 5). The final minimum element size is defined as 0.4. It is noted that the first point and last point of the polygon are the same, closing the polygon box.

```
*CONTROL_ADAPTERIVE
$ ADPFREQ      ADPTOL      ADPTYP      MAXLVL      TBIRTH      TDEATH      LCADP      IOFLAG
&adpfq1          5.0           2           5           0.0 1.000E+20           1
$ ADPSIZE      ADPASS      IREFLG      ADPENE      ADPTH      MEMORY      ORIENT      MAXEL
       0.4           1           0 &lookfd           0.0           0           0           0
$ IADPE90      NCFREQ      IADPCL      ADPCTL      CBIRTH      CDEATH      LCLVL
       -1
$-----1-----2-----3-----4-----5-----6-----7-----8
*DEFINE_CURVE_BOX_ADAPTERIVITY
$ ID          PID      LEVEL      DIST1
   99           1           1        -25.0
$(3E20.0):
   -59.573399      -6.698870      -40.224651
   -90.728516      24.456253      -40.224651
...
   -23.213169      18.088070      -10.954337
   14.353654      16.130911      -10.954337
   -31.070744      -5.785467      -40.487387
   -59.573399      -6.698870      -40.224651
```

*DEFINE

*DEFINE_CURVE_COMPENSATION_CONSTRAINT

*DEFINE_CURVE_COMPENSATION_CONSTRAINT_OPTION

Purpose: Allow for the definition of a localized die face region for springback compensation of stamping tools. This keyword is only available for double precision.

Options available include:

BEGIN

END

NOTE: *DEFINE_CURVE_COMPENSATION_CONSTRAINT_BEGIN and *DEFINE_CURVE_COMPENSATION_CONSTRAINT_END are not valid in the context of a general keyword input deck. Instead, they may only be used inside of an *INCLUDE_COMPENSATION_CURVE include file.

The required option, which must be either BEGIN or END, distinguishes between two different closed curves. When taken together, these curves identify a portion of the die for applying springback compensation, and a transition region for which compensation smoothly tapers off.

Card 1	1	2	3	4	5	6	7	8
Variable	CRVID	INOUT	TYPE					
Type	I	I	I					
Default	none	0	none					

Point Cards. Include as many cards as needed. This input ends at the next keyword ("*") card. Only the projection of this curve onto the *xy*-plane is used.

Card 2	1	2	3	4	5	6	7	8
Variable	X		Y		Z			
Type	F		F		F			
Default	0.0		0.0		0.0			

VARIABLE	DESCRIPTION
CRVID	Curve ID; must be unique. The curve must be closed: its first and last point <i>must</i> coincide.
INOUT	Flag to indicate local area to be compensated: EQ.0: For this option, the compensated region of the die consists of all points for which the projection onto the <i>xy</i> -plane is exterior to the projection of the BEGIN curve. The projection of the END curve is assumed exterior to the BEGIN curve. The transition region, then, consists of all die points for which the projection is between the BEGIN and END curves. All other points on the die are uncompensated. EQ.1: For this option, the compensated region of the die consists of all points for which the projection onto the <i>xy</i> -plane is interior to the projection of the BEGIN curve. The projection of the END curve is assumed exterior to the BEGIN curve. The transition region, then, consists of all die points for which the projection is between the BEGIN and END curves. All other points on the die are uncompensated. See Figure 17-16 .
TYPE	Type code - must be "0".
X	<i>x</i> -coordinate of a point on the curve.
Y	<i>y</i> -coordinate of a point on the curve.
Z	<i>z</i> -coordinate of a point on the curve.

Motivation:

Sometimes springback occurs in a localized region of the die face. Since other parts of the die face are better left undisturbed, a localized compensation makes the most sense to bring the part shape back to the design intent. A typical example will be the front portion along the grill and headlamp or the rear portion along the windshield of a trimmed hood inner panel. A decklid (or trunk lid) inner also exhibits similar needs. Once the localized areas are identified, iterative compensation scheme may be employed within this localized region to bring the springback panel back to design shape.

Modeling Details:

Referring to [Figure 17-16](#), the keywords *COMPENSATION_CONSTRAINT_BEGIN and

*DEFINE

*DEFINE_CURVE_COMPENSATION_CONSTRAINT

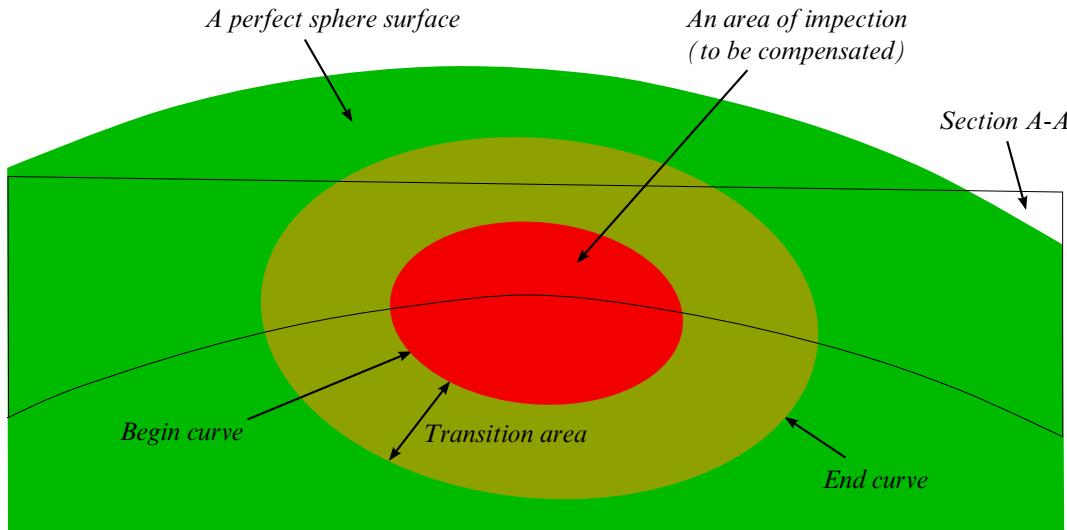


Figure 17-16. Local area compensation.

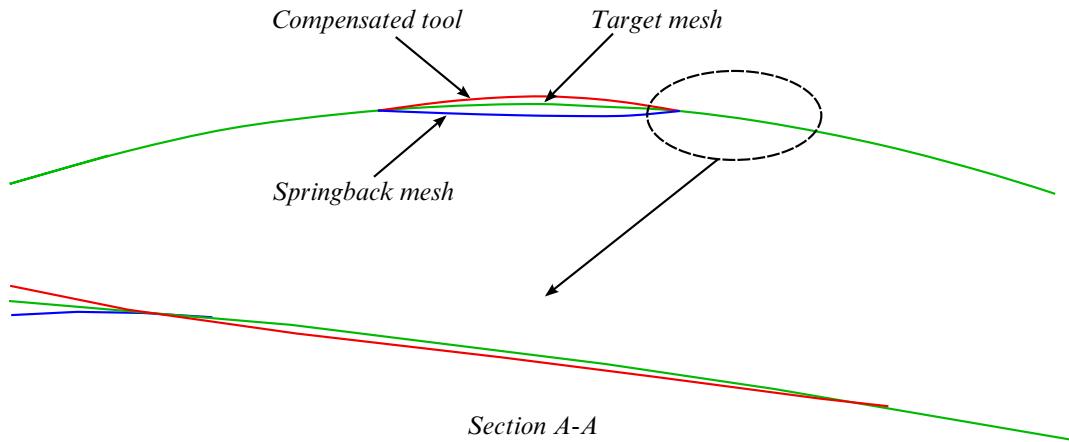
*COMPENSATION_CONSTRAINT_END must be used together in a file, which in turn will be included in keyword *INCLUDE_COMPENSATION_CURVE. The keyword "BEGIN" precedes the keyword "END;" each is defined by discrete points. In addition, each curve must form a closed loop. The area formed between the two curves is a transition area and will be affected in the compensated tooling. Multiple disconnected curves can be joined together and output in the .xyz format required here using *Curve → Merge → Multiple Method* in LS-PrePost4.0 and later.

The curve can be a 3D piecewise linear curve with coordinates in x , y and z . However, z -coordinates are ignored; meaning the tooling to be compensated must be positioned so the draw direction is in the global z -direction; otherwise an error will occur. In addition, it is assumed that both "blank before springback" and "blank after springback" will be smaller than rigid tools in dimension. Also, the rigid tool meshes should be discretized fine enough to provide enough degrees of freedom for the compensation.

Example – Single Region:

A complete input deck is provided below for a local compensation simulation. The keyword files state1.k and state2.k consist of model (nodes and elements) information for the blank before and after springback, respectively. If the blank is adaptively refined, the adaptive constraints must be included in the keyword files. The keyword file tools.k consists of the stamping tools (with PID 1, 2, 3 and 4) positioned in the home position. The keyword file curvesxy.xyz includes this keyword with both variations defining the two closed-loop curves used to define a localized area.

```
*KEYWORD
*TITLE
LS-Dyna971 Compensation Job
$-----1-----2-----3-----4-----5-----6-----7-----8
*INTERFACE_COMPENSATION_NEW
$ METHOD SL SF ELREF PSIDm UNDCT ANGLE NOLINEAR
```

**Figure 17-17.** Local compensation details.

```

6      10.000     0.700      0      1      0      0      1
*INCLUDE_COMPENSATION_BLANK_BEFORE_SPRINGBACK
state1.k
*INCLUDE_COMPENSATION_BLANK_AFTER_SPRINGBACK
state2.k
*INCLUDE_COMPENSATION_DESIRED_BLANK_SHAPE
state1.k
*INCLUDE_COMPENSATION_COMPENSATED_SHAPE
state1.k
*INCLUDE_COMPENSATION_CURRENT_TOOLS
tools.k
*INCLUDE_COMPENSATION_CURVE
curvesxy.xyz
*SET_PART_LIST
    1
1,2,3,4
*END

```

A portion of the file **curvesxy.xyz** is shown below,

```

*KEYWORD
*DEFINE_CURVE_COMPENSATION_CONSTRAINT_BEGIN
$ CID IN/OUT TYPE
    1      1      0
    -1.86925e+02   1.83338e+03   -1.55520e+01
    -1.83545e+02   1.83003e+03   -1.55469e+01
    -1.80162e+02   1.82668e+03   -1.55428e+01
    -1.91811e+02   1.83884e+03   -1.56014e+01
    -1.90187e+02   1.83701e+03   -1.55852e+01
    -1.88560e+02   1.83519e+03   -1.55688e+01
    -1.86925e+02   1.83338e+03   -1.55520e+01
*DEFINE_CURVE_COMPENSATION_CONSTRAINT_END
$ CID IN/OUT TYPE
    2      1      0
    -4.07730e+02   1.61371e+03   -8.04858e+01
    -3.84480e+02   1.59890e+03   -7.99169e+01
    -3.61193e+02   1.58423e+03   -7.93471e+01
    -3.37832e+02   1.56984e+03   -7.87756e+01
    -4.49289e+02   1.67556e+03   -8.04582e+01
    -4.35672e+02   1.65473e+03   -8.05162e+01
    -4.21764e+02   1.63396e+03   -8.05530e+01
    -4.07730e+02   1.61371e+03   -8.04858e+01
*END

```

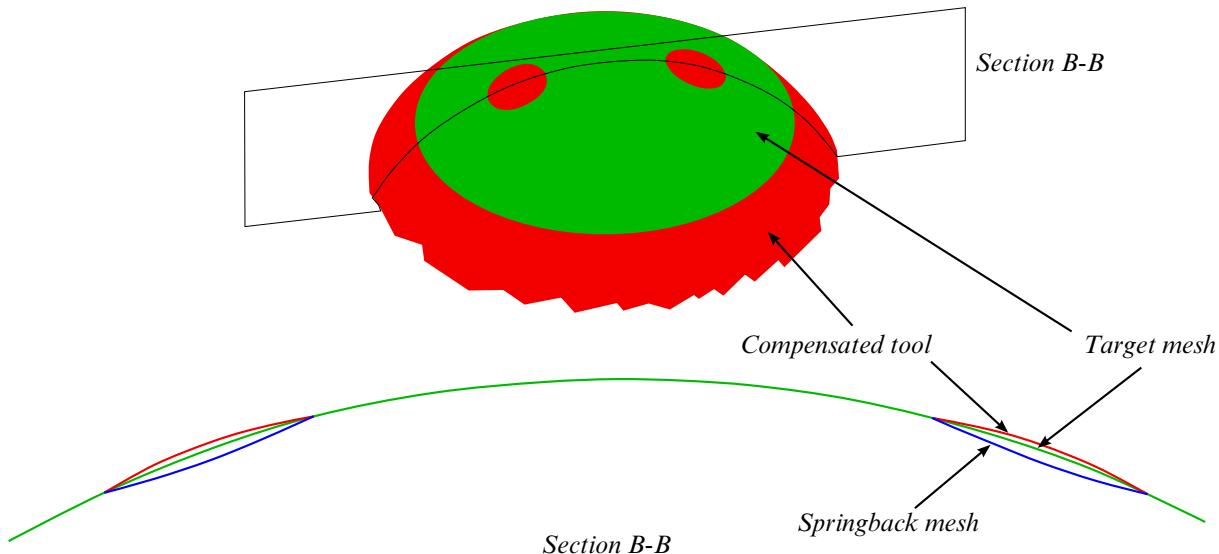


Figure 17-18. Multi-region local compensation.

It is noted the first point and last point are exactly the same, forming a closed loop. In [Figure 17-16](#), local area compensation is to be performed in the center portion of a rigid sphere. Based on springback and target meshes, the compensated tool mesh is obtained, and smooth transition areas are achieved (see [Figure 17-17](#)). Here the compensation scale factor of 0.7 is used.

Example – Multiple Regions:

Multi-region localized compensation is also possible by defining multiple pairs of the BEGIN and END versions of this keyword, each forming a localized region. For example, for localized compensation of two regions, the file `curvesxy.xyz` will read as follows,

```
*KEYWORD
*DEFINE_CURVE_COMPENSATION_CONSTRAINT_BEGIN
$ CID IN/OUT TYPE
    1      1      0
    3.67967e+02  1.63423e+03  -6.98532e+01
    3.60669e+02  1.62992e+03  -6.92921e+01
    3.53586e+02  1.62525e+03  -6.88777e+01
    :
    :
*DEFINE_CURVE_COMPENSATION_CONSTRAINT_END
$ CID IN/OUT TYPE
    2      1      0
    4.12534e+02  1.75537e+03  -5.83975e+01
    3.98853e+02  1.75264e+03  -5.58860e+01
    3.85292e+02  1.74921e+03  -5.35915e+01
    :
    :
*DEFINE_CURVE_COMPENSATION_CONSTRAINT_BEGIN
$ CID IN/OUT TYPE
    3      1      0
    -4.37478e+02  2.67393e+03  -1.70421e+02
    -4.45605e+02  2.67209e+03  -1.71724e+02
    -4.53649e+02  2.66985e+03  -1.72894e+02
    :
    :
*DEFINE_CURVE_COMPENSATION_CONSTRAINT_END
$ CID IN/OUT TYPE
```

DEFINE_CURVE_COMPENSATION_CONSTRAINT**DEFINE**

```
        4           1           0
-4.49426e+02   2.79057e+03   -2.18740e+02
-4.63394e+02   2.78749e+03   -2.20955e+02
-4.77223e+02   2.78370e+03   -2.22938e+02
        :
        :
        :
*END
```

[Figure 17-18](#) (top) shows an example of two localized areas of the sphere to be compensated. The compensation results are shown in [Figure 17-18](#) (bottom). Again, a compensation scale factor of 0.7 was used and smooth transition areas are achieved.

*DEFINE

*DEFINE_CURVE_DRAWBEAD

*DEFINE_CURVE_DRAWBEAD

Purpose: This keyword simplifies the definition of a draw bead, which previously required the use of many keywords.

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

Card 1	1	2	3	4	5	6	7	8
Variable	CID	TCTYPE	VID	PID	BLKID	PERCT	LCID	
Type	I	I	I	I	I	F	I	
Default	none	none	none	none	none	0.0	none	

Point Cards. For TCTYPE = 1 define points on the curve. Input is terminated at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	CX	CY	CZ					
Type	F	F	F					
Default	0.0	0.0	0.0					

IGES Card. For TCTYPE = 2 set an IGES file.

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

VARIABLE	DESCRIPTION
CID	Draw bead curve ID; must be unique.
TCTYPE	Flag to indicate input curve data format: EQ.1: XYZ data, EQ.2: IGES format data.
VID	Vector ID, as defined by *DEFINE_VECTOR. This vector is used to project the supplied curves to the rigid tool, defined by the PID below.
PID	Part ID of a rigid tool to which the curves are projected and attached.
BLKID	Part ID of the blank.
PERCT	Draw bead lock percentage or draw bead force. GT.0: Percentage of the full lock force for the bead defined. This is the ratio of desired restraining force over the full lock force. The value should be between 0.0 and 100.0. LT.0: Absolute value is the draw bead force.
LCID	Load curve ID defining material hardening curve of the sheet blank, BLKID.
CX, CY, CZ	Points on the curve.
FILENAME	IGES file name.

Remarks:

1. This feature implements the following input algorithm for drawbeads:
 - a) It reads a draw bead curve in either XYZ or IGES format
 - b) projects the curve to the rigid tool specified
 - c) creates extra node set and attaches it to the rigid tool.
 - d) With supplied material hardening curve (LCID), full lock force is calculated.

There is no need to define *CONTACT_DRAWBEAD and *CONSTRAINED_RIGID_BODIES since they are treated internally within the code.

2. The “curve” menu in LS-PrePost can be used to break or join multiple disconnected curves, and output in either ‘XYZ’ or IGES format.
3. The following partial keyword example defines a draw bead curve ID 98 (IGES file “bead1.iges”) to restrain blank part ID 63. Full lock force is calculated from the strain hardening curve ID 400. The draw bead is projected along vector ID 991, and is attached to a rigid tool of part ID 3.

```
$-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----  
-8  
*KEYWORD  
*DEFINE_VECTOR  
991,0.0,0.0,0.0,0.0,0.0,10.0  
*DEFINE_CURVE_DRAWBEAD  
$ CID TCTYPE VID PID BLKID PERCT LCID  
      98      2    991     3      63   52.442    400  
bead1.iges  
*MAT_037  
$ MID R0 E PR SIGY ETAN R  
HCLID  
      1 7.89E-09 2.00E+05      0.3      240.0          1.6  
400  
*DEFINE_CURVE  
400  
0.0,240.0  
0.02,250.0  
...  
1.0, 350.0  
*END
```

Revision information:

This feature is available starting in LS-DYNA R5 Revision 62464.

***DEFINE_CURVE_DUPLICATE**

Purpose: Define a curve by optionally scaling and offsetting the abscissa and ordinates of another curve defined by the *DEFINE_CURVE keyword.

Card 1	1	2	3	4	5	6	7	8
Variable	LCID	RLCID	SFA	SFO	OFFA	OFFO		
Type	I	I	F	F	F	F		
Default	none	none	1.	1.	0.	0.		

VARIABLE	DESCRIPTION
LCID	Load curve ID. Tables (see *DEFINE_TABLE) and load curve ID's must be unique.
RLCID	Reference load curve ID.
SFA	Scale factor for abscissa value of curve ID, RLCID. This value scales the SFA value defined for RLCID. EQ.0.0: default set to 1.0.
SFO	Scale factor for ordinate value (function) of curve ID, RLCID. This value scales the SFO value defined for RLCID. EQ.0.0: default set to 1.0.
OFFA	Offset for abscissa values. This value is added to the OFFA value defined for RLCID.
OFFO	Offset for ordinate values (function). This value is added to the OFFO value defined for RLCID.

*DEFINE

*DEFINE_CURVE_ENTITY

*DEFINE_CURVE_ENTITY

Purpose: Define a curve of straight line segments and circular arcs that defines an axisymmetric surface. This curve can only be used with the keyword *CONTACT_ENTITY for the load curve entity GEOTYP = 11.

Card 1	1	2	3	4	5	6	7	8
Variable	LCID	SFA	SFO	SFR	OFFA	OFFO	OFFR	
Type	I	F	F	F	F	F	F	
Default	none	1.	1.	1.	0.	0.	0.	

Point Cards. Put one point per card. Include as many cards as needed. This input terminates at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	A_i		O_i		R_i		IFLAG	
Type	F		F		F		I	
Default	0.0		0.0		optional		\downarrow	

VARIABLE	DESCRIPTION
LCID	Load curve ID. Tables (see *DEFINE_TABLE) and load curves may not share common IDs. LS-DYNA allows load curve IDs and table IDs to be used interchangeably. A unique number has to be defined.
SFA	Scale factor for axis value. This is useful for simple modifications. EQ.0.0: default set to 1.0.
SFO	Scale factor for radius values. This is useful for simple modifications. EQ.0.0: default set to 1.0.

VARIABLE	DESCRIPTION
SFR	Scale factor for circular radius. This is useful for simple modifications. EQ.0.0: default set to 1.0.
OFFA	Offset for axis values. See Remark 1 .
OFFO	Offset for radius values. See Remark 1 .
OFFR	Offset for circular radius. See Remark 1 .
A _i	Z-axis coordinates along the axis of rotation.
O _i	Radial coordinates from the axis of rotation
R _i	Radius of arc between points (A _i , O _i) and (A _{i+1} , O _{i+1}). If zero, a straight line segment is assumed.
IFLAG	Defined if R _i > 0.0. Set to 1.0 if the center of the arc is inside the axisymmetric surface and to -1.0 if the center is outside the axisymmetric surface.

Remarks:

1. **Scaling.** The load curve values are scaled after the offsets are applied, that is:

$$\text{Axis value} = \text{SFA} \times (\text{Defined value} + \text{OFFA})$$

$$\text{Radius value} = \text{SFO} \times (\text{Defined value} + \text{OFFO})$$

$$\text{Circular value} = \text{SFR} \times (\text{Defined value} + \text{OFFR})$$

*DEFINE

*DEFINE_CURVE_FEEDBACK

*DEFINE_CURVE_FEEDBACK

Purpose: Define information that is used as the solution evolves to scale the ordinate values of the specified load curve ID. This keyword is usually used in connection with sheet metal forming calculations.

Card 1	1	2	3	4	5	6	7	8
Variable	LCID	PID	BOXID	FLDID				
Type	I	I	I	I				
Default	none	none	0	none				

Card 2	1	2	3	4	5	6	7	8
Variable	FSL	TSL	SFF	SFT	BIAS			
Type	F	F	F	F	F			
Default	none	none	1.0	1.0	0.0			

VARIABLE	DESCRIPTION
LCID	ID number for load curve to be scaled.
PID	Active part ID for load curve control
BOXID	Box ID. Elements of specified part ID contained in box are checked. EQ.0: all elements of the active part are checked.
FLDID	Load curve ID which defines the flow limit diagram as shown in Figure 17-19 .
FSL	If the ratio, $r = \varepsilon_{\text{major,workpiece}} / \varepsilon_{\text{major,fl}}^{\text{fld}}$, exceeds FSL, then the scale factor for flow, SFF, is active.
TSL	Thickness strain limit. If the thickness strain limit is exceeded, then the scale factor for thickening, SFT, is active.

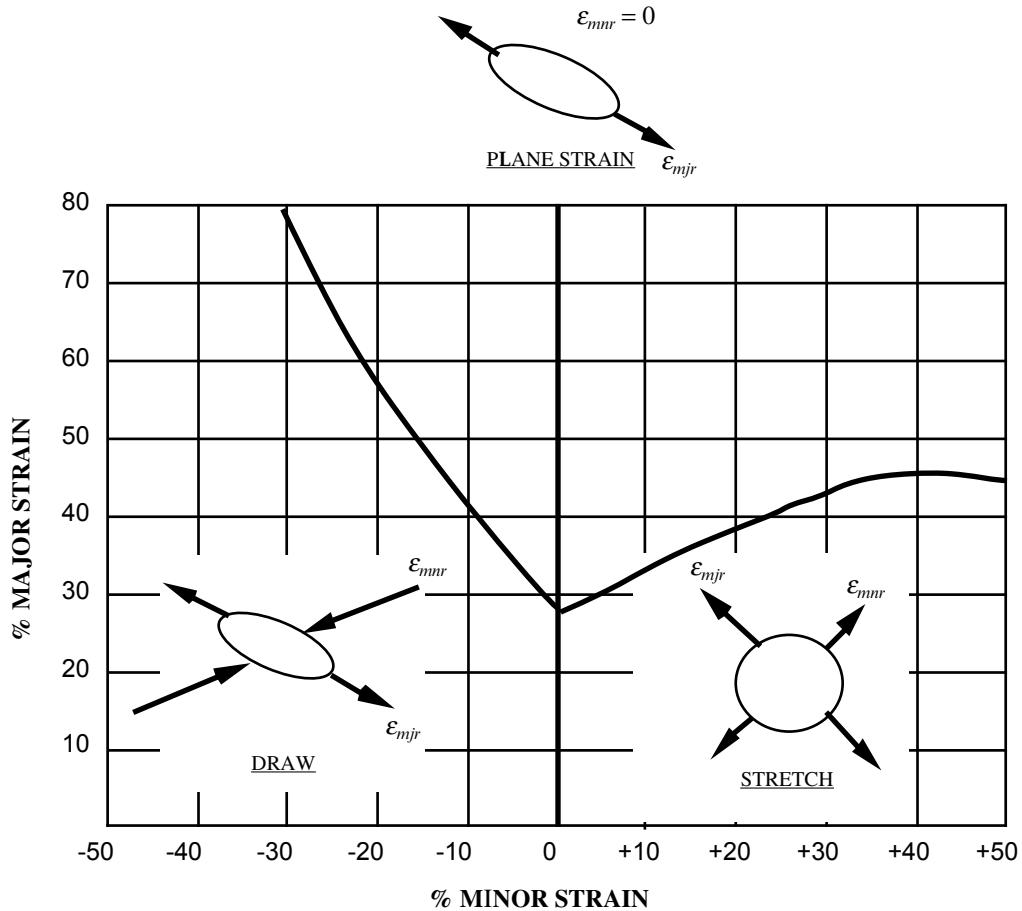


Figure 17-19. Flow limit diagram.

VARIABLE	DESCRIPTION
SFF	Scale factor for the flow limit diagram.
SFT	Scale factor for thickening.
BIAS	Bias for combined flow and thickening. Bias must be between -1 and 1.

Remarks:

This feature scales the ordinate values of a load curve according to a computed scale factor, S_f , that depends on both the major strain, r , and the through thickness, t . At each time step the load curve is scaled by S_f according to,

$$S_{\text{scaled load curve}}^{n+1} = S_f(r, t) \times S_{\text{load curve}}^n$$

where the superscript denotes the time step. The scale factor depends on r , which is a strain measure defined as,

$$r = \frac{\epsilon_{\text{major}_{\text{workpiece}}}}{\epsilon_{\text{major}_{\text{fld}}}}.$$

The scale factor, then, is given by,

$$S_f = \begin{cases} 1 & r < \text{FSL}, \quad t < \text{TSL} \\ \text{SFF} & r > \text{FSL}, \quad t < \text{TSL} \\ \text{SFT} & r < \text{FSL}, \quad t > \text{TSL} \\ \frac{1}{2}(1 - \text{BIAS}) \times \text{SFF} + \frac{1}{2}(1 + \text{BIAS}) \times \text{SFT} & r > \text{FSL}, \quad t > \text{TSL} \end{cases}$$

Usually SFF is slightly less than unity and SFT is slightly greater than unity so that $S_{\text{load curve}}$ changes insignificantly from time step to time step.

***DEFINE_CURVE_FLC**

Purpose: This keyword allows for defining the Forming Limit Diagram (FLD) using sheet metal thickness t and strain hardening value n . This keyword is applicable to shell elements only.

This feature is available in LS-DYNA Revision 61435 and later releases.

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

VARIABLE	DESCRIPTION
LCID	Load curve ID.
TH	Sheet metal thickness.
TN	Strain hardening value of the sheet metal, as in power law (Swift).

Remarks:

- Related Keywords.** This keyword is used in conjunction with keyword *MAT_TRANSVERSELY_ANISOTROPIC_ELASTIC_PLASTIC_NLP_FAILURE and is for shell elements only. For detailed formula of calculating the FLD based on sheet metal thickness and n -value, please refer to the following paper: *Ming F. Shi, Shawn Gelisse, "Issues on the AHSS Forming Limit Determination", ID-DRG 2006.*
- Material and Geometric Restrictions.** It is noted that this FLD calculation method is limited to sheet metal steels with thickness equal to or less than 2.5 mm, and it is not suitable for aluminum sheets. For aluminum sheets, *DEFINE_CURVE can be used to input the FLC for the field ICFLD in *MAT_TRANSVERSELY_ANISOTROPIC_ELASTIC_PLASTIC_NLP_FAILURE.

Example:

In a validation example shown in [Figure 17-20](#), a single shell element is stretched in three typical strain paths (linear): uniaxial, plane strain and equi-biaxial. Strain limits for each

*DEFINE

*DEFINE_CURVE_FLC

path are recovered when the history variable (Formability Index limit in *MAT_037) reaches 1.0, shown in [Figure 17-21](#). The top most point (strain limit) of each strain path coincides with the FLC curve calculated according to the paper, indicating the FLC defined by this keyword is working correctly. As shown in a partial keyword file below, the FLC is defined using a thickness value of 1.5 and n -value of 0.159. The LCID of 891 is used in keyword *MAT_TRANSVERSELY_ANISOTROPIC_ELASTIC_PLASTIC_NLP_FAILURE.

```
*MAT_TRANSVERSELY_ANISOTROPIC_ELASTIC_PLASTIC_NLP_FAILURE
$      MID      RO      E      PR      SIGY      ETAN      R      HLCID
$      1 7.830E-09 2.070E+05    0.28      0.0      0.0   -0.864      200
$     IDY      EA      COE      ICFLD
$                           891
*DEFINE_CURVE_FLC
$ LCID, TH, TN
891,1.5,0.159
$ DP600 NUMISHEET'05 Xmbr, Power law fitted
*DEFINE_CURVE
200
0.000,395.000
0.001,425.200
0.003,440.300
...
```

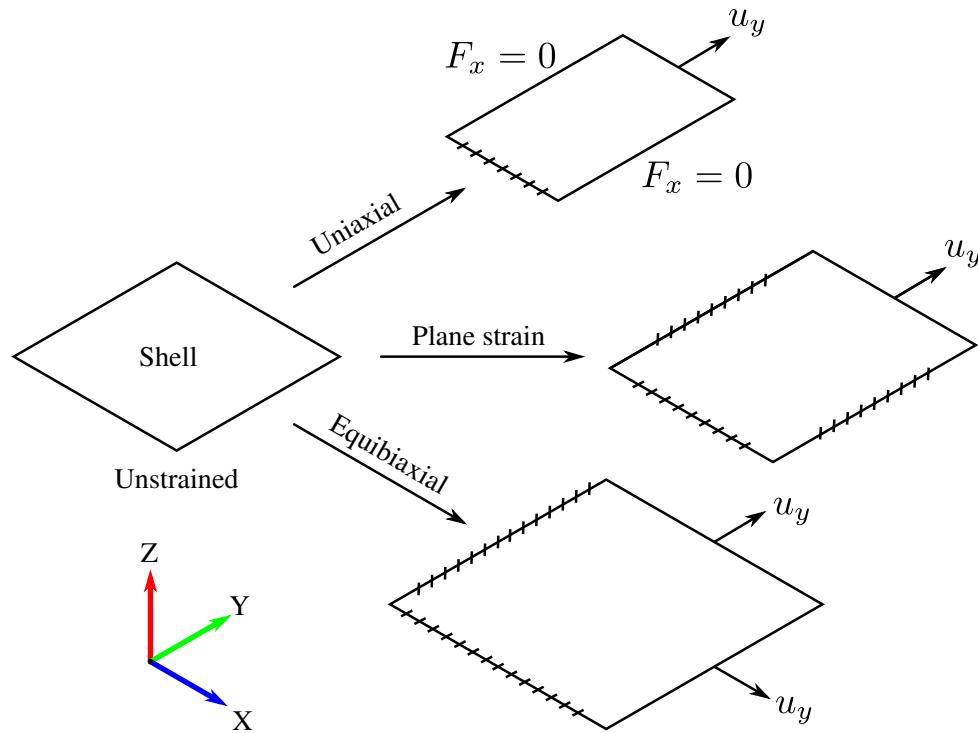


Figure 17-20. A single shell strained in three different strain paths

***DEFINE_CURVE_FLC**

***DEFINE**

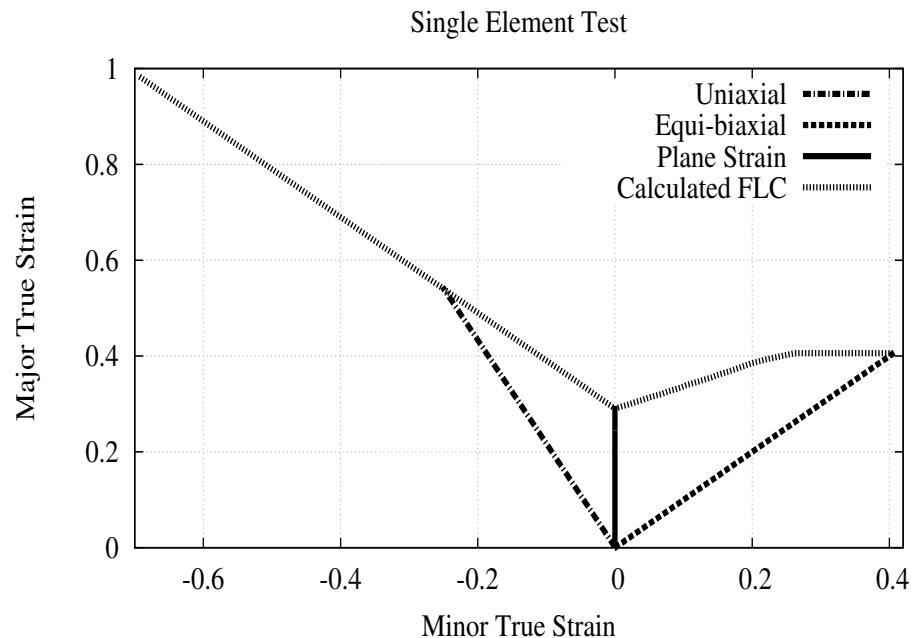


Figure 17-21. Validation of the FLC defined by this keyword

DEFINE**DEFINE_CURVE_FLD_FROM_TRIAXIAL_LIMIT*****DEFINE_CURVE_FLD_FROM_TRIAXIAL_LIMIT**

Purpose: Create a forming limit diagram (FLD) curve from a given stress triaxial limit curve, which can be referred to by material models utilizing a FLD curve, such as *MAT_037_NLP_FAILURE or *MAT_260B. The conversion assumes plane stress and Von-Mises yield criterion. The converted FLD curve can be found in the ".o" file (a scratch file from batch queue run). A related keyword is *DEFINE_CURVE_TRIAXIAL_LIMIT_FROM_FLD.

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

Point Cards. Put one pair of points per card (2E20.0). Input is terminated at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	A1		01					
Type	E20.0		E20.0					
Default	0.0		0.0					

VARIABLE	DESCRIPTION
LCID	The FLD curve ID to be created.
A1, A2, ...	Abscissas represent stress triaxialities, typically ranging from -1/3 to 2/3.
O1, O2, ...	Ordinates represent equivalent plastic strains to fracture.

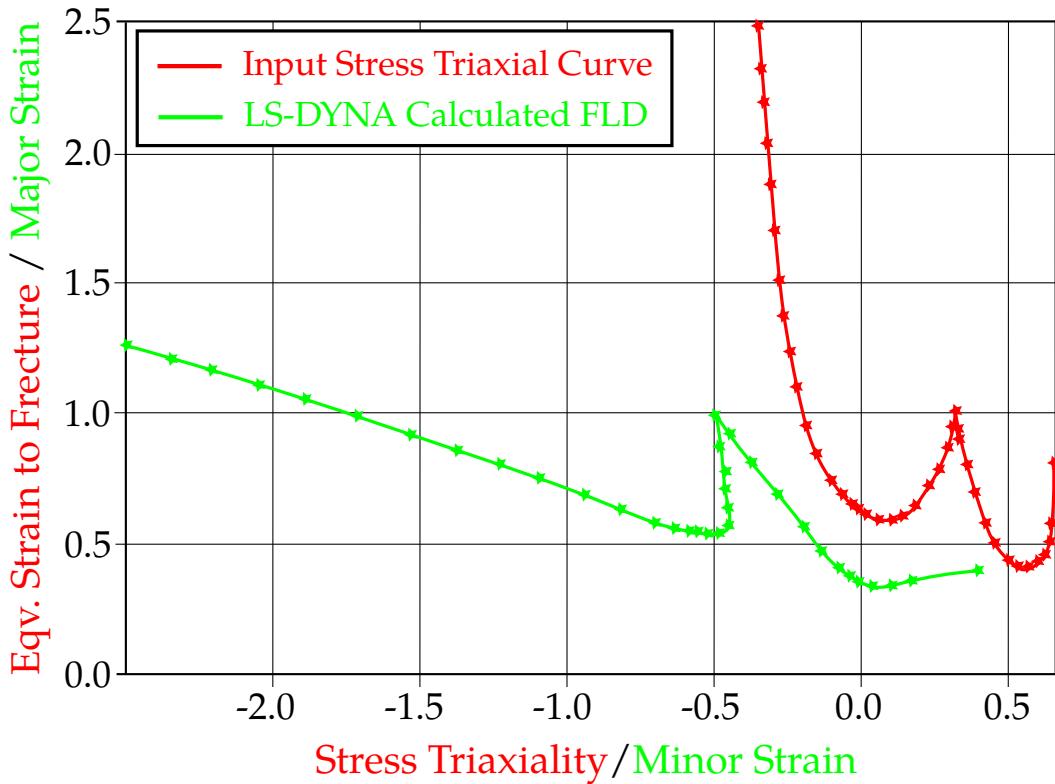


Figure 17-22. LS-DYNA calculated FLD from input stress triaxial curve [1].

Example:

The keyword input below includes the FLD limit curve obtained from Li et al. [1]. Figure 17-22 shows the calculated FLD curve from a uniaxial tension model on a single element using LS-DYNA, which matches the FLD curve from the paper.

```
*DEFINE_CURVE_FLD_FROM_TRIAXIAL_LIMIT
909
-.3284545, 2.485632
-.3193636, 2.327586
-.3102727, 2.198276
-.3011818, 2.04023
-.2875454, 1.882184
-.2739091, 1.70977
-.2602727, 1.522989
-.2466363, 1.37931
-.2239091, 1.235632
-.2011818, 1.106322
-.1648182, .9626437
-.133, .8477012
-8.754542E-02, .7471265
-4.663633E-02, .6896552
-1.481815E-02, .6465518
3.36367E-03, .632184
3.972731E-02, .6178162
8.518185E-02, .6034483
.1397273, .6034483
.1897273, .6465518
.2351819, .7183908
```

***DEFINE**

***DEFINE_CURVE_FLD_FROM_TRIAXIAL_LIMIT**

```
.2715455, .7758621  
.3033637, .862069  
.3306364, .9770116  
.3488182, .9195403  
.3715455, .8189656  
.3988182, .7040231  
.4351819, .5890805  
.4715455, .5028736  
.517, .4310345  
.5533637, .4166667  
.5760909, .4166667  
.617, .4310345  
.6397273, .4885058  
.6533636, .5603449  
.6670001, .8045977  
*end
```

Revision Information:

This feature is available starting from Revision 116631.

References:

- [1] Li, Yaning et al, "Prediction of shear-induced fracture in sheet metal forming," Journals of Material Processing Technology, Volume 210, issue 14, (2010).

DEFINE_CURVE_FUNCTION**DEFINE*****DEFINE_CURVE_FUNCTION**

Purpose: Define a curve [for example, load (ordinate value) versus time (abscissa value)] where the ordinate is given by a function expression. The function can reference other curve definitions, kinematical quantities, forces, interpolating polynomials, intrinsic functions, and combinations thereof. Please note that many functions require the definition of a local coordinate system (see [Remark 1](#) below). To output the curve to an ASCII database, see ***DATABASE_CURVOUT**. This command is not for defining curves for material models. Note that arguments appearing in square brackets “[]” are optional.

Card 1	1	2	3	4	5	6	7	8
Variable	LCID	SIDR						
Type	I	I						
Default	none	0						

Function Cards. Insert as many cards as needed. These cards are combined to form a single line of input. The next keyword (“*”) card terminates this input.

Card 2	1	2	3	4	5	6	7	8
Variable				FUNCTION				
Type				A80				
Remarks				1				

VARIABLE**DESCRIPTION**

LCID

Load curve ID. Tables (see ***DEFINE_TABLE**) and load curves may not share common ID's. LS-DYNA allows load curve ID's and table ID's to be used interchangeably. A unique number has to be defined.

SIDR

Stress initialization by dynamic relaxation:

EQ.0: Load curve used in transient analysis only or for other applications,

EQ.1: Load curve used in stress initialization but not transient analysis,

DEFINE**DEFINE_CURVE_FUNCTION**

VARIABLE	DESCRIPTION
	EQ.2: Load curve applies to both initialization and transient analysis.
FUNCTION	Arithmetic expression involving a combination of the following possibilities.

Constants and Variables:

FUNCTION	DESCRIPTION
TIME	Current simulation time
TIMESTEP	Current simulation time step
PI	Proportionality constant relating the circumference of a circle to its diameter
DTOR	Degrees to radians conversion factor ($\pi/180$)
RTOD	Radians to degrees conversion factor ($180/\pi$)
NCYCLE	Current integration cycle
IDRFLG	Carries a value of 1 (unity) during dynamic relaxation and 0 (zero) in the transient phase

Intrinsic Functions:

FUNCTION	DESCRIPTION
$\text{ABS}(a)$	Absolute value of a
$\text{AINT}(a)$	Nearest integer whose magnitude is not larger than a
$\text{ANINT}(a)$	Nearest whole number to a
$\text{MOD}(a_1, a_2)$	Remainder when integer a_1 is divided by integer a_2
$\text{SIGN}(a_1, a_2)$	Transfer sign of a_2 to magnitude of a_1
$\text{MAX}(a_1, a_2)$	Maximum of a_1 and a_2
$\text{MIN}(a_1, a_2)$	Minimum of a_1 and a_2
$\text{SQRT}(a)$	Square root of a

FUNCTION	DESCRIPTION
EXP(a)	e raised to the power of a
LOG(a)	Natural logarithm of a
LOG10(a)	Log base 10 of a
SIN(a)	Sine of a
COS(a)	Cosine of a
TAN(a)	Tangent of a
ASIN(a)	Arc sine of a
ACOS(a)	Arc cosine of a
ATAN(a)	Arc tangent of a
ATAN2(a_1, a_2)	Arc tangent of a_1/a_2
SINH(a)	Hyperbolic sine of a
COSH(a)	Hyperbolic cosine of a
TANH(a)	Hyperbolic tangent of a

Load Curves:

FUNCTION	DESCRIPTION
LC n	Ordinate value of curve n defined elsewhere (see *DEFINE_CURVE)
DELAY(LC n , t_{delay} , y_{def})	Delays curve n , defined by *DEFINE_CURVE_FUNCTION, *DEFINE_FUNCTION or DEFINE_CURVE, by a constant t_{delay} when simulation time $\geq t_{\text{delay}}$, and sets the delayed curve value to a constant y_{def} when time $< t_{\text{delay}}$, that is,

$$f_{\text{delay}}(t) = \begin{cases} f(t - t_{\text{delay}}) & t \geq t_{\text{delay}} \\ y_{\text{def}} & t < t_{\text{delay}} \end{cases}$$

For a nonlinear curve, a t_{delay} equal to more than 5,000 time steps might compromise the accuracy and must be used with caution.

When t_{delay} is a negative integer, delay time is input in terms

*DEFINE

*DEFINE_CURVE_FUNCTION

FUNCTION	DESCRIPTION
	of time step; $ t_{\text{delay}} $ is the number of delay time steps. In this case, $ t_{\text{delay}} $ is limited to a maximum of 100. For example, $t_{\text{delay}} = -2$ delays the curve by 2 time steps.

Coordinate Functions:

FUNCTION	DESCRIPTION
$\text{CX}(n)$	Value of x -coordinate for node n .
$\text{CY}(n)$	Value of y -coordinate for node n .
$\text{CZ}(n)$	Value of z -coordinate for node n .

Displacement Functions:

FUNCTION	DESCRIPTION
$\text{DM}(n_1[, n_2])$	Magnitude of translational displacement of node n_1 relative to node n_2 . Node n_2 is optional and if omitted, the displacement is computed relative to ground.
$\text{DMRB}(n)$	Magnitude of translational displacement of rigid body having a part ID of n
$\text{DX}(n_1[, n_2, n_3])$	x -translational displacement of node n_1 relative to node n_2 expressed in the local coordinate system of node n_3 . In other words, at any time t , the function returns the component of relative displacement that lies in the x -direction of the local coordinate system at time t . If node n_2 is omitted, it defaults to ground. If node n_3 is not specified, the displacement is reported in the global coordinate system.
$\text{DY}(n_1[, n_2, n_3])$	y -translational displacement of node n_1 relative to node n_2 expressed in the local coordinate system of node n_3 . In other words, at any time t , the function returns the component of relative displacement that lies in the y -direction of the local coordinate system at time t . If node n_2 is omitted it defaults to ground. If node n_3 is not specified, the displacement is reported in the global coordinate system.
$\text{DZ}(n_1[, n_2, n_3])$	z -translational displacement of node n_1 relative to node n_2 expressed in the local coordinate system of node n_3 . In other

FUNCTION	DESCRIPTION
	words, at any time t , the function returns the component of relative displacement that lies in the z -direction of the local coordinate system at time t . If node n_2 is omitted, it defaults to ground. If node n_3 is not specified, the displacement is reported in the global coordinate system.
DXRB(n)	x -translational displacement of rigid body having a part ID of n
DYRB(n)	y -translational displacement of rigid body having a part ID of n
DZRB(n)	z -translational displacement of rigid body having a part ID of n
AX($n_1[, n_2]$)	Rotational displacement of node n_1 about the local x -axis of node n_2 . If n_2 is not specified, then it defaults to ground. In computing this value it is assumed that the rotation about the other two axes (y -, z -axes) of node n_2 is zero. See Remark 1 .
AX2($n_1[, n_2]$)	Rotational displacement of node n_1 about the local x -axis of node n_2 . If n_2 is not specified, then it defaults to the local x -axis of node n_1 . This is recommended over AX, since it has fewer limitations. See Remark 1 .
AY($n_1[, n_2]$)	Rotational displacement of node n_1 about the local y -axis of node n_2 . If n_2 is not specified, then it defaults to ground. In computing this value it is assumed that the rotation about the other two axes (x -, z -axes) of node n_2 is zero. See Remark 1 .
AY2($n_1[, n_2]$)	Rotational displacement of node n_1 about the local y -axis of node n_2 . If n_2 is not specified, then it defaults to the local y -axis of node n_1 . This is recommended over AY, since it has fewer limitations. See Remark 1 .
AZ($n_1[, n_2]$)	Rotational displacement of node n_1 about the local z -axis of node n_2 . If n_2 is not specified, then it defaults to ground. In computing this value it is assumed that the rotation about the other two axes (x -, y -axes) of node n_2 is zero. See Remark 1 .
AZ2($n_1[, n_2]$)	Rotational displacement of node n_1 about the local z -axis of node n_2 . If n_2 is not specified, then it defaults to the local z -axis of node n_1 . This is recommended over AZ, since it has fewer limitations. See Remark 1 .

DEFINE**DEFINE_CURVE_FUNCTION**

FUNCTION	DESCRIPTION
PSI($n_1[, n_2]$)	First angle in the body2:313 Euler rotation sequence which orients node n_1 in the frame of node n_2 . If n_2 is omitted, it defaults to ground. See Remark 1 .
THETA($n_1[, n_2]$)	Second angle in the body2:313 Euler rotation sequence which orients node n_1 in the frame of node n_2 . If n_2 is omitted, it defaults to ground. See Remark 1 .
PHI($n_1[, n_2]$)	Third angle in the body2:313 Euler rotation sequence which orients node n_1 in the frame of node n_2 . If n_2 is omitted, it defaults to ground. See Remark 1 .
YAW($n_1[, n_2]$)	First angle in the body3:321 yaw-pitch-roll rotation sequence which orients node n_1 in the frame of node n_2 . If n_2 is omitted, it defaults to ground. See Remark 1 .
PITCH($n_1[, n_2]$)	Second angle in the body3:321 yaw-pitch-roll rotation sequence which orients node n_1 in the frame of node n_2 . If n_2 is omitted, it defaults to ground. See Remark 1 .
ROLL($n_1[, n_2]$)	Third angle in the body3:321 yaw-pitch-roll rotation sequence which orients node n_1 in the frame of node n_2 . If n_2 is omitted, it defaults to ground. See Remark 1 .

Velocity Functions:

FUNCTION	DESCRIPTION
VM($n_1[, n_2]$)	Magnitude of translational velocity of node n_1 relative to node n_2 . Node n_2 is optional and if omitted the velocity is computed relative to ground.
VR($n_1[, n_2]$)	Relative radial translational velocity of node n_1 relative to node. If node n_2 is omitted, it defaults to ground.
VX($n_1[, n_2, n_3]$)	x -component of the difference between the translational velocity vectors of node n_1 and node n_2 in the local coordinate system of node n_3 . If node n_2 is omitted, it defaults to ground. Node n_3 is optional and if not specified the global coordinate system is used.
VY($n_1[, n_2, n_3]$)	y -component of the difference between the translational velocity vectors of node n_1 and node n_2 in the local coordinate system of node n_3 . If node n_2 is omitted it defaults to ground. Node n_3 is

FUNCTION	DESCRIPTION
	optional and if not specified the global coordinate system is used.
VZ($n_1[, n_2, n_3]$)	z -component of the difference between the translational velocity vectors of node n_1 and node n_2 in the local coordinate system of node n_3 . If node n_2 is omitted, it defaults to ground. Node n_3 is optional and if not specified the global coordinate system is used.
WM($n_1[, n_2]$)	Magnitude of angular velocity of node n_1 relative to node n_2 . Node n_2 is optional and if omitted the angular velocity is computed relative to ground.
WX($n_1[, n_2, n_3]$)	x -component of the difference between the angular velocity vectors of node n_1 and node n_2 in the local coordinate system of node n_3 . If node n_2 is omitted, it defaults to ground. Node n_3 is optional and if not specified the global coordinate system is used. See Remark 1 .
WY($n_1[, n_2, n_3]$)	y -component of the difference between the angular velocity vectors of node n_1 and node n_2 in the local coordinate system of node n_3 . If node n_2 is omitted, it defaults to ground. Node n_3 is optional and if not specified the global coordinate system is used. See Remark 1 .
WZ($n_1[, n_2, n_3]$)	z -component of the difference between the angular velocity vectors of node n_1 and node n_2 in the local coordinate system of node n_3 . If node n_2 is omitted, it defaults to ground. Node n_3 is optional and if not specified the global coordinate system is used. See Remark 1 .

Acceleration Functions:

FUNCTION	DESCRIPTION
ACCM($n_1[, n_2]$)	Magnitude of translational acceleration of node n_1 relative to node n_2 . Node n_2 is optional and if omitted the acceleration is computed relative to ground. See Remark 1 .
ACCX($n_1[, n_2, n_3]$)	x -component of the difference between the translational acceleration vectors of node n_1 and node n_2 in the local coordinate system of node n_3 . If node n_2 is omitted, it defaults to ground. Node n_3 is optional and if not specified the global coordinate system is used. See Remark 1 .

DEFINE**DEFINE_CURVE_FUNCTION**

FUNCTION	DESCRIPTION
ACCY($n_1[, n_2, n_3]$)	y -component of the difference between the translational acceleration vectors of node n_1 and node n_2 in the local coordinate system of node n_3 . If node n_2 is omitted, it defaults to ground. Node n_3 is optional and if not specified the global coordinate system is used. See Remark 1 .
ACCZ($n_1[, n_2, n_3]$)	z -component of the difference between the translational acceleration vectors of node n_1 and node n_2 in the local coordinate system of node n_3 . If node n_2 is omitted, it defaults to ground. Node n_3 is optional and if not specified the global coordinate system is used. See Remark 1 .
WDTM($n_1[, n_2]$)	Magnitude of angular acceleration of node n_1 relative to node n_2 . Node n_2 is optional and if omitted the angular acceleration is computed relative to ground. See Remark 1 .
WDTX($n_1[, n_2, n_3]$)	x -component of the difference between the angular acceleration vectors of node n_1 and node n_2 in the local coordinate system of node n_3 . If node n_2 is omitted, it defaults to ground. Node n_3 is optional and if not specified the global coordinate system is used. See Remark 1 .
WDTY($n_1[, n_2, n_3]$)	y -component of the difference between the angular acceleration vectors of node n_1 and node n_2 in the local coordinate system of node n_3 . If node n_2 is omitted, it defaults to ground. Node n_3 is optional and if not specified the global coordinate system is used. See Remark 1 .
WDTZ($n_1[, n_2, n_3]$)	z -component of the difference between the angular acceleration vectors of node n_1 and node n_2 in the local coordinate system of node n_3 . If node n_2 is omitted, it defaults to ground. Node n_3 is optional and if not specified the global coordinate system is used. See Remark 1 .

Generic Force Functions:

FUNCTION	DESCRIPTION
FM($n_1[, n_2]$)	Magnitude of the SPC force acting on node n_1 minus the force acting on node n_2 . Node n_2 is optional; if omitted the force acting only on n_1 is returned. See Remark 1 .
FX($n_1[, n_2, n_3]$)	x -component of SPC force acting on node n_1 as computed in the optional local system of node n_3 . If n_2 is specified, the force acting

FUNCTION	DESCRIPTION
	on n_2 is subtracted from the force acting on n_1 . See Remark 1 .
FY($n_1[, n_2, n_3]$)	y -component of SPC force acting on node n_1 as computed in the optional local system of node n_3 . If n_2 is specified, the force acting on n_2 is subtracted from the force acting on n_1 . See Remark 1 .
FZ($n_1[, n_2, n_3]$)	z -component of SPC force acting on node n_1 as computed in the optional local system of node n_3 . If n_2 is specified, the force acting on n_2 is subtracted from the force acting on n_1 . See Remark 1 .
TM($n_1[, n_2]$)	Magnitude of SPC torque acting on node n_1 minus the torque acting node n_2 . Node n_2 is optional; if omitted the torque acting only on n_1 is returned. See Remark 1 .
TX($n_1[, n_2, n_3]$)	x -component of the SPC torque acting on node n_1 as computed in the optional local system of node n_3 . If n_2 is specified, the torque acting on n_2 is subtracted from the torque acting on n_1 . See Remark 1 .
TY($n_1[, n_2, n_3]$)	y -component of the SPC torque acting on node n_1 as computed in the optional local system of node n_3 . If n_2 is specified, the torque acting on n_2 is subtracted from the torque acting on n_1 . See Remark 1 .
TZ($n_1[, n_2, n_3]$)	z -component of the SPC torque acting on node n_1 as computed in the optional local system of node n_3 . If n_2 is specified, the torque acting on n_2 is subtracted from the torque acting on n_1 . See Remark 1 .

Sensor Functions:

FUNCTION	DESCRIPTION
SENSOR(n)	Returns a value of 1.0 if *SENSOR_CONTROL of control ID n has a status of “on”. If the sensor has a status of “off”, then the returned value is equal to the value of the TYPEID field on *SENSOR_CONTROL when the TYPE field is set to “function”, otherwise SENSOR(n) returns zero.
SENSORD($n, idflt$)	Returns the current value of *SENSOR_DEFINE sensor having ID n . Idflt is the optional filter ID defined using *DEFINE_FILTER.

Contact Force Functions:

FUNCTION	DESCRIPTION
RCFORC(<i>id, iba, comp, local</i>)	<p>Returns the component <i>comp</i> (see description below) of contact interface <i>id</i> (see *CONTACT_..._ID) as calculated in the local coordinate system <i>local</i> (see *DEFINE_COORDINATE_...). If <i>local</i> equals zero, then forces are reported in the global coordinate system. Forces are reported for the SURFA side when <i>iba</i> = 1 or SURFB side when <i>iba</i> = 2.</p> <p>The admissible values of <i>comp</i> and their corresponding force component are as follows:</p> <ul style="list-style-type: none">comp.EQ.1: <i>x</i> force componentcomp.EQ.2: <i>y</i> force componentcomp.EQ.3: <i>z</i> force componentcomp.EQ.4: Resultant force

Element Specific Functions:

FUNCTION	DESCRIPTION
BEAM(<i>id, jflag, comp, rm</i>)	<p>Returns the force component <i>comp</i> (see description below) of beam <i>id</i> as calculated in the local coordinate system <i>rm</i>. Forces are reported in the global coordinate system if <i>rm</i> is zero. If <i>rm</i> equals -1 the beam's <i>r, s</i>, and <i>t</i> force/moment is returned. If <i>jflag</i> is set to zero, then the force/torque acting on <i>n</i>₁ end of the beam is returned; otherwise if <i>jflag</i> is set to 1, the force/torque on the <i>n</i>₂ end of the beam is returned. See *ELEMENT_BEAM for the nodal connectivity rule defining <i>n</i>₁ and <i>n</i>₂.</p> <p>Admissible values of <i>comp</i> are 1-8 and correspond to the following components:</p> <ul style="list-style-type: none">comp.EQ.1: Force magnitudecomp.EQ.2: <i>x</i> force (axial <i>r</i>-force, <i>rm</i> = -1)comp.EQ.3: <i>y</i> force (<i>s</i>-shear force, <i>rm</i> = -1)comp.EQ.4: <i>z</i> force (<i>t</i>-shear force, <i>rm</i> = -1)comp.EQ.5: Torque magnitude

FUNCTION	DESCRIPTION
	<p>comp.EQ.6: x torque (torsion, rm = -1) comp.EQ.7: y torque (s-moment, rm = -1) comp.EQ.8: z torque (t-moment, rm = -1)</p>
ELHIST(<i>eid</i> , <i>etype</i> , <i>comp</i> , <i>ipt</i> , <i>local</i>)	<p>Returns the elemental quantity <i>comp</i> (see description below) of element <i>eid</i> as calculated in the local coordinate system <i>local</i>. Quantities are reported in the global coordinate system if <i>local</i> is zero. The parameter <i>ipt</i> specifies whether the quantity is for a particular integration point or maximum, minimum, or averaging is applied across the integration points.</p> <p>The following element classes, specified with <i>etype</i>, are supported:</p> <ul style="list-style-type: none">etype.EQ.0: Solidetype.EQ.2: Thin shell <p>Following are admissible values of <i>comp</i> and the corresponding elemental quantity:</p> <ul style="list-style-type: none">comp.EQ.1: x stresscomp.EQ.2: y stresscomp.EQ.3: z stresscomp.EQ.4: xy stresscomp.EQ.5: yz stresscomp.EQ.6: zx stresscomp.EQ.7: Effective plastic straincomp.EQ.8: Hydrostatic pressurecomp.EQ.9: Effective stresscomp.EQ.11: x straincomp.EQ.12: y straincomp.EQ.13: z straincomp.EQ.14: xy straincomp.EQ.15: yz straincomp.EQ.16: zx strain <p>Integration point options, specified with <i>ipt</i>, are:</p>

DEFINE**DEFINE_CURVE_FUNCTION**

FUNCTION	DESCRIPTION
	ipt.GE.1: Quantity is reported for integration point number <i>ipt</i>
	ipt.EQ.-1: Maximum of all integration points (default)
	ipt.EQ.-2: Average of all integration points
	ipt.EQ.-3: Minimum of all integration points
	ipt.EQ.-4: Lower surface integration point
	ipt.EQ.-5: Upper surface integration point
	ipt.EQ.-6: Middle surface integration point
	The local coordinate option <i>local</i> currently defaults to the global coordinate system for solid elements and other coordinate system options are unavailable. In the case of thin shell elements the quantity is reported only in the element local coordinate system.
	local.EQ.1: Global coordinate system (solid elements)
	local.EQ.2: Element coordinate system (thin shell elements)
JOINT(<i>id</i> , <i>jflag</i> , <i>comp</i> , <i>rm</i>)	Returns the force component <i>comp</i> (see description below) due to rigid body joint <i>id</i> as calculated in the local coordinate system <i>rm</i> . If <i>jflag</i> is set to zero, then the force/torque acting on n_1 end of the joint is returned. The force/torque on the n_2 end of the joint is returned if <i>jflag</i> is set to 1. See *CONSTRAINED_JOINT for the rule defining n_1 and n_2 . Admissible values of <i>comp</i> are 1-8 and correspond to the following components: <ul style="list-style-type: none">comp.EQ.1: Force magnitudecomp.EQ.2: <i>x</i> forcecomp.EQ.3: <i>y</i> forcecomp.EQ.4: <i>z</i> forcecomp.EQ.5: Torque magnitudecomp.EQ.6: <i>x</i> torque

DEFINE_CURVE_FUNCTION**DEFINE**

FUNCTION	DESCRIPTION
	comp.EQ.7: y torque
	comp.EQ.8: z torque

Nodal Specific Functions:

FUNCTION	DESCRIPTION
TEMP(n)	Returns the temperature of node n
DIST(n)	Returns distance traveled by node n
EPOT(n)	Returns the electric potential of node n of a piezoelectric material
ECHG(n)	Returns the reactive electric charge of node n of a piezoelectric material, conjugate to prescribed nodal electric potential.
ECHGBC(n)	Returns the summation of the reactive electric charges of all nodes associated with a prescribed electric potential boundary condition (see *BOUNDARY_PZEPOT) with an ID = n .

General Functions

FUNCTION	DESCRIPTION
CHEBY($x, x_0, a_0, \dots, a_{30}$)	Evaluates a Chebyshev polynomial at the user specified value x . The parameters $x_0, a_0, a_1, \dots, a_{30}$ are used to define the constants for the polynomial defined by: $C(x) = \sum a_j T_j(x - x_0)$ where the functions T_j is defined recursively as $T_j(x - x_0) = 2(x - x_0)T_{j-1}(x - x_0) - T_{j-2}(x - x_0)$ where $T_0(x - x_0) = 1$ $T_1(x - x_0) = x - x_0$
FORCOS($x, x_0, \omega[, a_0, \dots, a_{30}]$)	Evaluates a Fourier cosine series at the user specified value x . The parameters $x_0, a_0, a_1, \dots, a_{30}$ are used to define the constants for the series defined by: $F(x) = \sum a_j T_j(x - x_0)$ where

DEFINE**DEFINE_CURVE_FUNCTION**

FUNCTION	DESCRIPTION
	$T_j(x - x_0) = \cos[j\omega(x - x_0)]$
FORSIN($x, x_0, \omega[, a_0, \dots, a_{30}]$)	Evaluates a Fourier sine series at the user specified value x . The parameters $x_0, a_0, a_1, \dots, a_{30}$ are used to define the constants for the series defined by:
	$F(x) = \sum a_j T_j(x - x_0)$
	where
	$T_j(x - x_0) = \sin[j\omega(x - x_0)]$
IF(a_1, a_2, a_3, a_4)	Arithmetic if conditional where a_i can be a constant or any legal expression described in *DEFINE_CURVE_FUNCTION. For example, $a_1='CX(100)'$ sets the first argument to be the x -coordinate of node 100.
	$\text{IF} = \begin{cases} \text{the value of } a_2 & \text{if the value of } a_1 < 0 \\ \text{the value of } a_3 & \text{if the value of } a_1 = 0 \\ \text{the value of } a_4 & \text{if the value of } a_1 > 0 \end{cases}$
MORLET($freq, mag$)	Evaluates Morlet wavelet function with central frequency of $freq$ and peak value of mag :
	$M(t) = mag \times e^{-0.5 \times g^2(t)} \times \cos(5 \times f(t) - 20)$
	where
	$f(t) = 1.25 \times freq \times t$
	and
	$g(t) = f(t) - 4.$
PIDCTL($meas, ref, kp, ki, kd, tf, ei0, sint, umin, umax$)	Evaluates the control signal of a PID controller
	$u(t) = kp \times e(t) + ki \times \int_0^t e(\tau) d\tau + kd \times \frac{de(t)}{dt}$
	where $e(t)$ is the control error defined as the difference between the reference value ref and the measured value, $meas$
	$e(t) = ref - meas$
	The control parameters are proportional gain kp , integral gain ki , derivative gain kd and low-pass filter tf for the derivative calculation
	$\frac{de(t_n)}{dt} = \frac{de(t_{n-1})}{dt} \frac{tf}{\Delta t + tf} + \frac{\Delta t}{\Delta t + tf} \times \frac{e(t_n) - e(t_{n-1})}{\Delta t}$

FUNCTION	DESCRIPTION
	<i>ei0</i> is the initial integral value at time = 0.
	<i>sint</i> is the sampling interval. <i>umin</i> and <i>umax</i> , the lower and upper limit of a control signal, can be used to represent the saturation limits of an actuator. When the signal is not within the limits, it is clipped to the saturation limit, that is, integration is skipped to avoid integrator wind-up.
	Input parameter can be a constant or any legal expression described in *DEFINE_CURVE_FUNCTION. For example, <i>meas</i> = 'CX(100)' measures the <i>x</i> -coordinate of node 100, <i>ref</i> = 'LC200' uses curve 200 as the reference value.
POLY(<i>x, x₀, a₀, ..., a₃₀</i>)	Evaluates a standard polynomial at the user specified value <i>x</i> . The parameters <i>x₀, a₀, a₁, ..., a₃₀</i> are used to define the constants for the polynomial defined by:
	$P(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)^2 + \dots + a_n(x - x_0)^n$
RICKER(<i>freq, mag</i>)	Evaluates Ricker wavelet function with central frequency of <i>freq</i> and peak value of <i>mag</i> :
	$R(t) = mag \times (1 - 2 \times g(t)) / e^{g(t)} ,$ where
	$f(t) = PI \times freq \times (t - 1/freq)$ and
	$g(t) = f^2(t)$
SHF(<i>x, x₀, a, ω[, φ, b]</i>)	Evaluates a Fourier sine series at the user specified value <i>x</i> . The parameters <i>x₀, a₀, a₁, ..., a₃₀</i> are used to define the constants for the series defined by:
	$SHF = a \sin[\omega(x - x_0) - \phi] + b$
STEP(<i>x, x₀, h₀, x₁, h₁</i>)	Approximates the Heaviside function with a cubic polynomial using the equation:
	$STEP = \begin{cases} h_0 & \text{if } x \leq x_0 \\ h_0 + (h_1 - h_0) \left[\frac{(x - x_0)}{(x_1 - x_0)} \right]^2 \left\{ 3 - 2 \left[\frac{(x - x_0)}{(x_1 - x_0)} \right] \right\} & \text{if } x_0 < x < x_1 \\ h_1 & \text{if } x \geq x_1 \end{cases}$

Electromagnetic solver (EM) Functions

FUNCTION	DESCRIPTION
EM_ELHIST(<i>iele</i> , <i>ifield</i> , <i>idir</i>)	Returns the elemental quantity of element <i>iele</i> in the global reference frame.
EM_NDHIST(<i>inode</i> , <i>ifield</i> , <i>idir</i>)	Returns the nodal quantity of node <i>inode</i> in the global reference frame.
EM_PAHIIST(<i>ipart</i> , <i>ifield</i> , <i>idir</i>)	Returns the value integrated over the whole part given by <i>ipart</i> . <i>ifield</i> can be 7, 8 and 11 only. Admissible values of <i>ifield</i> are 1-10 and correspond to the following variables: <ul style="list-style-type: none">EQ.1: Scalar potentialEQ.2: Vector potentialEQ.3: Electric fieldEQ.4: B fieldEQ.5: H fieldEQ.6: Current densityEQ.7: Lorentz forceEQ.8: Joule heatingEQ.9: ConductivityEQ.10: Relative permeabilityEQ.11: Magnetic energy (in the conductor only) Admissible values of <i>idir</i> are 1-4 and correspond to the following components: <ul style="list-style-type: none">EQ.1: <i>x</i>-componentEQ.2: <i>y</i>-componentEQ.3: <i>z</i>-componentEQ.4: Norm

ICFD solver (ICFD) Functions

FUNCTION	DESCRIPTION
ICFD_NDHIST(<i>inode</i>, <i>ifield</i>, <i>idir</i>)	<p>Returns the nodal quantity of node <i>inode</i> in the global reference frame.</p> <p>Admissible values of <i>ifield</i> are 1-8 and correspond to the following variables:</p> <ul style="list-style-type: none"> EQ.1: Node position EQ.2: Velocity field EQ.3: Time-averaged velocity EQ.4: Density EQ.5: Dynamic viscosity EQ.6: Pressure EQ.7: Turbulent dynamic viscosity EQ.8: Level set <p>Admissible values of <i>idir</i> are 1-4 and correspond to the following components:</p> <ul style="list-style-type: none"> EQ.1: <i>x</i>-component EQ.2: <i>y</i>-component EQ.3: <i>z</i>-component EQ.4: Norm
ICFD_PAHIST(<i>ipart</i>, <i>ifield</i>, <i>idir</i>)	<p>Returns the value integrated over the whole ICFD surface part given by <i>ipart</i>.</p> <p>Admissible values of <i>ifield</i> are 1-11 and correspond to the following variables:</p> <ul style="list-style-type: none"> EQ.1: Pressure force EQ.2: Shear force EQ.3: Total force EQ.4: Moment due to pressure EQ.5: Moment due to shear EQ.6: Total moment EQ.7: Flow rate EQ.8: Time-averaged flow rate

FUNCTION	DESCRIPTION
	EQ.9: Pressure
	EQ.10: Time averaged Pressure
	EQ.11: Area
	Admissible values of <i>idir</i> are 1-4 and correspond to the following components:
	EQ.1: <i>x</i> -component
	EQ.2: <i>y</i> -component
	EQ.3: <i>z</i> -component
	EQ.4: Norm

Remarks:

1. **Local Coordinate Systems Required for Rotational Motion.** A local coordinate system *must* be attached to nodes if they are referenced by functions involving rotational motion, for example, angular displacement or angular velocity. The local coordinate system is attached to the node using *DEFINE_COORDINATE_NODES and FLAG = 1 is a requirement. Furthermore, the three nodes which comprise the coordinate system must lie on the same body. Similarly, a local coordinate system must also be attached to node n_3 if n_3 is referenced in functions: DX, DY, DZ, VX, VY, VZ, WX, WY, WZ, ACCX, ACCY, ACCZ, WDTX, WDTY, WDTZ, FX, FY, FZ, TX, TY, or TZ.
2. **Default is Radians.** Unless otherwise noted units of radians are always used for the arguments and output of functions involving angular measures.
3. **Reserved Variable Names.** See Appendix U for a list of variable names reserved internally by LS-DYNA.

The following examples serve only as an illustration of syntax.

Example 1:

Define a curve 10 whose ordinate is,

$$f(x) = \frac{1}{2}(\text{ordinate of load curve 9}) \times (\text{magnitude of translation velocity at node 22})^3.$$

```
*DEFINE_CURVE_FUNCTION
10
0.5*lc9*vm(22)**3
```

Example 2:

Define a curve 101 whose ordinate is,

$$f(x) = -2(\text{z translational displacement of node 38}) \times \sin(20\pi t).$$

```
*DEFINE_CURVE_FUNCTION  
101  
-2.*dz(38)*sin(2.*pi*10.*time)
```

Example 3:

Define a curve 202 whose ordinate is,

$$f(x) = \begin{cases} \cos(4\pi t) & \text{if } t \leq 5. \\ 0. & \text{if } t > 5. \end{cases}$$

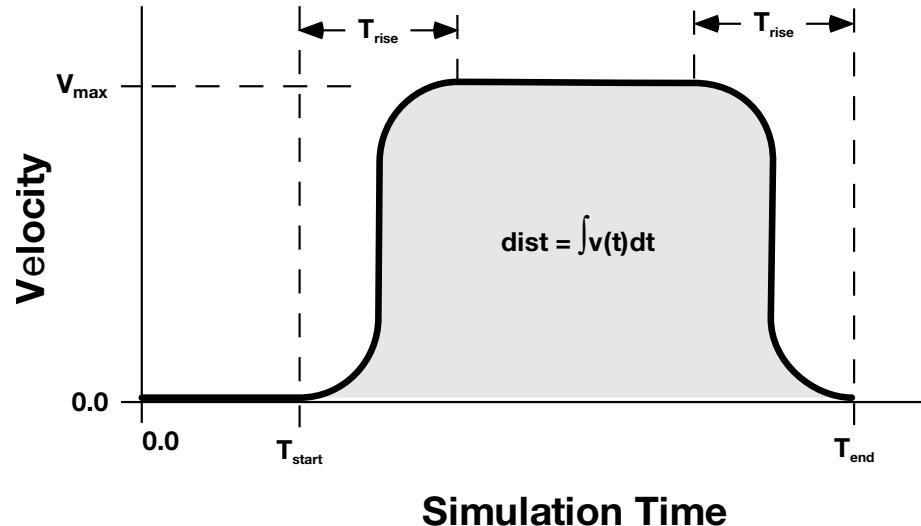
```
*DEFINE_CURVE_FUNCTION  
202  
If(TIME-5.,COS(4.*PI*TIME),COS(4.*PI*TIME),0.)
```

*DEFINE

*DEFINE_CURVE_SMOOTH

*DEFINE_CURVE_SMOOTH

Purpose: Define a smoothly varying curve using few parameters. This shape is useful for velocity control of tools in metal forming applications.



Simulation Time

Figure 17-23. Smooth curve created automatically using *DEFINE_CURVE_SMOOTH. This shape is commonly used to control velocity of tools in metal forming applications as shown in this figure but can also be used for other applications in place of any standard load curve.

Card	1	2	3	4	5	6	7	8
Variable	LCID	SIDR	DIST	TSTART	TEND	TRISE	VMAX	
Type	I	I	F	F	F	F	F	
Default	none	none	none	none	none	none	none	

VARIABLE

DESCRIPTION

LCID Load curve ID, must be unique

SIDR Stress initialization by dynamic relaxation:

EQ.0: Load curve is used in transient analysis only or for other applications,

EQ.1: Load curve is used in stress initialization but not transient analysis.

VARIABLE	DESCRIPTION
	EQ.2: Load curve applies to both initialization and transient analysis.
DIST	Total distance tool will travel (area under curve).
TSTART	Time curve starts to rise
TEND	Time curve returns to zero. If TEND is nonzero, VMAX will be computed automatically to satisfy required travel distance DIST. Input either TEND or VMAX.
TRISE	Rise time
VMAX	Maximum velocity (maximum value of curve). If VMAX is non-zero, TEND will be computed automatically to satisfy required travel distance DIST. Input either TEND or VMAX.

*DEFINE

*DEFINE_CURVE_STRESS

*DEFINE_CURVE_STRESS

Purpose: This keyword defines a material hardening curve based on a few commonly used material hardening laws. The hardening curve can also be a weighted combination of some of the laws. The load curve ID for this curve can be referenced in the load curve ID field used by a specific material model. This feature is applicable to all material models with a hardening curve that can be defined by a load curve using *DEFINE_CURVE.

Curve Definition Card(s). Only one card is needed unless ITYPE = 11. When ITYPE = 11, a second card with the form of CARD 1 is needed with the same LCID as the first card. See [Remark 2](#).

Card 1	1	2	3	4	5	6	7	8
Variable	LCID	ITYPE	P1	P2	P3	P4	P5	P6
Type	I	I	F	F	F	F	F	F
Default	none	none	none	1	none	none	none	none

VARIABLE

DESCRIPTION

LCID

Load curve ID for the stress-strain curve to be generated.

ITYPE

Type of hardening law:

EQ.1: Swift power law:

$$\sigma = K(e_0 + \epsilon_p)^n,$$

where σ is true effective stress, e_0 is the residual plastic strain at the initial yield point, K is a strength coefficient, ϵ_p is true effective plastic strain, and n is the work hardening coefficient.

EQ.2: Voce law with form:

$$\sigma = \sigma_0 + R_{\text{sat}}(1.0 - e^{-\zeta \epsilon_p}),$$

where σ_0 is the initial yield stress, R_{sat} is the stress differential between σ_0 and the saturated stress, and ζ is a strain coefficient.

EQ.3: Voce law with form:

$$\sigma = A - Be^{-C\epsilon_p},$$

where A , B , and C are material constants.

VARIABLE**DESCRIPTION**

EQ.4: Hockett-Sherby law:

$$\sigma = A - Be^{-C\epsilon_p^H},$$

where A , B , C , and H are material constants.

EQ.5: Stoughton-Yoon hardening law:

$$\sigma = A - Be^{-C\epsilon_p^m} + D\epsilon_p,$$

where A , B , C , m and D are material constants, such that $0 < m < 1.0$ and $D \geq 0.0$. According to Stoughton-Yoon, “with the exception of metals exhibiting Yield Point Elongation (YPE) effects, this function can represent the stress-strain response for both mild and AHSS steel and aluminum, from the initial yield point, throughout the small strain range, up to the highest strains realized in bulge tests”. Note that if $D = 0.0$, this law reduces to the Hockett-Sherby law (ITYPE = 4). Also note that if $m = 1.0$ and $D = 0.0$, this law reduces to the ITYPE = 3 Voce law.

EQ.11: A weighted combination of ITYPE = 1 and any of the other ITYPES. See [Remark 2](#).P1, P2, P3
P4, P5, P6

ITYPE	P1	P2	P3	P4	P5	P6
1	K	n	e_0			
2	σ_0	R_{sat}	ζ	w_2		
3	A	B	C	w_2		
4	A	B	C	H	w_2	
5	A	B	C	M	D	w_2
11	K	n	e_0	w_1		

Where w_1 and w_2 are the weighting factors used for ITYPE = 11; see [Remark 2](#).

Remarks:

1. **Default for P2.** For ITYPE = 1, the default value (also the minimum) for P2 is 0.000001.

2. **Combining Hardening Laws.** With ITYPE = 11, the Swift power law (ITYPE = 1) can be combined with another law through a weighted sum. To do this two cards are needed with the same LCID. The first card has ITYPE = 11 in place of ITYPE = 1. This card also requires a weight, w_1 , in the P4 field which is the weight applied to the Swift power law in the sum. The second card has the ITYPE of the law that is being summed to the Swift power law. This card also requires a weight, w_2 , in the Pi field specified for that ITYPE. The following example generates a hardening curve (LCID 90903) that is the sum of 0.5 (w_1) times the Swift power law and 0.8 (w_2) times the Hockett-Sherby law (ITYPE = 4). This curve then has the equation

$$\sigma = 0.5 \left(350.0 (0.01 + \epsilon_p)^{0.22} \right) + 0.8 \left(162.2 - 72.2 e^{-4.34\epsilon_p^{1.2}} \right).$$

```
*MAT_037
$      MID        RO        E        PR        SIGY        ETAN        R        HLCID
$      1 7.900E-09 2.070E+05     0.30
$*DEFINE_CURVE_STRESS
$-----1-----2-----3-----4-----5-----6-----7-----8
$ ITYPE = 1: power law.  P1 = K, P2 = n, P3 = e0, P4 = 0.5.
$ ITYPE = 4: Hockett-Sherby law.  P1 = A, P2 = B, P3 = C, P4 = H, P5 = 0.8.
$           VOCE: sigma = A-B*exp(-C*eps**H)
$      LCID        TYPE        P1        P2        P3        P4        P5
$      90903        11        350.0     0.22      0.01      0.5
$      90903        4         162.2     72.2      4.34      1.2      0.8
```

Revision Information:

This feature is available starting from Revision 113640. ITYPE = 5 (Stoughton-Yoon) is available starting from Revision 114803.

DEFINE_CURVE_TRIAXIAL_LIMIT_FROM_FLD**DEFINE*****DEFINE_CURVE_TRIAXIAL_LIMIT_FROM_FLD**

Purpose: Create a stress triaxial limit curve from a given forming limit diagram (FLD) curve, which can be referred to by material models utilizing a stress triaxial limit curve, such as *MAT_ADD_EROSION or *MAT_260B, or by keywords, such as *CONTROL_FORMING_ONESTEP. The conversion assumes plane stress and Von-Mises yield criterion. The converted stress triaxial curve can be found in the ".o" file (a scratch file from batch queue run). A related keyword is *DEFINE_CURVE_FLD_FROM_TRIAXIAL_LIMIT.

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

Point Cards. Put one pair of points per card (2E20.0). Input is terminated at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	A1		01					
Type	E20.0		E20.0					
Default	0.0		0.0					

VARIABLE	DESCRIPTION
LCID	The stress triaxial limit curve ID to be created.
A1, A2, ...	Abscissas represent minor true strains of a FLD curve.
O1, O2, ...	Ordinates represent major true strains of a FLD curve.

***DEFINE**

***DEFINE_CURVE_TRIAXIAL_LIMIT_FROM_FLD**

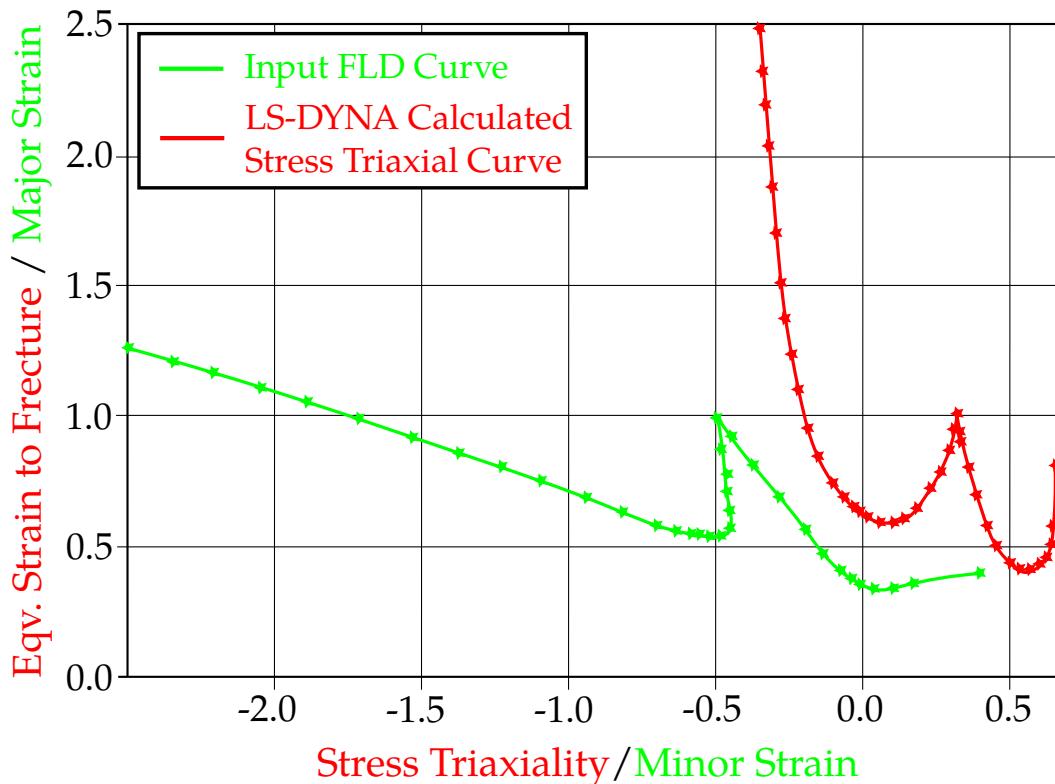


Figure 17-24. LS-DYNA calculated stress triaxial curve from FLD curve [1].

Example:

The keyword input below includes the FLD limit curve obtained from Li et al. [1]. Using a uniaxial tension model on a single element, [Figure 17-24](#) shows the calculated stress triaxial curve from LS-DYNA, which matches the triaxial curve from the paper.

```
*DEFINE_CURVE_TRIAXIAL_LIMIT_FROM_FLD
909
-2.485543, 1.260929
-2.326915, 1.211872
-2.196562, 1.173440
-2.037161, 1.115458
-1.876514, 1.064689
-1.701195, 0.9987057
-1.511570, 0.9169926
-1.364909, 0.8546170
-1.215432, 0.8004163
-1.080362, 0.7465187
-0.9267878, 0.6888055
-0.8039308, 0.6348159
-0.6904781, 0.5923799
-0.6199200, 0.5716710
-0.5669760, 0.5526081
-0.5458882, 0.5490728
-0.5156967, 0.5524927
-0.4797515, 0.5568793
-0.4477362, 0.5742416
-0.4447934, 0.6287722
-0.4554417, 0.7088588
```

```
-0.4556653, 0.7716601
-0.4686948, 0.8609383
-0.4924086, 0.9770012
-0.4380930, 0.9192049
-0.3606666, 0.8170972
-0.2779830, 0.6991526
-0.1942268, 0.5787448
-0.1300228, 0.4857036
-6.8558961E-02, 0.4028141
-2.8324760E-02, 0.3741716
-1.5549607E-03, 0.3616189
5.8079619E-02, 0.3408428
0.1153265, 0.3534369
0.1781329, 0.3710328
0.4022586, 0.4023391
*end
```

Revision Information:

This feature is available starting from Revision 116631.

References:

- [1] Li, Yining et al, "Prediction of shear-induced fracture in sheet metal forming," Journals of Material Processing Technology, Volume 210, issue 14, (2010).

***DEFINE_CURVE_TRIM_{OPTION}**

Available options include:

<BLANK>

3D

2D

Purpose: Define curves and controls for sheet blank trimming in sheet metal forming. This keyword can also be used to define mesh adaptivity along a curve within a band prior to the start of a simulation, using the variable TCTOL in this keyword and *CONTROL_ADAPTIVE_CURVE.

NOTE: The option NEW was replaced by 2D starting in Revision 115518.

The three keyword options for *DEFINE_CURVE_TRIM cause different trimming algorithms to be used. The <BLANK> (or no option) case is the oldest method. It is used for 2D trimming. However, it is not recommended because it is slow due to each node on the blank being checked to determine if it is inside the curve or not. Because this method is not recommended, the remarks section will only discuss the 2D and 3D keyword options. When the keyword option 3D is used, the trimming is processed based on the element normal, meaning that the trimming curve is projected to the nearest element using the element normal. The option 2D is used to trim in a fixed direction specified by a vector, that is, the curve is projected onto the model using the same vector. This option is also a 2D trimming. Currently this keyword applies to:

- 2D and 3D adaptive trimming of adaptive shell elements,
- 2D and 3D adaptive trimming of non-adaptive solids,
- 2D and 3D adaptive trimming of adaptive-meshed sandwiched parts (limit to a core of one layer of solid elements with outer layers of shell elements; see "IFSAND" under *CONTROL_ADAPTIVE),
- 2D and 3D adaptive trimming of adaptive-meshed sandwiched parts (a core of multiple layers of solid elements with outer layers of shell elements),
- 2D and 3D adaptive trimming (in conjunction with *CONTROL_FORMING_TRIMMING_SOLID_REFINEMENT) of adaptive-meshed sandwiched parts (a core of multiple layers of solid elements with outer layers of shell elements), and,
- 2D trimming of thick shell elements (TSHELL).

This keyword is not applicable to axisymmetric solids or 2D plane strain/stress elements. Related keywords include *ELEMENT_TRIM, *CONTROL_FORMING_TRIMMING, *CONTROL_FORMING_TRIMMING_SOLID_REFINEMENT, *CONTROL_ADAPTIVE_CURVE, *INCLUDE_TRIM, and *INCLUDE. Another closely related keyword is

***CONTROL_FORMING_TRIM_MERGE**, which automatically closes an open-ended trim curve within a user-specified tolerance.

NOTE: Trimming of shell and solid elements in LS-PrePost is supported starting with LS-PrePost 4.0, under *Application* → *MetalForming* → *Easy Setup*.

Card Summary:

Card 1a. Include this card if the keyword option is not used.

TCID	TCTYPE			TCTOL			
------	--------	--	--	-------	--	--	--

Card 1b. Include this card for the 2D keyword option.

TCID	TCTYPE	TFLG	TDIR	TCTOL	DEPTH	NSEED1	NSEED2
------	--------	------	------	-------	-------	--------	--------

Card 1c. Include this card for the 3D keyword option.

TCID	TCTYPE		TDIR	TCTOL	TOLN	NSEED1	NSEED2
------	--------	--	------	-------	------	--------	--------

Card 2a. Include as many of this card as needed if TCTYPE = 1. The next keyword ("*") card terminates this input.

CX	CY	CZ		
----	----	----	--	--

Card 2b. Include this card if TCTYPE = 2.

FILENAME

Data Card Definitions:

No Keyword Option Card. Include this card if not using a keyword option.

Card 1a	1	2	3	4	5	6	7	8
Variable	TCID	TCTYPE			TCTOL			
Type	I	I			F			
Default	none	1			0.25			

DEFINE**DEFINE_CURVE_TRIM**

VARIABLE	DESCRIPTION
TCID	ID number for trim curve
TCTYPE	Trim curve type: EQ.1: Curve data in XYZ format, obtained following procedures outlined in Figures under *INTERFACE_BLANKSIZE. In addition, only this format is allowed in *INTERFACE_COMPENSATION_3D. EQ.2: IGES trim curve
TCTOL	Tolerance limiting size of small elements created during trimming (see Figure 17-26). LT.0: "Simple" trimming, producing jagged edge mesh When used together with *CONTROL_ADAPTIVE_CURVE, it is a distance from the curve out (both sides). Within this distance the blank mesh will be refined, as stated in remarks below.

2D Keyword Option Card. Including this card if using the 2D keyword option.

Card 1b	1	2	3	4	5	6	7	8
Variable	TCID	TCTYPE	TFLG	TDIR	TCTOL	DEPTH	NSEED1	NSEED2
Type	I	I	I	I	F	F	I	I
Default	none	1	none	0	0.25	0.0	none	none

VARIABLE	DESCRIPTION
TCID	ID number for trim curve
TCTYPE	Trim curve type: EQ.1: Curve data in XYZ format, obtained following procedures outlined in Figures under *INTERFACE_BLANKSIZE. In addition, only this format is allowed in *INTERFACE_COMPENSATION_3D. EQ.2: IGES trim curve
TFLG	Element removal option:

VARIABLE	DESCRIPTION
	EQ.-1: Remove material outside curve. EQ.1: Remove material inside curve.
TDIR	ID of a vector (*DEFINE_VECTOR) giving the trim direction (see Figure 17-25). EQ.0: Default vector (0.0,0.0,1.0) is used. The curve is defined in global XY-plane and projected onto the mesh in the global Z-direction to define trim line.
TCTOL	Tolerance limiting size of small elements created during trimming (see Figure 17-26). LT.0: "Simple" trimming, producing jagged edge mesh When used together with *CONTROL_ADAPTIVE_CURVE, it is a distance from the curve out (both sides). Within this distance the blank mesh will be refined, as stated in remarks below.
DEPTH	The trimming depth is DEPTH – 1. If the distance between the element and the curve is larger than this value, then it will not be cut. This feature prevents trimming through to the opposite side of the part.
NSEED1/ NSEED2	A node ID on the blank in the area that remains after trimming: LT.0: NSEED i is a node which may not necessarily be on the blank. See Remark 6 .

3D Keyword Option Card. Include this card for the 3D keyword option.

Card 1c	1	2	3	4	5	6	7	8
Variable	TCID	TCTYPE	TFLG	TDIR	TCTOL	TOLN	NSEED1	NSEED2
Type	I	I	I	I	F	F	I	I
Default	none	1	0	1	0.25	2.0	none	none

VARIABLE	DESCRIPTION
TCID	ID number for trim curve

DEFINE**DEFINE_CURVE_TRIM**

VARIABLE	DESCRIPTION
TCTYPE	Trim curve type: EQ.1: Curve data in XYZ format, obtained following procedures outlined in Figures under *INTERFACE_BLANKSIZE. In addition, only this format is allowed in *INTERFACE_COMPENSATION_3D. EQ.2: IGES trim curve
TFLG	Side of surface for which trimming will start for sandwich parts: EQ.1: Top surface (default) EQ.2: Bottom surface
TDIR	Indicate whether the trim curve is near the top or bottom surface of the solids or laminates (>Revision 101964); see 3D (normal) trimming of solid elements . EQ.1: Trim curve is located near the top surface (default). EQ.-1: Trim curve is located near the bottom surface.
TCTOL	Tolerance limiting size of small elements created during trimming (see Figure 17-26). LT.0: "Simple" trimming, producing jagged edge mesh When used together with *CONTROL_ADAPTIVE_CURVE, it is a distance from the curve out (both sides). Within this distance the blank mesh will be refined, as stated in remarks below.
TOLN	Maximum gap between the trimming curve and the mesh. If the gap is bigger than this value, this section in the curve will not be used.
NSEED1/ NSEED2	A node ID on the blank in the area that remains after trimming: LT.0: NSEED <i>i</i> is a node which may not necessarily be on the blank. See Remark 6 .

DEFINE_CURVE_TRIM**DEFINE**

Point Cards. Additional cards for TCTYPE = 1. Put one point per card (2E20.0). Input is terminated at the next keyword ("*") card.

Card 2a	1	2	3	4	5	6	7	8
Variable	CX		CY		CZ			
Type	F		F		F			
Default	0.0		0.0		0.0			

VARIABLE **DESCRIPTION**

CX, CY, CZ *X, Y, Z-coordinates of trim curve.*

IGES File Card. Additional card for TCTYPE = 2.

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

VARIABLE **DESCRIPTION**

FILENAME Name of IGES database containing trim curve(s).

Trimming Capability Summary:

This keyword and its options deal with trimming of the following scenarios:

	2D (along one direction)	3D (element normal)	2D & 3D Double Trim	Adaptive mesh
Shell	Yes	Yes	Yes	Yes
Solids	Yes	Yes	Yes	N/A
Sandwiched Parts (Laminates)	Yes	Yes	Yes	Yes
TSHELL	Yes	N/A	N/A	N/A

Remarks:

1. **IGES Trim Curve.** Only IGES entities 110 and 106 are supported when using TCTYPE = 2. The eZ-Setup for trimming function ensures correct IGES files are written for a trimming simulation.
2. **Trim Curve Requirements.** Enclosed trimming curves (same start and end points) are required for all options. Also, for each enclosed trimming curve, only one curve segment is acceptable for the option 3D; while several curve segments are acceptable with the option 2D. Curves can be manipulated through the *Merge* and *break* features in LS-PrePost4.0, found under *Curve/Merge* (always select *piecewise* under *Merge*) and *break*.

For 3D trimming, trim curves need to be sufficiently close to the part. Trim curves can be processed using curve projection onto the mesh in LS-PrePost. This feature is accessible under *GeoTol/Project/Closest Proj/Project to Element/By Part*. Using a double precision LS-DYNA executable may also help in this situation.

For the option 2D, Revision 68643 and later releases enable trimming of a part where trim lines go beyond the part boundary. This is illustrated in [Figure 17-31](#).

3. **Order of Trimming.** This keyword in combination with *ELEMENT_TRIM trims the requested parts before a job starts (pre-trimming). This combination can also handle an adaptive mesh. If the keyword *ELEMENT_TRIM is not included, the parts are trimmed after the job is terminated (post-trimming).
4. **TCTOL.** The trimming tolerance TCTOL limits the size of the smallest element created during trimming. A value of 0.0 places no limit on element size. A value of 0.5 restricts new elements to be at least half of the size of the parent element. A value of 1.0 allows no new elements to be generated, only repositioning of existing nodes to lie on the trim curve. A negative tolerance value activates "simple" trimming, where entire elements are removed, leaving a jagged edge.
5. **Mesh Refinement along the Trim Curve for Shell Elements.** When large elements are along the trim curve, the blank mesh can be pre-adapted along the trim curve before trimming by adding the keyword *CONTROL_ADAPTIVE_CURVE for a better quality trim edge. Care should be taken when using this keyword since an excessive number of elements can be created when setting the fields. Note that this keyword can only be used for shell elements.

The variable TCTOL can be used to control the mesh refinement along a curve when used together with *CONTROL_ADAPTIVE_CURVE. In this scenario, it is the distance of the entire refinement width. The mesh will be refined in the beginning of the simulation. This method offers greater control on the number of elements to be generated during mesh refinement. A detailed description and

example are provided under the manual section under *CONTROL_ADAPTIVE_CURVE.

Sometimes it is helpful to conduct a check of the trimmed mesh along the edge in the same trimming input deck using the keyword *CONTROL_CHECK_SHELL. This is especially useful for the next continued process simulation. For detailed usage, see that manual page.

6. **Seed Nodes.** A seed node is used to define which side of the drawn panel will be kept after trimming. With the frequent application of adaptive re-meshing, the seed node for trimming is often unknown until the draw forming is complete. When NSEED i is negative, an extra node unrelated to the blank and tools can be used as the seed node, enabling simulating a trimming process independent of the previous process simulation results. The extra node can be defined using keyword *NODE. If the seed node is too far away from the blank, it will be projected onto the blank and the new position will be used as the seed node. Typically, this node can be selected from the stationary tool in its home position. Selecting a seed node is quite easy in *Trimming* process of LS-PrePost4.0 eZSetup for metal forming application.

A partial keyword input example for the trimming of a double-attached *NUMISHEET2002 fender outer* with the option 2D is listed below, where a 2D trimming is performed with IGES file *doubletrim.iges* in the global Z-axis, with two nodes of negative ID 43356 and 18764 assigned to the variables NSEED1 and NSEED2, respectively. The two seed nodes are defined off the stationary lower post and do not necessarily need to be a part of the post as shown in [Figure 17-27](#). The drawn panels in the wire frame are shown in [Figure 17-28](#), along with the thickness/thinning contour ([Figure 17-29](#)). In [Figure 17-30](#), the drawn panels are trimmed and separated.

```
*KEYWORD
*CONTROL_TERMINATION
0.000
*CONTROL_SHELL
.....
*CONTROL_OUTPUT
.....
*DATABSE_BINARY_D3PLOT
.....
*DATABSE_EXTENT_BINARY
.....
$-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
*SET_PART_LIST
.....
*PART
Blank
.....
*SECTION_SHELL
.....
*MAT_3 -PARAMETER_BARLAT
.....
$-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
*INCLUDE_TRIM
```

```
drawn.dynain
*ELEMENT_TRIM
  1
*DEFINE_CURVE_TRIM_2D
$#   TCID      TCTYPE      TFLG      TDIR      TCTOL      DEPTH      NSEED1      NSEED2
    1          2                  0        0.250           1       -43356      -18764
doubletrim.iges
$-----1-----2-----3-----4-----5-----6-----7-----8
*NODE
18764,-184.565,84.755,78.392
43356,-1038.41,119.154,78.375
*INTERFACE_SPRINGBACK_LSDYNA
.....
*END
```

Alternatively, if the fields $NSEED_i$ are not defined, the seeds can be defined using *DEFINE_TRIM_SEED_POINT_COORDINATES. A partial keyword input is provided below for trimming of the same double-attached fender outer.

```
*INCLUDE_TRIM
drawn.dynain
*ELEMENT_TRIM
  1
*DEFINE_CURVE_TRIM_2D
$#   TCID      TCTYPE      TFLG      TDIR      TCTOL      DEPTH      NSEED1      NSEED2
    1          2                  0        0.250           1
doubletrim.iges
*DEFINE_TRIM_SEED_POINT_COORDINATES
$   NSEED      X1      Y1      Z1      X2      Y2      Z2
    2     -184.565    84.755   78.392   -1038.41   119.154    78.375
```

7. **Nodal Temperature.** Starting in Revision 81105, trimming of blank with nodal temperature (related to hot and warm stamping) is available. At the end of the trimming, the file new_temp_ic.inc will be output along with the usual dynain file. Furthermore, the keyword *CONTROL_CHECK_SHELL can be used together when trimming with nodal temperature.
8. ***INCLUDE_TRIM.** We recommend using the keyword *INCLUDE_TRIM at all times, either for trimming or for mesh refinement purposes, except in the case where the to-be-trimmed sheet blank has no stress and strain information (no *INITIAL_STRESS_SHELL, and *INITIAL_STRAIN_SHELL cards present in the sheet blank keyword or dynain file). In that case the keyword *INCLUDE must be used. For solid element and thick shell element trimming, the keyword *INCLUDE_TRIM *must* be used. A check box to indicate that the blank is free of stress and strain information is provided in the Trimming process in the eZ-Setup for users to set up a trimming input deck under the circumstance.

2D and 3D trimming of solid elements and laminates:

In all trimming of solids and laminates, only a dynain file is written out (no d3plot files will be output).

2D trimming of solid elements

As of Revision 92088, 2D (option 2D) trimming in any direction (defined by a vector) of solid elements is available. An illustration of the 2D trim is shown in [Figure 17-32](#). A partial keyword example is provided below, where trim curves in trimcurves2d.iges are being used to perform a solid element trimming along a vector defined along the global Z-axis.

```
*KEYWORD
*INCLUDE_TRIM
incoming.dynain
$-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
*PARAMETER_EXPRESSION
...
*CONTROL_TERMINATION
$      ENDTIM
0.0
*CONTROL_OUTPUT
...
*DATABSE_XXX
...
*PART
Solid Blank
$#      pid      secid      mid
&blk1pid &blk1sec &blk1mid
*SECTION_Solid
&blk1sec,&elform
*MAT_PIECEWISE_LINEAR_PLASTICITY
...
$-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
$      Trim cards
$-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
*CONTROL_FORMING_TRIMMING
$      PSID
&blk1sid
*DEFINE_TRIM_SEED_POINT_COORDINATES
$ NSEED,X1,Y1,Z1,X2,Y2,Z2
1,&seedx,&seedy,&seedz
$-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
*DEFINE_CURVE_TRIM_2D
$#      tcid      tctype      tflg      tdir      tctol      depth      nseed1      nseed2
      2          2           0          1       0.10000   1.000000          0          0
$# filename
trimcurves2d.iges
*DEFINE_VECTOR
$#      vid      xt      yt      zt      xh      yh      zh      cid
      1      0.000     0.000    0.000     0.000     0.000   1.000000          0
*INTERFACE_SPRINGBACK_LSDYNA
$      PSID
&blk1sid,&nshv
*END
```

Currently, 2D trimming of solids in some cases may be approximate. The trimming will trim the top and bottom sides of the elements, not crossing over to the other sides. This can be seen, for instance, in trimming involving a radius.

3D (normal) trimming of solid elements

As of Revision 93467 3D trimming (option 3D) of solid elements is available. The trim curve should be created based on the solid element normal. The field TDIR indicates whether the curve is closer to the top (TDIR = 1) or bottom (TDIR = -1) surface of the solid; see [Figure 17-34](#). The projection of trim curves onto either the top or bottom surface of the blank is important to ensure a smooth and successful trimming.

To change the example in the 2D case to the 3D case, the option 2D is changed to 3D, and trimcurves3d.iges is used. The field TDIR is set to “1” since the trim curve is on the positive side of the element normal ([Figure 17-34](#)). Since 3D trimming is along the element normal directions, *DEFINE_VECTOR is no longer needed.

```
*DEFINE_CURVE_TRIM_3D
$#      tcid      tctype      tflg      tdir      tctol      toln      nseed1      nseed2
      2          2           0          1     0.10000   1.000000         0          0
$# filename
trimcurves3d.iges
```

2D and 3D trimming of non-adaptive-meshed sandwiched parts (laminates)

2D and 3D trimming of non-adaptive-meshed laminates are available starting in Revision 92289. The laminates can have multiple layers of solid elements, sandwiched by a top and a bottom layer of shell elements. Note that the nodes of the shell elements must share the nodes with the solid elements at the top and bottom layers. See [Figure 17-33](#) as an example. The input deck is similar to those used for trimming of solid elements, except the variable ITYP under *CONTROL_FORMING_TRIMMING should be set to “1” to activate the trimming of laminates in both 2D and 3D conditions:

```
*CONTROL_FORMING_TRIMMING
$-----1-----2-----3-----4-----5-----6-----7-----8
$      PSID          ITYP
&blkSID          1
```

Again, in the case of 3D trimming, the projection of trim curves onto either the top or bottom surface of the blank is important to ensure a smooth and successful trim.

2D and 3D trimming of adaptive-meshed sandwiched parts - one core layer of solids

2D and 3D trimming of adaptive-meshed laminates with one core layer of solid elements sandwiched by shell elements are available starting in Revision 108770. The shell elements must share the same nodes as the solid elements.

*Note elements are refined automatically along the trim curves until no dependent nodes (see *CONSTRAINED_ADAPTIVITY) would be cut by the trim curves. In addition, the keyword *CONTROL_ADAPTIVE_CURVE must not be used since it is only applied to shell elements and would cause error termination otherwise. This trimming unlike for shell elements*

requires no additional adaptivity-related keyword inputs for mesh refinement. An example of the trimming on the 2005 NUMISHEET Cross Member is shown in [Figure 17-35](#).

2D and 3D trimming of adaptive-meshed sandwiched parts - multiple layers of solids:

This capability is available starting in Dev 134513 when used with the keyword *CONTROL_FORMING_TRIMMING_SOLID_REFINEMENT. This keyword allows for the elements along the trim curves to be refined. Please refer to the corresponding manual pages.

2D trimming of thick shell elements (TSHELL):

2D trimming of thick shells supported starting from Revision 107957. Note that, by definition, thick shells have only one layer of solid elements, and is defined by keyword Note also *INCLUDE_TRIM (not *INCLUDE) must be used to include the dynain file to be trimmed. The input deck for 2D trimming of TSHELL is similar to what is used for trimming of solid elements.

2D and 3D trimming of double-attached solids and laminates:

These features are available starting in Revision 110140. Both seed point coordinates can be specified in *DEFINE_TRIM_SEED_POINT_COORDINATES to define a seed coordinate for each part, as shown below:

```
*DEFINE_TRIM_SEED_POINT_COORDINATES
$      NSEED          X1          Y1          Z1          X2          Y2          Z2
      2    -184.565    84.755    78.392   -1038.41   119.154    78.375
```

In [Figure 17-36](#), a 2D double-trimming example on a sandwiched part is shown using the 2005 NUMISHEET cross member. Two trim curves and two seed nodes are defined for each to be trimmed portion.

Note again, *INCLUDE_TRIM (not *INCLUDE) must be used to include the dynain file to be trimmed.

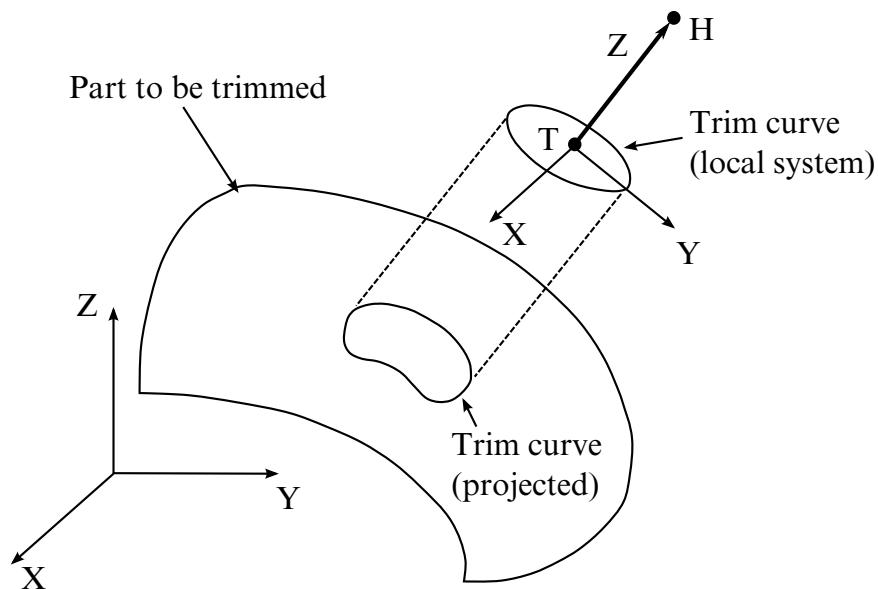


Figure 17-25. Trimming Orientation Vector. The tail (T) and head (H) points define a local coordinate system (x, y, z). The global coordinate system is named (X, Y, Z). The local x -direction is constructed in the Xz -plane. If X and z nearly coincide ($|X \cdot z| > 0.95$), then the local x -direction is instead constructed in the Yz -plane. Trim curve data is input in the xy -plane and projected in the z -direction onto the deformed mesh to obtain the trim line.

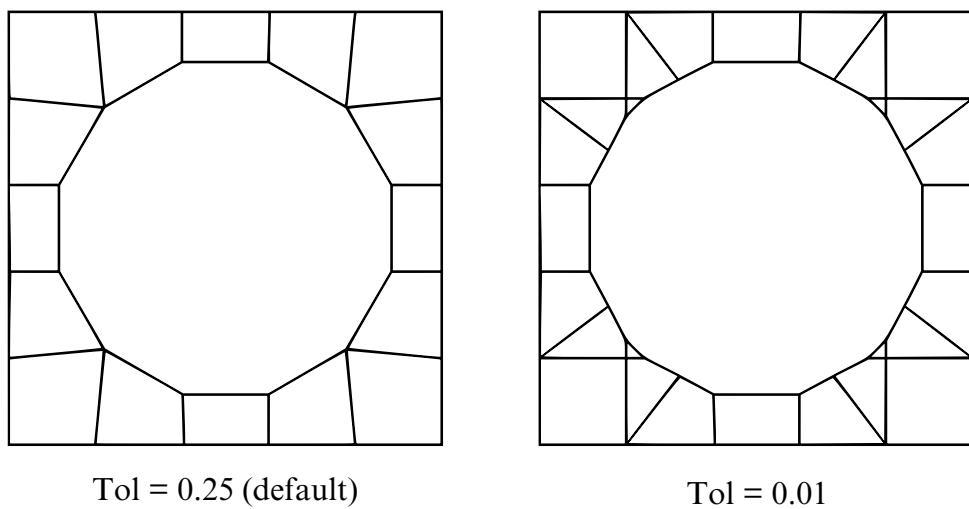


Figure 17-26. Trimming Tolerance. The tolerance limits the size of the small elements generated during trimming. The default tolerance (left) produces large elements. Using a tolerance of 0.01 (right) allows smaller elements and more detail in the trim line.

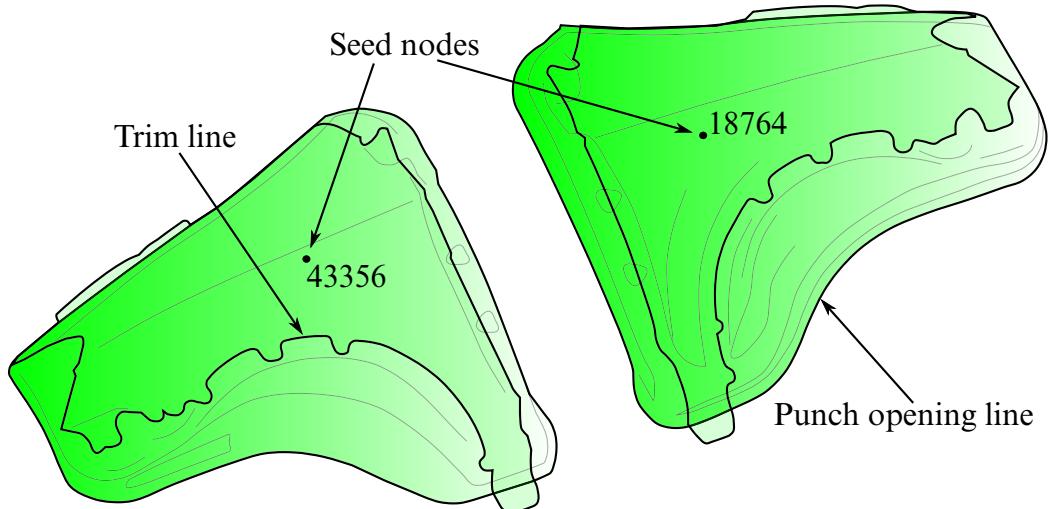


Figure 17-27. Trimming of a double-attached part (NUMISHEET2002 Fender Outer).

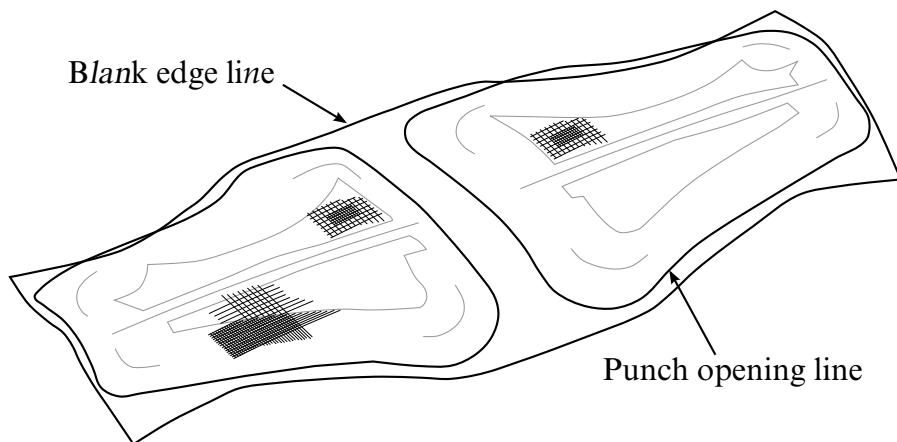


Figure 17-28. The fender outer (draw complete) in wireframe mode.

***DEFINE**

***DEFINE_CURVE_TRIM**

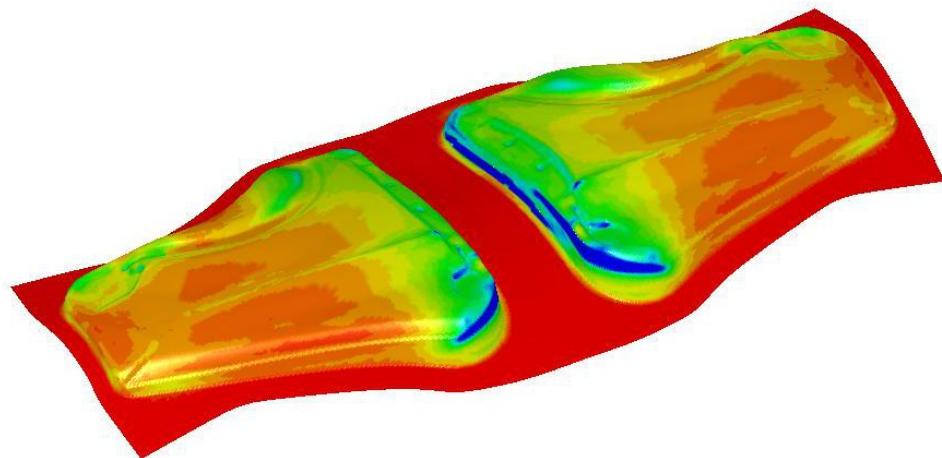


Figure 17-29. The fender outer - thickness/thinning plot on the drawn panel.

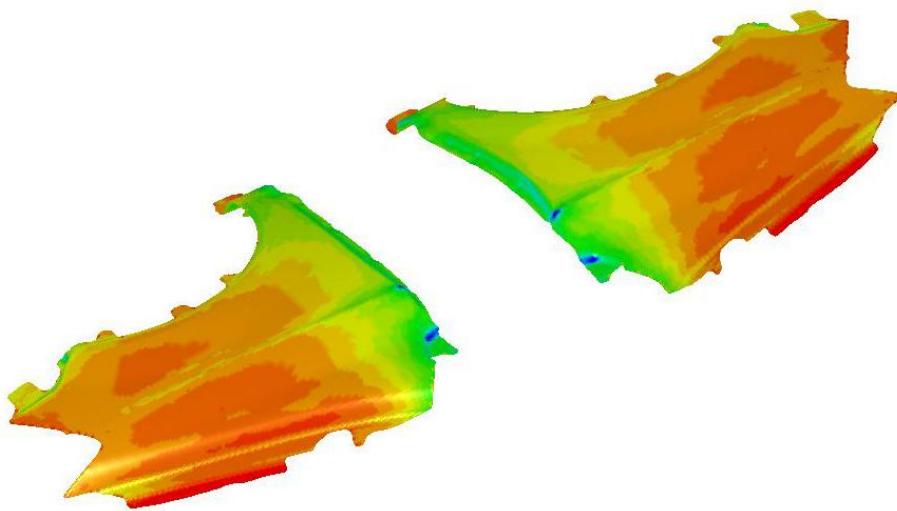


Figure 17-30. The fender outer trim complete using the NSEED1/NSEED2 feature.

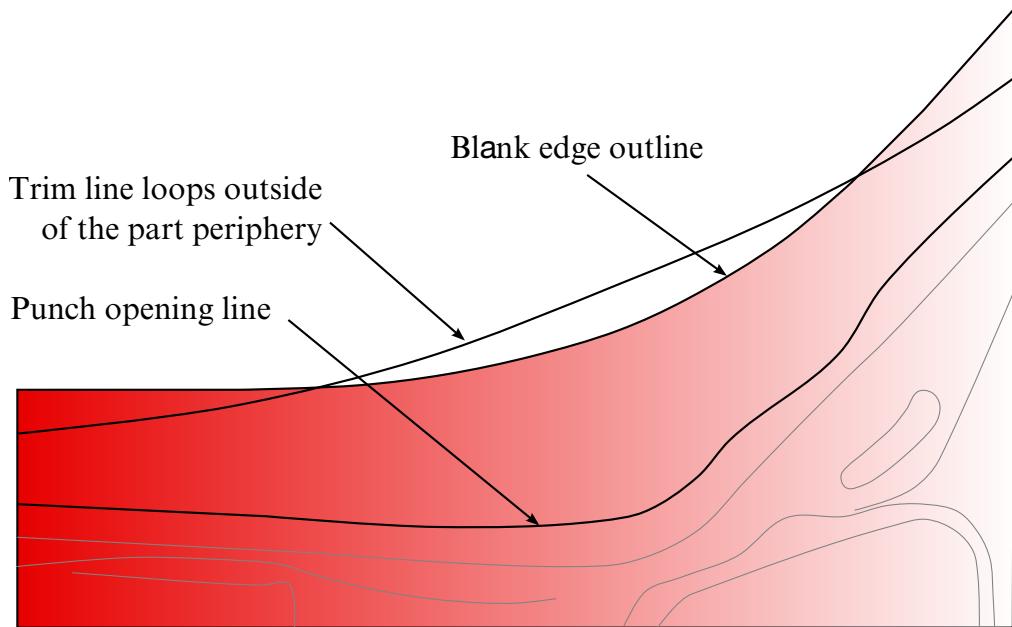


Figure 17-31. Revision 68643 deals with trim curves going beyond part boundary.

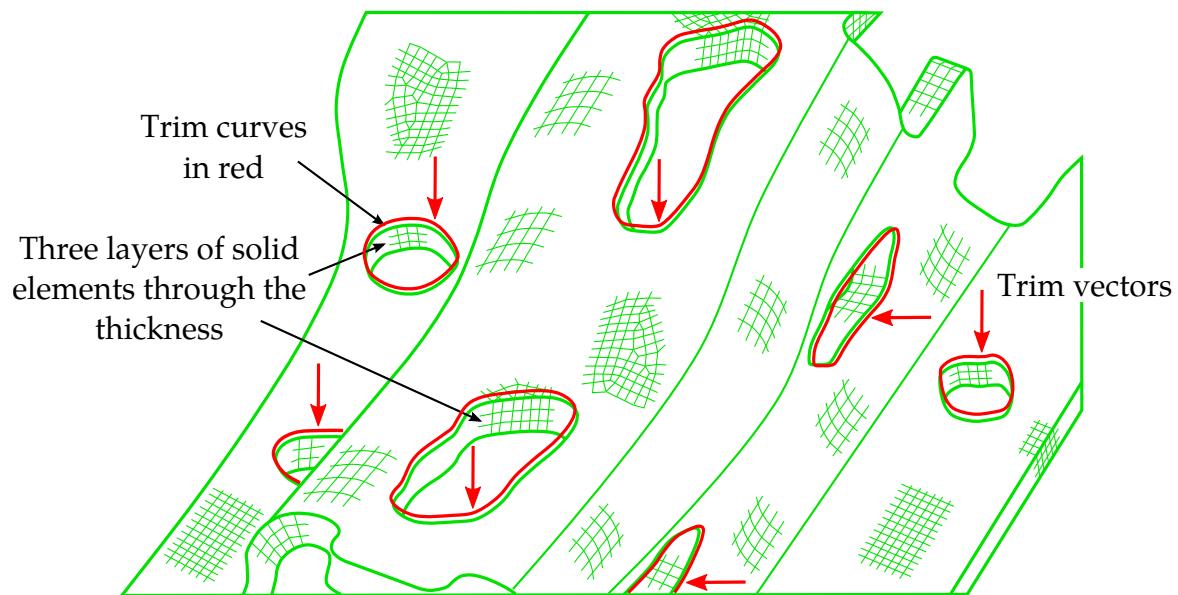


Figure 17-32. 2D trimming of solids using ***DEFINE_CURVE_TRIM_2D**

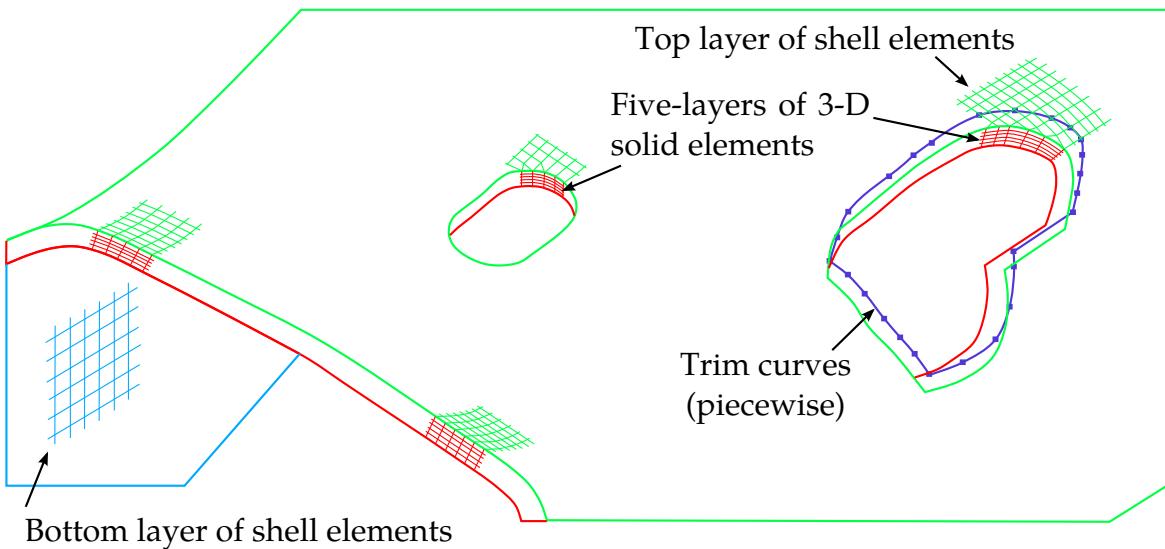
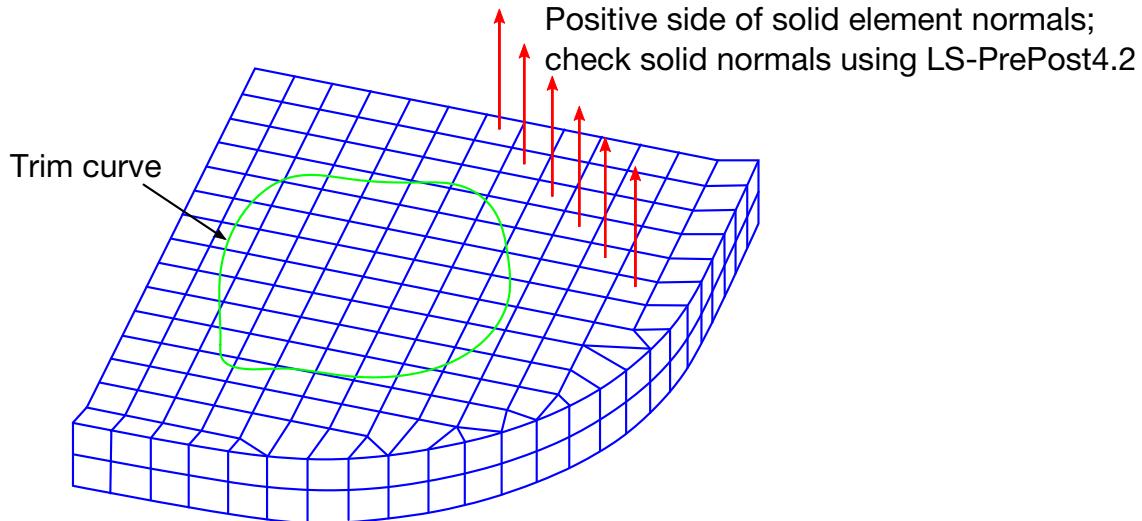
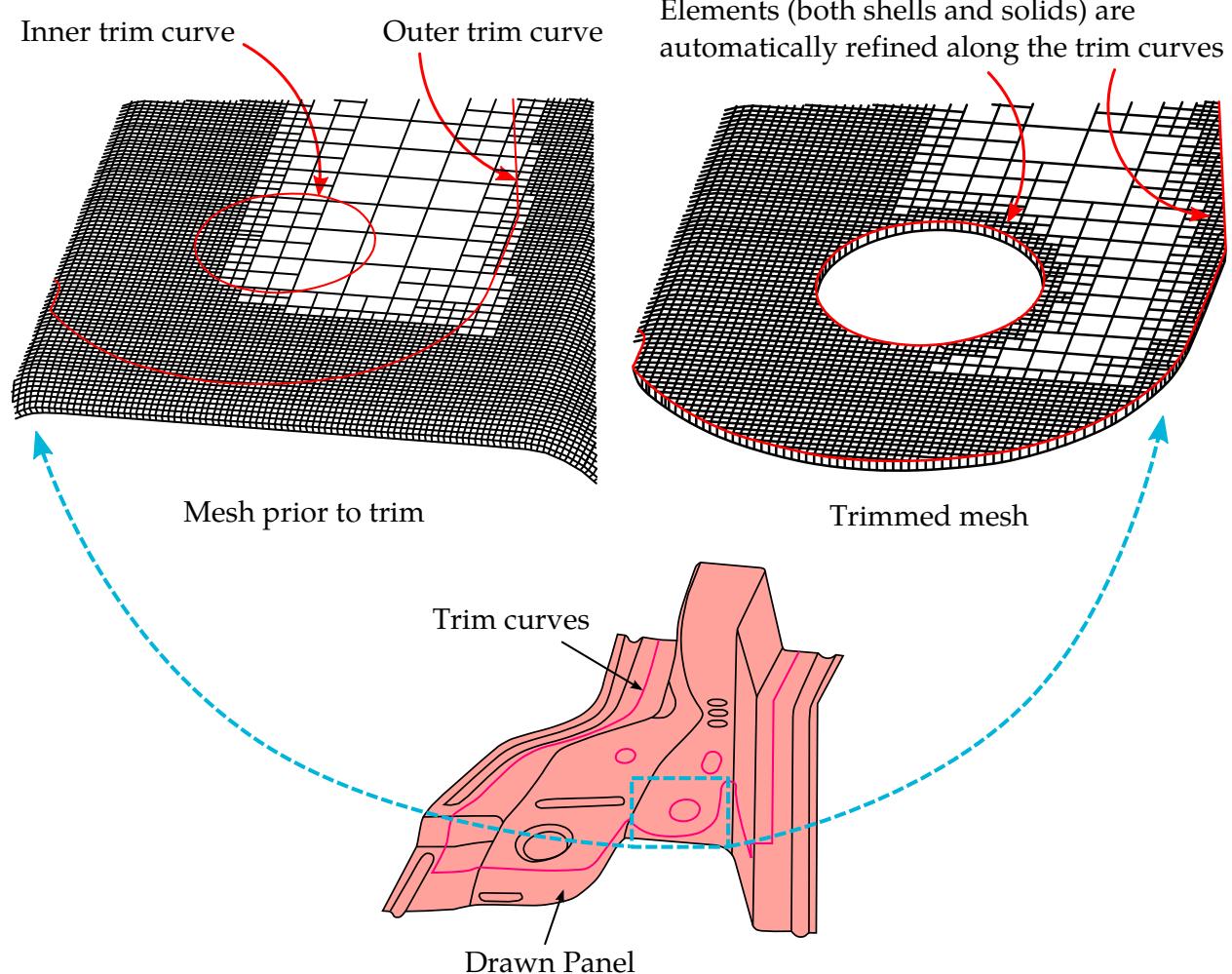


Figure 17-33. 3D trimming of laminates (a core of multiple-layers of solid elements with top and bottom layers of shell elements) using *DEFINE_CURVE_TRIM_3D. Note that shell elements must share the same nodes with the solid elements at the top and bottom layer.



All solid element normals must be consistent. If trim curve is close to the positive normal side, set TDIR=1; otherwise set TDIR=-1. Respacing the curve with more points, project the respaced curve to the top or bottom solid surface may help the trimming.

Figure 17-34. Define trim curve for 3D trimming of solid and laminates.



2005 NUMISHEET Cross Member -
Drawn Panel and Trim Curves

Figure 17-35. Trimming of an adaptive mesh on a sandwiched part. Meshes are automatically refined along the trim curves. Note that only one layer of solid element as a core is allowed for adaptive-meshed sandwich parts without CONTROL_FORMING_TRIMMING_SOLID_REFINEMENT. Also, the top and bottom layer of shells must share the same nodes as the solid elements.

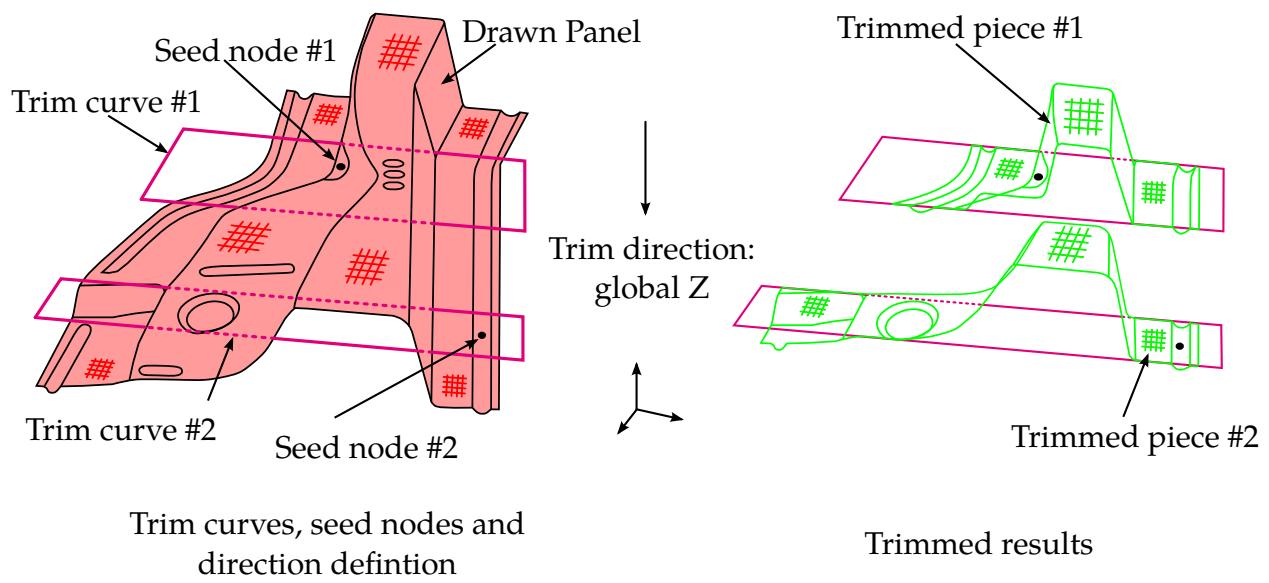


Figure 17-36. An example of 2D trimming of sandwiched part (laminates) from 2005 NUMISHEET benchmark - cross member.

***DEFINE_DE_ACTIVE_REGION**

Purpose: To define an interested region for Discrete Elements (DE) for high efficiency collision pair searching. Any DE leaving this domain will not be considered in the future DE searching and will also be disabled in the contact algorithm.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	TYPE	XM/R	YM	ZM	TBIRTH	TDEATH	NFREQ
Type	I	I	F	F	F	F	F	I
Default	none	0	0.	0.	0.	0.	10^{20}	1

VARIABLE	DESCRIPTION
ID	Set ID/Box ID/Node ID
TYPE	Type for ID: EQ.0: Part set ID EQ.1: Box ID EQ.2: Node ID (moving sphere) EQ.3: Node ID (box-shaped region that moves with this node)
XM, YM, ZM	For TYPE = 0 or 1, factor for region's margin on each direction based on region length. The static coordinates limits are determined either by the part set or box. To provide a buffer zone, these factors, XM, YM, and ZM, can be used. The margin in each direction is calculated in the following way:

Let X_{\max} and X_{\min} be the limits in the x -direction given by the set or box. Then,

$$\Delta X = X_{\max} - X_{\min} .$$

The margin is, then, computed from the input as,

$$X_{\text{margin}} = XM \times \Delta X$$

Then the corresponding limits for the active region are,

$$X'_{\max} = X_{\max} + X_{\text{margin}}$$

$$X'_{\min} = X_{\min} - X_{\text{margin}}$$

DEFINE**DEFINE_DE_ACTIVE_REGION**

VARIABLE	DESCRIPTION
	For TYPE = 3, the region is a box that moves with the node. The node is also the region's center. The location of the region is updated every NFREQ cycles. XM, YM, and ZM give the distance in each direction that the region extends from the node. For instance, let X_{node} be the position of the node in the x -direction. Let X_{max} and X_{min} be minimum and maximum x values defining the region. X_{max} and X_{min} are then calculated as:
	$X_{max} = X_{node} + XM$ $X_{min} = X_{node} - XM$
R	Radius of the region which is centered at the Node ID for TYPE = 2
TBIRTH, TDEATH	Birth and death times for the active region when Node ID is used (TYPE = 2)
NFREQ	Number of cycles between updates of the region's location for TYPE = 3 (default = 1)

DEFINE_DE_BOND**DEFINE*****DEFINE_DE_BOND**

Purpose: Define a bond model for bonds between discrete element spheres (DES). Note that [*DEFINE_DE_BOND_OVERRIDE](#) may override the bond properties between DES.

Card Summary:

Card 1. This card is required.

SID	STYPE	BDFORM	IDIST	MAXCN	BINARY		
-----	-------	--------	-------	-------	--------	--	--

Card 2. This card is required.

PBN	PBS	PBNS	PBSS	SFA	ALPHA	BENDSF	MAXGAP
-----	-----	------	------	-----	-------	--------	--------

Card 3a. Include this card when BDFORM = 3.

ARSP	BRSP						
------	------	--	--	--	--	--	--

Card 3b. Include this blank card when IDIST \neq 0 and BDFORM \neq 3.

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

Card 4a. Include this card for IDIST = -1 or 1.

PBN_SD	PBS_SD	PBNS_SD	PBSS_SD				
--------	--------	---------	---------	--	--	--	--

Card 4b. Include this card for IDIST = -2 or 2.

MPBN	MPBS	MPBNS	MPBSS				
------	------	-------	-------	--	--	--	--

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	SID	STYPE	BDFORM	IDIST	MAXCN	BINARY		
Type	I	I	I	I	I	I		
Default	none	0	1	0	0	0		

VARIABLE**DESCRIPTION**

SID Node set, part set, or part ID for which bond properties apply

DEFINE**DEFINE_DE_BOND**

VARIABLE	DESCRIPTION
STYPE	SID type: EQ.0: DES node set EQ.2: DES part set EQ.3: DES part
BDFORM	Bond formulation: EQ.1: Linear bond formulation (default) EQ.3: Bond strength that increases linearly with deformation rate. See Remark 3 .
IDIST	Distribution of bond properties (see Remarks 4 and 5): EQ.-2: Weibull distribution (non-deterministic) EQ.-1: Gaussian distribution (non-deterministic) EQ.0: Single property (default) EQ.1: Gaussian distribution (deterministic) EQ.2: Weibull distribution (deterministic)
MAXCN	Maximum coordination number for outputting the DES particles that belong to a bond. If the coordination number of a DES particle that belongs to a bond is less than or equal to this value, the DES particle is output at the initial time state. EQ.0: Do not print out any bonded DES particles at the initial time state.
BINARY	Output file type for printing the bonded DES particle if the coordination number is less than or equal to MAXCN: EQ.0: Do not write the data to a file. EQ.1: Write an ASCII file. EQ.2: Write the data to binary database binout. EQ.3: Write the data to an ASCII file and to the binary database binout.

Bond Formulation Card.

Card 2	1	2	3	4	5	6	7	8
Variable	PBN	PBS	PBNS	PBSS	SFA	ALPHA	BENDSF	MAXGAP
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	1.0	0.0	1.0	10^{-4}

VARIABLE	DESCRIPTION
PBN	Parallel-bond modulus [Pa]. See Remarks 1 and 2 . For IDIST = -1 and 1, this value is the mean. For IDIST = -2 and 2 with MPBN ≠ 0, this value is the scale parameter.
PBS	Parallel-bond stiffness ratio, shear stiffness/normal stiffness. See Remark 2 . For IDIST = -1 and 1, this value is the mean. For IDIST = -2 and 2 with MPBS ≠ 0, this value is the scale parameter.
PBNS	Parallel-bond maximum normal stress. A zero value defines an infinite maximum normal stress. For IDIST = -1 and 1, this value is the mean. For IDIST = -2 and 2 with MPBNS ≠ 0, this value is the scale parameter.
PBSS	Parallel-bond maximum shear stress. A zero value defines an infinite maximum shear stress. For IDIST = -1 and 1, this value is the mean. For IDIST = -2 and 2 with MPBSS ≠ 0, this value is the scale parameter.
SFA	Bond radius multiplier
ALPHA	Numerical damping, $0.0 \leq \text{ALPHA} \leq 1.0$
BENDSF	Influence of bending/twisting on the bond failure criteria (see Remark 6).
	EQ.-1.0: No bending/twisting is considered in the bond failure criteria.
	EQ.0.0: Defaults to 1.0.
	GT.0.0: Scale factor for the bending/twisting component in the bond failure criteria.
MAXGAP	Maximum gap between two bonded spheres:

DEFINE**DEFINE_DE_BOND**

VARIABLE	DESCRIPTION
	GT.0.0: Defines the ratio of the smaller radius of two bonded spheres as the maximum gap, that is, MAX-GAP $\times \min(r_1, r_2)$.
	LT.0.0: Absolute value is used as the maximum gap.

Rate Sensitivity Parameters Card (BDFORM = 3). Include this card when BDFORM = 3.

Card 3a	1	2	3	4	5	6	7	8
Variable	ARSP	BRSP						
Type	F	F						
Default	0.0	0.0						

VARIABLE	DESCRIPTION
ARSP	Deformation rate sensitivity parameter for parallel-bond maximum normal stress. A zero value ignores deformation rate effects on bond strength.
BRSP	Deformation rate sensitivity parameter for parallel-bond maximum shear stress. A zero value ignores deformation rate effects on bond strength. t

Include this blank card when BDFORM \neq 3 and IDIST \neq 0.

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

DEFINE_DE_BOND**DEFINE**

Standard Deviation for Gaussian Distribution Card. Include this card if IDIST = -1 or 1.

Card 4a	1	2	3	4	5	6	7	8
Variable	PBN_SD	PBS_SD	PBNS_SD	PBSS_SD				
Type	F	F	F	F				
Default	0.0	0.0	0.0	0.0				

VARIABLE	DESCRIPTION
PBN_SD	Standard deviation for parallel-bond modulus. PBN on Card 2 is the mean. EQ.0.0: Uniform distribution
PBS_SD	Standard deviation for parallel-bond stiffness ratio, shear stiffness/normal stiffness. PBS on Card 2 is the mean. EQ.0.0: Uniform distribution
PBNS_SD	Standard deviation for parallel-bond maximum normal stress. PB-NS on Card 2 is the mean. EQ.0.0: Uniform distribution
PBSS_SD	Standard deviation for parallel-bond maximum shear stress. PBSS on Card 2 is the mean. EQ.0.0: Uniform distribution

Shape Parameters for Weibull Distribution Card. Include this card if IDIST = -2 or 2.

Card 4b	1	2	3	4	5	6	7	8
Variable	MPBN	MPBS	MPBNS	MPBSS				
Type	F	F	F	F				
Default	0.0	0.0	0.0	0.0				

DEFINE**DEFINE_DE_BOND**

VARIABLE	DESCRIPTION
MPBN	Shape parameter (heterogeneity index) for parallel-bond modulus. PBN on Card 2 is the associated scale parameter. EQ.0.0: No distribution. PBN is the parallel-bond modulus.
MPBS	Shape parameter (heterogeneity index) for parallel-bond stiffness ratio, shear stiffness/normal stiffness. PBS on Card 2 is the associated scale parameter. EQ.0.0: No distribution. PBS is the parallel-bond stiffness ratio.
MPBNS	Shape parameter (heterogeneity index) for parallel-bond maximum normal stress. PBNS on Card 2 is the associated scale parameter.. EQ.0.0: No distribution. PBNS is the parallel-bond maximum normal stress.
MPBSS	Shape parameter (heterogeneity index) for parallel-bond maximum shear stress. PBSS on Card 2 is the associated scale parameter. EQ.0.0: No distribution. PBSS is the parallel-bond maximum shear stress.

Remarks:

1. **Normal force.** The normal force between two bonded discrete elements with radii r_1 and r_2 is calculated as

$$\Delta f_n = \frac{PBN}{(r_1 + r_2)} \times A \times \Delta u_n ,$$

where

$$A = \pi r_{\text{eff}}^2$$

$$r_{\text{eff}} = \min(r_1, r_2) \times SFA$$

2. **Shear force.** The shear force is calculated as

$$\Delta f_s = PBS \times \frac{PBN}{(r_1 + r_2)} \times A \times \Delta u_s .$$

3. **Linearly increasing bond strength with increasing loading rate.** With BD-FORM = 3, bond strength increases linearly with an increased loading rate. The following equations give the maximum normal and shear stresses:

$$\sigma_n(v_{\text{rel}}) = \text{PBNS} + \text{ARSP} \times \frac{v_{\text{rel}}}{r_1 + r_2}$$

$$\tau(v_{\text{rel}}) = \text{PBSS} + \text{BRSP} \times \frac{v_{\text{rel}}}{r_1 + r_2}$$

Here PBNS and PBSS correspond to the maximum bond strength in quasistatic loading, ARSP and BRSP are the rate sensitivity parameters, and v_{rel} is the relative velocity between two bonded particles with radii r_1 and r_2 .

4. **Distributions of bond properties.** IDIST determines the distribution of bond properties for the DES in SID. It specifically affects the parallel-bond modulus, stiffness ratio, maximum normal stress, and maximum shear stress. IDIST = 0 means all the DES in SID have the same bond properties.

The distribution of bond properties for IDIST = -1 and 1 follows a Gaussian distribution:

$$f(x|\text{prop_mean}, \text{prop_sd}) = \frac{1}{\text{prop_sd}\sqrt{2\pi}} e^{\frac{(x-\text{prop_mean})^2}{2\times\text{prop_sd}^2}},$$

where prop_mean is the mean value of the property, prop_sd is the standard deviation, and x is the statistical distribution of the property. For this distribution, you input the following pairs to specify the mean and standard deviation for the four bond properties: (PBN, PBN_SD), (PBS, PBS_SD), (PBNS, PBNS_SD), and (PBSS, PBSS_SD).

The distribution of bond properties for IDIST = -2 and 2 follows a Weibull distribution:

$$f(n|\text{prop_sf}, \text{mprop}) = \frac{\text{mprop}}{\text{prop_sf}} \left(\frac{n}{\text{prop_sf}} \right)^{\text{mprop}-1} e^{-\left(\frac{n}{\text{prop_sf}} \right)^{\text{mprop}}},$$

where prop_sf is the scale factor for the property, mprop is the shape factor for the property, and n is the statistical distribution of the property. For this distribution, you input the following pairs to specify the scale factor and shape factor for the four bond properties: (PBN, MPBN), (PBS, MPBS), (PBNS, MPBNS), and (PBSS, MPBSS).

5. **Deterministic versus non-deterministic distributions.** We have implemented both deterministic and non-deterministic versions for the distributions of bond properties ($\text{IDIST} \neq 0$). By deterministic, we mean that each run uses the same seed and has the same sequence of numbers that fit the distribution. Thus, multiple runs with the same input deck yield the same results. In contrast, the non-deterministic algorithms obtain a random seed each time. Therefore, each run with the same input deck may have a different solution as the sequence of numbers changes each time.

6. **BENDSF.** BENDSF enables removing or reducing the influence of twisting/bending on the bond failure criteria. The following equations determine bond failure:

$$\bar{\sigma}_{\max} = \frac{-\bar{F}_n}{A} + \text{BENDSF} \times \frac{|\bar{M}_s|r_{\text{eff}}}{I} \geq \text{PBNS}$$
$$\bar{\tau}_{\max} = \frac{-\bar{F}_s}{A} + \text{BENDSF} \times \frac{|\bar{M}_n|r_{\text{eff}}}{J} \geq \text{PBSS}$$

Here, the $\bar{-}$ means average. I and J are the moment of inertia and polar moment of inertia. A and r_{eff} are as described in [Remark 1](#).

DEFINE_DE_BOND_OVERRIDE**DEFINE*****DEFINE_DE_BOND_OVERRIDE**

Purpose: Override the bond values specified with [*DEFINE_DE_BOND](#) for bonds between certain discrete element spheres. This feature allows you to define heterogeneous bonds.

Unless [*DEFINE_DE_BOND_OVERRIDE](#) is required for defining heterogeneous bonds, we recommend not using it. The algorithms invoked by including [*DEFINE_DE_BOND_OVERRIDE](#) require more data passing during the calculation. Thus, it is more computationally expensive.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	STYPE		IRAND				
Type	I	I		I				
Default	none	0		0				

Card 2	1	2	3	4	5	6	7	8
Variable	PBN	PBS	PBN_S	PBS_S	SFA	ALPHA	BENDSF	
Type	F	F	F	F	F	F	F	
Default	none	none	none	none	1.0	0.0	1.0	

Additional card if IRAND = 1

Card 3	1	2	3	4	5	6	7	8
Variable	PBNMAX	PBSMAX	PBN_SMX	PBS_SMX				
Type	F	F	F	F				
Default	none	none	none	none				

DEFINE**DEFINE_DE_BOND_OVERRIDE**

VARIABLE	DESCRIPTION
SID	Node set, part set, or part for which the bond properties will be overridden
STYPE	SID type: EQ.0: DES node set EQ.2: DES part set EQ.3: DES part
IRAND	Random distribution of bond properties: EQ.0: Single property (default) EQ.1: Uniform distribution
PBN	Parallel-bond modulus [Pa]. For IRAND = 1, this is the minimum value in the distribution. See Remarks 1 and 2 .
PBS	Parallel-bond stiffness ratio. shear stiffness/normal stiffness. For IRAND = 1, PBS is the minimum value of the parallel-bond stiffness ratio in the distribution. See Remark 2 .
PBN_S	Parallel-bond maximum normal stress. A zero value defines an infinite maximum normal stress. For IRAND = 1, PBN_S is the minimum value used for the parallel-bond maximum normal stress in the distribution.
PBS_S	Parallel-bond maximum shear stress. A zero value defines an infinite maximum shear stress.
SFA	Bond radius multiplier
BENDSF	Influence of bending/twisting on the bond failure criteria (see Remark 4). EQ.-1.0: No bending/twisting is considered in the bond failure criteria. EQ.0.0: Defaults to 1.0. GT.0.0: Scale factor for the bending/twisting component in the bond failure criteria.
ALPHA	Numerical damping
PBNMAX	Maximum value of PBN. Used for random distribution.

VARIABLE	DESCRIPTION
PBSMAX	Maximum value of PBS. Used for random distribution.
PBN_SMAX	Maximum value of PBN_S. Used for random distribution.
PBS_SMAX	Maximum value of PBS_S. Used for random distribution.

Remarks:

1. **Normal force.** The normal force between two bonded discrete elements with radii r_1 and r_2 is calculated as

$$\Delta f_n = \frac{PBN}{(r_1 + r_2)} \times A \times \Delta u_n ,$$

where

$$A = \pi r_{\text{eff}}^2$$

$$r_{\text{eff}} = \min(r_1, r_2) \times \text{SFA}$$

For IRAND = 1, PBN is selected from the uniform distribution.

2. **Shear force.** The shear force is calculated as

$$\Delta f_s = PBS \times \frac{PBN}{(r_1 + r_2)} \times A \times \Delta u_s .$$

For IRAND = 1, PBN and PBS are selected from their uniform distributions.

3. **Maximum gap between bonded spheres.** The maximum gap between bonded spheres is inherited from MAXGAP on *DEFINE_DE_BOND and thus not specified with this keyword.
4. **BENDSF.** BENDSF enables removing or reducing the influence of twisting/bending on the bond failure criteria. The following equations determine bond failure:

$$\bar{\sigma}_{\max} = \frac{-\bar{F}_n}{A} + \text{BENDSF} \times \frac{|\bar{M}_s|r_{\text{eff}}}{I} \geq \text{PBNS}$$

$$\bar{\tau}_{\max} = \frac{-\bar{F}_s}{A} + \text{BENDSF} \times \frac{|\bar{M}_n|r_{\text{eff}}}{J} \geq \text{PBSS}$$

Here, the $\bar{-}$ means average. I and J are the moment of inertia and polar moment of inertia. A and r_{eff} are as described in [Remark 1](#).

DEFINE**DEFINE_DE_BY_PART*****DEFINE_DE_BY_PART**

Purpose: To define control parameters for discrete element sphere by part ID. This card overrides the values set in [*CONTROL_DISCRETE_ELEMENT](#).

Card 1	1	2	3	4	5	6	7	8
Variable	PID	NDAMP	TDAMP	FRICS	FRICR	NORMK	SHEARK	
Type	I	F	F	F	F	F	F	
Default	none	0.	0.	0.	0.	0.01	2/7	

Card 2	1	2	3	4	5	6	7	8
Variable	GAMMA	VOL	ANG					
Type	F	F	F					
Default	0.	0.	0.					

Card 3 is optional.

Card 3	1	2	3	4	5	6	7	8
Variable	LNORM	LSHEAR		FRICD	DC			
Type	I	I		F	F			
Default	0.	0.		FRICS	0.			

VARIABLE**DESCRIPTION**

PID Part ID of DES nodes

NDAMP Normal damping coefficient

TDAMP Tangential damping coefficient

VARIABLE	DESCRIPTION
FRICS	Static coefficient of friction (see Remarks): EQ.0.0: 3 DOF NE.0.0: 6 DOF (consider rotational DOF)
FRICR	Rolling friction coefficient
NORMK	Optional scale factor of normal spring constant (Default = 0.01)
SHEARK	Optional ratio between SHEARK/NORMK (Default = 2/7)
GAMMA	Liquid surface tension
VOL	Volume fraction
ANG	Contact angle
LNORM	Load curve ID of a curve that defines normal stiffness as a function of norm penetration ratio
LSHEAR	Load curve ID of a curve that defines shear stiffness as a function of norm penetration ratio
FRICD	Dynamic coefficient of friction. By default, FRICD = FRICS. See Remarks .
DC	Exponential decay coefficient. See Remarks .

Remarks:

The frictional coefficient is assumed to be dependent on the relative velocity v_{rel} of the two DEM in contact

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

DEFINE**DEFINE_DE_COHESIVE*****DEFINE_DE_COHESIVE**

Purpose: Define a cohesive force mode for discrete element spheres. This keyword overrides the values set in *CONTROL_DISCRETE_ELEMENT.

Card 1	1	2						
Variable	SID	STYPE						
Type	I	I						
Default	0	0						

Card 2	1	2	3	4				
Variable	GAMMA	VOL	ANG	GAP				
Type	F	F	F	F				
Default	0.	0.	0.	0.				

VARIABLE	DESCRIPTION
SID	Node set ID, part set ID or part ID defining DES with cohesive force
STYPE	SID type: EQ.0: node set EQ.1: part set EQ.2: part
GAMMA	Liquid surface tension
VOL	Volume fraction
ANG	Contact angle
GAP	Spatial limit for the existence of liquid bridge between particles. A liquid bridge will exist when the distance between two particles is less or equal to $\min(\text{GAP}, d_{\text{rup}})$ where d_{rup} is the rupture distance

VARIABLE**DESCRIPTION**

of the bridge automatically calculated by LS-DYNA.

EQ.0.0: maximum radius of DES

NE.0.0: value of spatial limit

*DEFINE

*DEFINE_DE_FLOW_DRAG

*DEFINE_DE_FLOW_DRAG

Purpose: Apply drag force to discrete element spheres (DES).

Card Summary:

Card 1. This card is required.

CD	RHO	MU	VX	VY	VZ	TBIRTH	TDEATH
----	-----	----	----	----	----	--------	--------

Card 2. This card is required.

VS	DFLAG	SFN	SFS				
----	-------	-----	-----	--	--	--	--

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	CD	RHO	MU	VX	VY	VZ	TBIRTH	TDEATH
Type	F	F	F	F	F	F	F	F
Default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	10^{20}

<u>VARIABLE</u>	<u>DESCRIPTION</u>
CD	Drag coefficient, C_d EQ.-1: C_d determined based on Morrison 2013. See Remark 2 . EQ.-2: C_d determined based on Allen 2018. See Remark 2 . Card 1.1 must be included.
RHO	Flow density, ρ
MU	Dynamic viscosity
VX, VY, VZ	Flow velocity
TBIRTH	Birth time
TDEATH	Death time

Card 2	1	2	3	4	5	6	7	8
Variable	VS	DFLAG	SFN	SFS				
Type	F	I	F	F				
Default	0.0	1	1.0	1.0				

VARIABLE	DESCRIPTION
VS	Sound speed, V_s for CD = -2. See Remark 2 .
DFLAG	Influence of neighbors on drag treatment: EQ.1: Consider only shadowing effect (default) EQ.2: See Remark 3 . EQ.3: See Remark 3 .
SFN, SFS	Scale factors for C_{d1_eff} and C_{d2_eff} respectively when DFLAG = 2.

Remarks:

1. **Drag Force.** The drag force is calculated as:

$$f_{\text{drag}} = \frac{1}{2} C_d \rho v^2 \frac{\pi D^2}{4} .$$

Here C_d is the drag coefficient and D is the diameter of the DES.

2. **Drag Coefficient.** For CD = -1 and -2, the drag coefficient is found using predefined correlations.

If CD = -1, the drag coefficient is determined by the following correlation (Morrison 2013):

$$C_d = \frac{24}{Re} + \frac{2.6 \left(\frac{Re}{5.0} \right)}{1 + \left(\frac{Re}{5.0} \right)^{1.52}} + \frac{0.411 \left(\frac{Re}{2.63 \times 10^5} \right)^{-7.94}}{1 + \left(\frac{Re}{2.63 \times 10^5} \right)^{-8.00}} + \frac{0.25 \frac{Re}{10^6}}{1 + \frac{Re}{10^6}} .$$

Re is the Reynolds number which for a sphere is defined as:

$$Re = \frac{\rho v D}{\mu} .$$

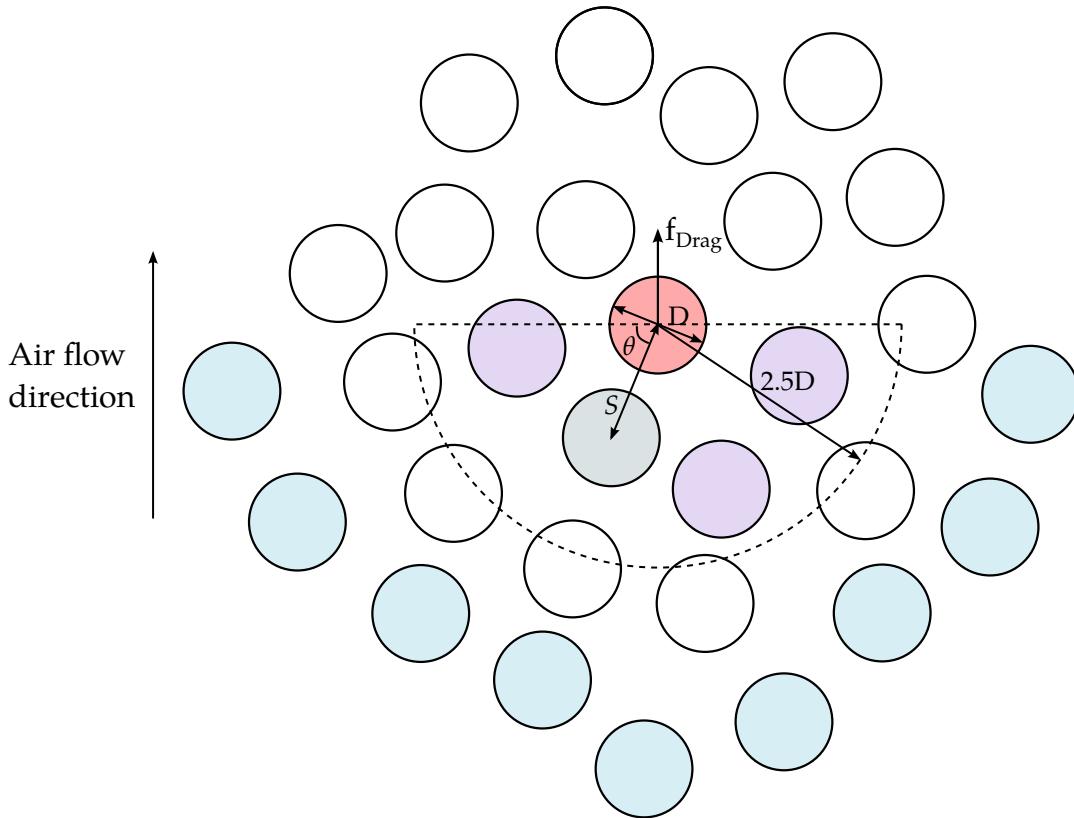


Figure 17-37. Schematic of how neighboring particles affect the drag treatment for DFLAG = 2 and 3. In this schematic the particles in blue are the leading particles that move at supersonic speed. We have developed two different two different methods for how the coefficient of drag is influenced by neighboring particles for particles that are not leading particles. In both methods, LS-DYNA checks for particles in the hemisphere in front of a given particle. In this schematic, the given particle is in red. The neighboring particles in the hemisphere are purple and grey. For DFLAG = 2, the grey particle is the closest neighbor of the red particle.

where ρ is the fluid density, v is the flow speed, D is the particle diameter, and μ is the dynamic viscosity of the fluid. C_d as obtained from this correlation is valid between $0.1 < \text{Re} < 10^6$.

If CD = -2, the drag coefficient is determined by the following correlation (Allen 2018):

$$C_d = \begin{cases} 0.995 & \text{for } M \geq 2.0 \\ 0.920 + 0.0375M & \text{for } 1.2 \leq M < 2.0 \\ -0.163 + 0.9400M & \text{for } 0.7 \leq M < 1.2 \\ 0.418 + 0.1100M & \text{for } 0.2 \leq M < 0.7 \\ 0.440 & \text{for } M < 0.2 \end{cases}$$

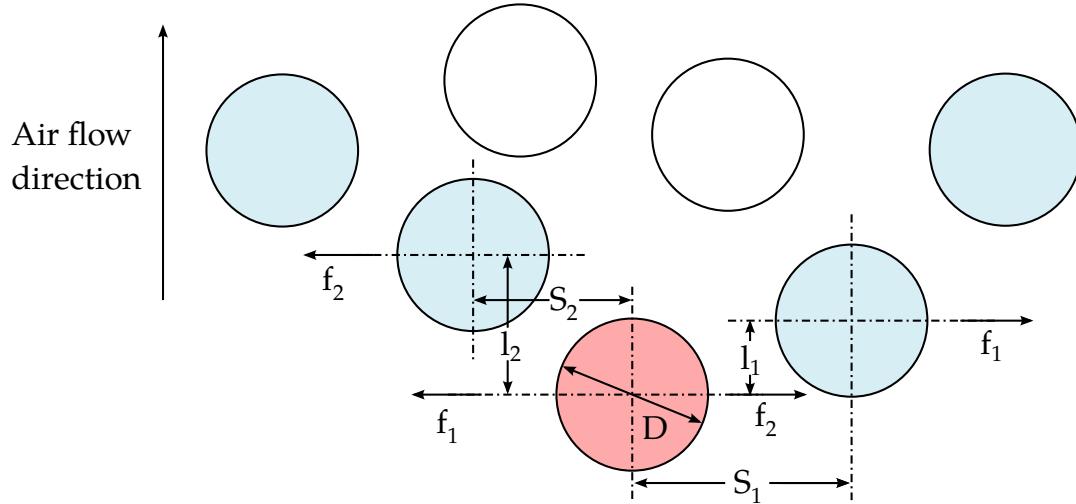


Figure 17-38. Illustration of perpendicular drag force in the lead particles for DFLAG = 2 and 3. In this schematic, the red particle interacts with both of its neighbors.

Here, M is the Mach number defined as:

$$M = \frac{V}{V_s},$$

where V is the pellet velocity and V_s is the sound speed.

3. **Influence of Neighbors on Drag Treatment.** The influence of neighboring particles on the drag treatment of a DES depends on the setting of DFLAG. By default, LS-DYNA only considers the shadowing effect (DFLAG = 1). We have implemented two other treatments.

For DFLAG = 2 and 3, consider a group of particles as shown in [Figure 17-37](#). The particles in blue are the leading particles in the flow. These particles travel with supersonic speed. They create a shock wave in front of them. Because of the shock wave, the leading particles may experience drag side forces perpendicular to the traveling direction from their neighboring leading particles (see [Figure 17-38](#)). For the particles in the layers behind the leading particles, their drag coefficient may be influenced by the particles in front of them. DFLAG = 2 and 3 are intended to model these two phenomena.

For both DFLAG = 2 and 3, we model the perpendicular side force with the same equation. These side forces are only from leading particle interactions and thus only apply to leading particles. The side drag coefficient is given by:

$$\frac{C_{d\text{perp_eff}}}{C_d} = \left(7.20e^{-2.644\frac{S}{D}} \right) \left(1 - \frac{l}{D} \right), \text{ for } l \leq D \text{ and } S < 2.5D$$

C_d in this equation is the normal drag coefficient, and D is the diameter of the particle. l and S are as indicated in [Figure 17-38](#), where l_1 and l_2 are examples of l , and S_1 and S_2 are examples of S .

For the drag force in the traveling direction, the coefficient of drag for the leading particles will be CD as specified in Card 1. The leading particles have the largest drag coefficient. The coefficient of drag for all the other particles are modified based on the value of DFLAG. For this modification, LS-DYNA looks at the hemisphere with radius $2.5D$ in front of a given particle in the flow as shown in [Figure 17-37](#).

- a) $DFLAG = 2$. The coefficient of drag is first determined with the following equation:

$$\frac{C_{dtrav_eff}}{C_d} = \left(1.0 + 4.644e^{-2.225\frac{S}{D}} \right) \cos \theta + \left(1 - e^{-0.5\frac{S}{D}} \right) \sin \theta, \text{ for } S < 2.5D$$

S is the distance from the center of the particle to its closest neighbor in the hemisphere. In [Figure 17-37](#), the particle of interest is red and its closest neighbor is grey. θ is defined as shown in the figure. C_d is the coefficient given in CD. This coefficient is then scaled down by considering the number of particles, NHP, in the hemisphere to get the traveling direction coefficient of drag:

$$C_{dtrav} = \max(0, 1.0 - 0.02 \times NHP) C_{dtrav_eff}$$

Note that if the particle has no neighboring particles, then the coefficient of drag is that given by Card 1.

- b) $DFLAG = 3$. The coefficient of drag is determined with the following equation (Numerical Investigation of Drag Forces on Particle Clouds in Non-Newtonian Flow, Efe Kinaci, Johannes Kepler University Linz, September 2015):

$$\frac{C_{dtrav_eff}}{C_d} = \alpha^{1-\chi}, \text{ for } S < 2.5D$$

Here,

$$C_d = \left(0.63 + \frac{4.8}{Re_i^{0.5}} \right)^2$$

$$\chi = 3.7 - 0.65e^{\frac{(1.5 - \log_{10} Re_i)^2}{2}}$$

$$Re_i = \frac{\alpha \rho v D}{\mu}$$

In the above equations, α is the void fraction, ρ is the fluid density, v is the flow velocity, D is the diameter of the particle, and μ is dynamic viscosity of the fluid.

If there is no neighboring particle within $2.5D$, the drag coefficient is based on the logic of $CD = -1$.

*DEFINE

*DEFINE_DE_HBOND

*DEFINE_DE_HBOND

Purpose: To define a heterogeneous bond model for discrete element sphere (DES). (This command, along with *INTERFACE_DE_HBOND, are no longer being developed.)

Card Summary:

Card 1. This card is required

SID	STYPE	HBDFM	IDIM				
-----	-------	-------	------	--	--	--	--

Card 1.1. This card is included if and only if HBDFM = 2.

PBK_SF	PBS_SF	FRGK	FRGS	BONDR	ALPHA	DMG	FRMDL
--------	--------	------	------	-------	-------	-----	-------

Card 2. This card is optional.

PRECRK	CKTYPE		ITFID				
--------	--------	--	-------	--	--	--	--

Data Cards:

Card 1	1	2	3	4	5	6	7	8
Variable	SID	STYPE	HBDFM	IDIM				
Type	I	I	I	I				
Default	none	0	1	3				

VARIABLE	DESCRIPTION
SID	DES nodes
STYPE	Set type for SID: EQ.0: DES node set EQ.1: DES node EQ.2: DES part set EQ.3: DES part

DEFINE_DE_HBOND**DEFINE**

VARIABLE	DESCRIPTION
HBDFM	Bond formulation: EQ.1: (Reserved) EQ.2: Nonlinear heterogeneous bond formulation for fracture analysis based on the general material models defined in the material cards. DES elements with different material models can be defined within one bond.
IDIM	Space dimension for DES bonds: EQ.2: For 2D plane strain problems EQ.3: For 3D problems

Nonlinear Heterogeneous Bond Card. Include this card if and only if HBDFM = 2.

Card 1.1	1	2	3	4	5	6	7	8
Variable	PBK_SF	PBS_SF	FRGK	FRGS	BONDR	ALPHA	DMG	FRMDL
Type	F	F	F	F	F	F	F	I
Default	1.0	1.0	none	none	none	0.0	1.0	1

VARIABLE	DESCRIPTION
PBK_SF	Scale factor for volumetric stiffness of the bond.
PBS_SF	Scale factor for shear stiffness of the bond.
FRGK	Critical fracture energy release rate for volumetric deformation due to the hydrostatic pressure. EQ.0: A zero value specifies an infinite energy release rate for unbreakable bonds. LT.0: A negative value defines the energy release rate under volumetric compression (i.e. positive pressure) and FRGS defined below is used under volumetric expansion (i.e. negative pressure).
FRGS	Critical fracture energy release rate for shear deformation. EQ.0: A zero value specifies an infinite energy release rate

DEFINE**DEFINE_DE_HBOND**

VARIABLE	DESCRIPTION
	for unbreakable bonds.
	FRGK.LT.0: See description for FRGK.
BONDR	Influence radius of the DES nodes.
ALPHA	Numerical damping
DMG	Continuous parameter for damage model. EQ.1.0: The bond breaks if the fracture energy in the bond reaches the critical value. Micro-damage is not calculated. GT.0.5.and.LT.1.0: Microdamage effects begin once the fracture energy reaches $DMG \times FMG[K,S]$. Upon the onset of microdamage, the computed damage ratio will increase (monotonically) as the fracture energy grows. Bond weakening from microdamage is modeled by reducing the bond stiffness in proportion to the damage ratio.
FRMDL	Fracture model: EQ.1: Fracture energy of shear deformation is calculated based on deviatoric stresses. EQ.2: Fracture energy of shear deformation is calculated based on deviatoric stresses, <i>excluding the axial component (along the bond)</i> . EQ.3,4: Same as 1&2, respectively, but FRGK and FRGS are read as the total failure energy density and will be converted to the corresponding critical fracture energy release rate. The total failure energy density is calculated as the total area under uniaxial tension stress-strain curve. EQ.5,6: Same as 3&4, respectively, as FRGK and FRGS are read as the total failure energy density but will not be converted. Instead, the failure energy within the bond will be calculated. Models 1&2 are more suitable for brittle materials, and Models 5&6 are easier for ductile materials. Models 3&4 can be used for moderately ductile fracture accordingly. This is the default fracture model applied to all DES parts, even if they have different material models. More fracture models can be

VARIABLE	DESCRIPTION							
	defined for different materials by specifying an interface ID (ITFID) in the optional card.							

Pre-Crack Card. This card is optional.

Card 2	1	2	3	4	5	6	7	8
Variable	PRECRK	CKTYPE		ITFID				
Type	I	I		I				
Default	none	0		0				

VARIABLE	DESCRIPTION
PRECRK	Shell set, define 3D surfaces of the pre-crack
CKTYPE	Type of PRECRK: EQ.0: Part set EQ.1: Part
ITFID	ID of the interface *INTERFACE_DE_HBOND, which defines different failure models for the heterogeneous bonds within each part and between two parts, respectively.

*DEFINE

*DEFINE_DE_INJECT_BONDED

*DEFINE_DE_INJECT_BONDED {OPTION}

Available options include:

<BLANK>

ELLIPSE

Purpose: This keyword injects bonded discrete element spheres (DES) from a specified region at a flow rate given by a user defined curve. Like *DEFINE_DE_INJECTION, when the option is blank, the region from which the DES emanate is assumed rectangular; the ELLIPSE option indicates that the region is elliptical. The first two cards are very similar to *DEFINE_DE_INJECTION. In contrast to *DEFINE_DE_INJECTION, this keyword contains additional cards to inject the bonded patterns. The bond properties of Card 3 are the same as the properties of Card 2 in *DEFINE_DE_BOND. In conjunction with the keyword *DEFINE_DE_INJECT_SHAPE, this keyword randomly selects different shape patterns and injects them at the mass flow rate (RMASS) defined in Card 2.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	SID	XC	YC	ZC	XL	YL	CID
Type	I	I	F	F	F	F	F	I
Default	none	none	0.0	0.0	0.0	0.0	0.0	0

Card 2	1	2	3	4	5	6	7	8
Variable	RMASS	VX	VY	VZ	TBEG	TEND		
Type	F	F	F	F	F	F		
Default	none	0.0	0.0	0.0	0.0	10^{20}		

DEFINE_DE_INJECT_BONDED**DEFINE**

Card 3	1	2	3	4	5	6	7	8
Variable	PBN	PBS	PBN_S	PBS_S	SFA	ALPHA	MAXGAP	
Type	F	F	F	F	F	F	F	
Default	none	none	0.0	0.0	1.0	0.0	10^{-4}	

Card 4	1							
Variable	NSHAPE							
Type	I							
Default	0							

Bonded Shape Patterns. Additional cards for NSHAPE $\neq 0$. Define NSHAPE patterns, requiring ceil(NSHAPE/8) cards.

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

VARIABLE	DESCRIPTION
PID	Part ID of new generated DES nodes
SID	Node set ID. Nodes and DES properties are generated automatically during input phase based on the user input and assigned to this SID.
XC, YC, ZC	x, y, z coordinates of the center of injection plane

DEFINE**DEFINE_DE_INJECT_BONDED**

VARIABLE	DESCRIPTION
XL	For rectangular planes XL specifies the planar length along the x -axis in the coordinate system specified by CID. For elliptical planes XL specifies the length of the major axis.
YL	For rectangular planes YL specifies the planar length along the y -axis in the coordinate system specified by CID. For elliptical planes YL specifies the length of the minor axis.
CID	Optional local coordinate system ID, see *DEFINE_COORDINATE_SYSTEM
RMASS	Mass flow rate GE.0.0: Constant mass flow rate LT.0.0: RMASS is a curve ID defining the mass flow rate as a function of time.
VX, VY, VZ	Vector components defining the initial velocity of injected DES specified relative the coordinate system defined by CID.
TBEG	Birth time; time for injection to begin
TEND	Death time; time for injection to end
PBN	Parallel-bond modulus [Pa].
PBS	Parallel-bond stiffness ratio. Shear stiffness/normal stiffness.
PBN_S	Parallel-bond maximum normal stress. A zero value defines an infinite maximum normal stress.
PBS_S	Parallel-bond maximum shear stress. A zero value defines an infinite maximum shear stress.
SFA	Bond radius multiplier.
ALPHA	Numerical damping.
MAXGAP	Maximum gap between two bonded spheres GT.0.0: When MAXGAP is positive, the maximum allowed gap is determined on a bond-by-bond basis as a function of the radii of the two involved spheres. The maximum gap is determined by multiplying the minimum of the two radii by the value of MAXGAP.

VARIABLE	DESCRIPTION
	LT.0.0: Absolute value is used as the maximum gap.
NSHAPE	Number of shape patterns.
ISHAPE	The pattern ID defined in *DEFINE_DE_INJECT_SHAPE. Only the first NSHAPE number of IDs will be used.

*DEFINE

*DEFINE_DE_INJECT_SHAPE

*DEFINE_DE_INJECT_SHAPE

Purpose: Define the bonded shape patterns. When used with *DEFINE_DE_INJECT_BONDED, these shape patterns will be injected with equal probability. Each pattern contains several discrete elements, and the pattern can be in any direction.

There are two options to specify a desired bonded injection pattern:

1. Give the position and radius of each DE in the pattern (IAUTO = 0). The relative positions and radii of the DEs should be carefully defined to generate bonds between the neighboring DEs; see the field MAXGAP in *DEFINE_DE_INJECT_ED_BONDED.
2. Select the shape from predefined pattern types (IAUTO = 1). This method requires the length of the desired shape in all directions and a particle radius. All the particles will have the same radius. The patterns are restricted to lines, prisms with equilateral triangle faces, and cuboids.

Card Summary:

Card Sets: Include as many sets of Cards 1 and 2a/b as desired. Each set defines one shape pattern. This input terminates at the next keyword ("*") card.

Card 1. This card is required.

ID	NDE	IAUTO	ITYPE				
----	-----	-------	-------	--	--	--	--

Card 2a. If IAUTO = 0, include NDE of this card, one for each DE in the pattern.

X	Y	Z	R				
---	---	---	---	--	--	--	--

Card 2b. Include this card if IAUTO = 1.

LX	LY	LZ	R				
----	----	----	---	--	--	--	--

Data Card Definitions:

Card 1	1	2	3	4				
Variable	ID	NDE	IAUTO	ITYPE				
Type	I	I	I	I				
Default	none	none	0	0				

VARIABLE	DESCRIPTION
ID	ID of the shape pattern
NDE	Number of DEs in this pattern
IAUTO	Flag for how to specify the bonded shape patterns: EQ.0: Give each particle's relative position and radius EQ.1: Use predefined pattern types
ITYPE	Bond particles patterns when IAUTO = 1: EQ.1: Line EQ.2: Cuboid EQ.3: Prism with equilateral triangle faces

Card 2 for IAUTO = 0. Include one card for each DE in the pattern (NDE total cards for each Card 1).

Card 2a	1	2	3	4				
Variable	X	Y	Z	R				
Type	F	F	F	F				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
X, Y, Z	Coordinates of the DEs in this pattern
R	Radii of the DEs in this pattern

DEFINE**DEFINE_DE_INJECT_SHAPE**

Card 2 for IAUTO = 1. Include this card to define the length of the bonded pattern as well as the radius for each DE in the pattern.

Card 2b	1	2	3	4				
Variable	LX	LY	LZ	R				
Type	F	F	F	F				
Default	none	none	none	none				

VARIABLE**DESCRIPTION**

LX, LY, LZ

Length of bonded pattern in X, Y, and Z-directions

R

Radius of each DE in this pattern. All DEs in the pattern have the same radius.

DEFINE_DE_INJECTION**DEFINE*****DEFINE_DE_INJECTION_{OPTION}**

Available options include:

<BLANK>

ELLIPSE

Purpose: This keyword injects discrete element spheres (DES) from specified a region at a flow rate given by a user defined curve. When the option is blank, the region from which the DES emanate is assumed rectangular. The ELLIPSE keyword option indicates that the region is to be elliptical.

Card Summary:

Card 1. This card is required.

PID	SID	XC	YC	ZC	XL	YL	CID
-----	-----	----	----	----	----	----	-----

Card 2. This card is required.

RMASS	RMIN	RMAX	VX	VY	VZ	TBEG	TEND
-------	------	------	----	----	----	------	------

Card 3. This card is optional.

IFUNC	NID	IMULTI	LCVX	LCVY	LCVZ	IRAND	
-------	-----	--------	------	------	------	-------	--

Card 3.1. Include this card when IMULTI > 1.

R1	P1	R2	P2	R3	P3	R4	P4
----	----	----	----	----	----	----	----

Card 1	1	2	3	4	5	6	7	8
Variable	PID	SID	XC	YC	ZC	XL	YL	CID
Type	I	I	F	F	F	F	F	I
Default	none	none	0.0	0.0	0.0	0.0	0.0	0

VARIABLE

DESCRIPTION

PID Part ID of generated DES nodes

DEFINE**DEFINE_DE_INJECTION**

VARIABLE	DESCRIPTION
SID	Node set ID. Nodes and DES properties are generated automatically during input phase based on the user input and assigned to this SID.
XC, YC, ZC	x, y, z coordinate of the center of injection plane
XL	For rectangular planes XL specifies the planar length along the x -axis in the coordinate system specified by CID. For elliptical planes XL specifies the length of the major axis.
YL	For rectangular planes YL specifies the planar length along the y -axis in the coordinate system specified by CID. For elliptical planes YL specifies the length of the minor axis.
CID	Optional local coordinate system ID; see *DEFINE_COORDINATE_SYSTEM or *DEFINE_COORDINATE_NOTES

Card 2	1	2	3	4	5	6	7	8
Variable	RMASS	RMIN	RMAX	VX	VY	VZ	TBEG	TEND
Type	F	F	F	F	F	F	F	F
Default	none	none	RMIN	0.0	0.0	0.0	0.0	10^{20}

VARIABLE	DESCRIPTION
RMASS	Mass flow rate (see Remark 2) LT.0: Curve ID
RMIN	Minimum DES radius (ignored if IMULTI > 1)
RMAX	Maximum DES radius (ignored if IMULTI > 1)
VX, VY, VZ	Vector components defining the initial velocity of injected DES specified relative the coordinate system defined by CID.
TBEG	Birth time
TEND	Death time

Optional card 3.

Card 3	1	2	3	4	5	6	7	8
Variable	IFUNC	NID	IMULTI	LCVX	LCVY	LCVZ	IRAND	
Type	I	I	I	I	I	I	I	
Default	0	0	1	0	0	0	0	

VARIABLE	DESCRIPTION
IFUNC	Distribution of particle radii (ignored if IMULTI > 1): EQ.0: Uniform distribution (default) EQ.1: Gaussian distribution (see Remark 1)
NID	An optional node ID. If defined, the center of injection plane follows the motion of this node.
IMULTI	Flag for giving a specified mass distribution of injected particles with given radii: EQ.1: Inject the particles with distribution IFUNC using the radii specified with RMIN and RMAX (default). GT.1: Inject particles with IMULTI different radii, R_i , with each different size having a specified mass distribution, P_i , given in Card 3.1. IMULTI cannot be greater than 4.
LCVX	Load curve defining initial injection velocity in the x -direction
LCVY	Load curve defining initial injection velocity in the y -direction
LCVZ	Load curve defining initial injection velocity in the z -direction
IRAND	Enable a more randomized injection pattern with IRAND = 1.

DEFINE**DEFINE_DE_INJECTION**

Included when IMULTI > 1.

Card 3.1	1	2	3	4	5	6	7	8
Variable	R1	P1	R2	P2	R3	P3	R4	P4
Type	F	F	F	F	F	F	F	F
Default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

VARIABLE	DESCRIPTION
R_i	Injected particle radius. IMULTI radii may be specified.
P_i	Mass percentage of injected particle with radius R_i

Remarks:

1. **Gaussian Distribution of Particle Radii.** The distribution of particle radii for IFUNC = 1 follows a Gaussian distribution:

$$(r|r_0, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(r-r_0)^2}{2\sigma^2}},$$

where the mean radius is given by

$$r_0 = \frac{1}{2} (r_{\max} + r_{\min})$$

and the standard deviation is

$$\sigma = \frac{1}{2} (r_{\max} - r_{\min}).$$

2. **Maximum Estimated Mass Flow Rate.** The maximum estimated mass flowrate injected through plane (XL,YL) is:

$$Q_{\max} = 0.9 \times \pi \times \rho \times XL \times YL \times \frac{1}{6} V_{\max}$$

where

$$V_{\max} = \sqrt{VX^2 + VY^2 + VZ^2}$$

If RMASS > Q_{\max} , LS-DYNA issues the warning message: "input mass flowrate may exceed the maximum estimated rate."

3. **Restrictions for Rebalancing.** Dynamic rebalancing does not work with injections. Thus, NCRB in *CONTROL_DISCRETE_ELEMENT should be set to zero

when using this feature. Instead, you can use *CONTROL_MPP_DECOMPOSITION_REDECOMPOSITION for rebalancing purposes.

***DEFINE**

***DEFINE_DE_INTERNAL_SKIP**

***DEFINE_DE_INTERNAL_SKIP**

Purpose: DES defined in PID will ignore internal particle to particle interaction force.

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

VARIABLE

DESCRIPTION

PID Part set ID or part ID. TYPE below indicates the ID type.

TYPE PID type:

EQ.0: Part set

EQ.1: Part

DEFINE_DE_MASSFLOW_PLANE**DEFINE*****DEFINE_DE_MASSFLOW_PLANE**

Purpose: Measure DES mass flow rate across a defined plane. See also the accompanying keyword *DATABASE_DEMASSFLOW which controls the output frequency.

Card 1	1	2	3	4	5	6	7	8
Variable	PRTCLSID	SURFSID	PTYPE	STYPE				
Type	I	I	I	I				
Default	0	0	0	0				

VARIABLE	DESCRIPTION
PRTCLSID	Node set ID, node ID, part set ID or part ID specifying DES to be measured. PTYPE below indicates the ID type specified by PRTCLSID.
SURFSID	Part set ID or part ID defining the surface. STYPE below indicates the ID type specified by SURFSID.
PTYPE	PRTCLSID type: EQ.0: Node set EQ.1: Node EQ.2: Part set EQ.3: Part
STYPE	SURFSID type: EQ.0: Part set EQ.1: Part

*DEFINE

*DEFINE_DE_MESH_BEAM

*DEFINE_DE_MESH_BEAM

Purpose: Generate and place discrete element sphere (DES) elements along the axis of beam elements. By default, the generated DES is treated as a shadow (inactive) element on the beam element. It only transfers forces between the ICFD element and beam element to make ICFD coupling easier.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	TYPE	NQUAD	DESPID	DESXID	NSID	RSF	IACTIVE
Type	I	I	I	I	I	I	F	I
Default	none	none	1	Rem 1	Rem 1	0	1.0	0

Card 2	1	2	3	4	5	6	7	8
Variable	MASS	INERTIA	RADIUS					
Type	F	F	F					
Default	none	none	none					

VARIABLE	DESCRIPTION
SID	Part or part set ID for the region of the mesh upon which the DES elements will be placed
TYPE	SID type: EQ.0: Part set ID EQ.1: Part ID
NQUAD	Number of equally spaced DES elements created along the axis of beam element (maximum NQUAD = 4)
DESPID	Part ID for generated DES elements
DESXID	Section ID for generated DES elements

VARIABLE	DESCRIPTION
NSID	Create a node set with ID NSID (see *SET_NODE) for the nodes generated by this keyword. By default, no node set is created.
RSF	Scale factor of DES radius
IACTIVE	Activate DES: EQ.0: DES is inactive and used as a shadow (default). EQ.1: DES is active.
MASS	DES Mass: GT.0: DES mass EQ.-1: The DES particle radius (r) is $0.5 \times$ Beam Length / NQUAD, the DES mass (m) is $4\pi\rho r^3/3$, and the moment of inertia is $2mr^2/5$. Input fields INERTIA and RADIUS are ignored.
INERTIA	Mass moment of inertia (ignored if MASS = -1)
RADIUS	Particle radius (ignored if MASS = -1)

Remarks:

1. **DESPID and DESXID.** The part ID and/or section ID will be generated by this card if they are not provided in the input.

DEFINE**DEFINE_DE_MESH_SURFACE*****DEFINE_DE_MESH_SURFACE**

Purpose: Generate and place discrete element sphere (DES) elements on the surface of shell elements. By default, the generated DES is treated as a shadow (inactive) element on the surface. It only transfers forces between the ICFD element and shell element to make ICFD coupling easier.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	TYPE	NQUAD	DESPID	DESXID	NSID	RSF	IACTIVE
Type	I	I	I	I	I	I	F	I
Default	none	none	1	Rem 1	Rem 1	0	1.0	0

VARIABLE	DESCRIPTION
SID	Part or part set ID for the region of the mesh upon which the DES elements will be placed
TYPE	SID type: EQ.0: Part set ID EQ.1: Part ID
NQUAD	Number of equally spaced DES elements created on a shell element in each local shell direction. For instance, NQUAD × NQUAD DES elements will be created on the surface a quad shell. (Maximum NQUAD = 4)
DESPID	Part ID for generated DES elements
DESXID	Section ID for generated DES elements
NSID	If defined, this card creates a node set with ID NSID (see *SET_NODE) for the nodes generated by this card.
RSF	Scale factor for determining the DES radius: GE.0.0: Scale factor on the internally determined radius based on shell thickness. LT.0.0: DES radius is $0.5 \times \text{RSF} \times (\text{maximum diagonal length})/\text{NQUAD}$

VARIABLE	DESCRIPTION
	for rectangular segments and $0.5 \times \text{RSF} \times (\text{maximum side length})/\text{NQUAD}$ for triangular segments.
IACTIVE	Activate DES: EQ.0: DES is inactive and used as a shadow (default). EQ.1: DES is active.

Remarks:

1. **DESPID and DESXID.** Part ID and/or section ID will be generated by this card if they are not provided in the input.

*DEFINE

*DEFINE_DE_MESH_VOLUME

*DEFINE_DE_MESH_VOLUME

Purpose: Generate and place discrete element sphere (DES) elements inside the volume of solid elements. By default, the generated DES is treated as a shadow (inactive) element in the volume. It only transfers forces between the ICFD element and solid element to make ICFD coupling easier.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	TYPE	NQUAD	DESPID	DESXID	NSID	RSF	IACTIVE
Type	I	I	I	I	I	I	F	I
Default	none	none	1	Rem 1	Rem 1	0	1.0	0

VARIABLE	DESCRIPTION
SID	Part or part set ID for the region of the mesh upon which the DES elements will be placed
TYPE	SID type: EQ.0: Part set ID EQ.1: Part ID
NQUAD	Number of equally spaced DES elements created on a shell element in each local shell direction. For instance, NQUAD × NQUAD DES elements will be created on the surface a quad shell. (Maximum NQUAD = 4)
DESPID	Part ID for generated DES elements
DESXID	Section ID for generated DES elements
NSID	If defined, this card creates a node set with ID NSID (see *SET_NODE) for the nodes generated by this card.
RSF	Scale factor for determining the DES radius: GE.0.0: Scale factor on the internally determined radius based on shell thickness
IACTIVE	Activate DES: EQ.0: DES is inactive and used as a shadow (default).

VARIABLE	DESCRIPTION
	EQ.1: DES is active.

Remarks:

1. **DESPID and DESXID.** The part ID and/or section ID for the generated DES particles will be created by this keyword if they are not provided in the input.

DEFINE**DEFINE_DE_PARTS_INTERACTION*****DEFINE_DE_PARTS_INTERACTION**

Purpose: Provide contact parameters for the interaction between discrete element spheres that are in different parts or in the same part. This card overrides the values set in [*CONTROL_DISCRETE_ELEMENT](#) and [*DEFINE_DE_BY_PART](#).

Card 1	1	2	3	4	5	6	7	8
Variable	PID1	PID2	NDAMP	TDAMP	FRICS	FRICR	NORMK	SHEARK
Type	I	I	F	F	F	F	F	F
Default	none	none	0.	0.	0.	0.	0.01	2/7

VARIABLE	DESCRIPTION
PID1	Part ID of DES nodes (see Remark 1)
PID2	Part ID of DES nodes
NDAMP	Normal damping coefficient
TDAMP	Tangential damping coefficient
FRICS	Static coefficient of friction (see Remark 2): EQ.0.0: 3 DOF NE.0.0: 6 DOF (consider rotational DOF)
FRICR	Rolling friction coefficient
NORMK	Optional scale factor of normal spring constant (Default = 0.01)
SHEARK	Optional ratio between SHEARK/NORMK (Default = 2/7)

Remarks:

- Interaction within a single part.** PID1 and PID2 can be the same part for providing contact parameters to specify interaction within a single part.
- Friction.** The friction coefficient is assumed to depend on the relative velocity v_{rel} of the two DEM in contact:

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

DEFINE_DE_PATTERN_OUTPUT**DEFINE*****DEFINE_DE_PATTERN_OUTPUT**

Purpose: Output when DES cross specified planes and what their coordinates are when they cross the planes. This feature is for visualizing the pattern of shotgun pellets at various locations during post-processing. The data is output to dem_pattern. With this keyword, you specify the line to which the planes are orthogonal and where the plane is along the line from an origin point.

Card Summary:

Card 1. This card is required.

PID	PTYPE	X0	Y0	Z0	XH	YH	ZH
-----	-------	----	----	----	----	----	----

Card 2. This card is required.

NSET							
------	--	--	--	--	--	--	--

Card 2.1. Include NSET/8 of this card.

DIST1	DIST2	DIST3	DIST4	DIST5	DIST6	DIST7	DIST8
-------	-------	-------	-------	-------	-------	-------	-------

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	PID	PTYPE	X0	Y0	Z0	XH	YH	ZH
Type	I	I	F	F	F	F	F	F
Default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

VARIABLE	DESCRIPTION
PID	Part ID/part set ID of the DES
PTYPE	Type for PID: EQ.0: Part EQ.1: Part set
XO, YO, ZO	Coordinates of the origin from which the distance to the planes is measured along the vector (XH,YH,ZH) – (XO,YO,ZO)

DEFINE**DEFINE_DE_PATTERN_OUTPUT**

VARIABLE		DESCRIPTION						
XH, YH, ZH		Head of the direction to which the planes are orthogonal						
Card 2	1	2	3	4	5	6	7	8
Variable	NSET							
Type	I							
Default	0							

VARIABLE		DESCRIPTION						
NSET		Number of planes for which patterns will be recorded						
Include NSET/8 of this card to specify the distance of each plane from the origin								
Card 2.1	1	2	3	4	5	6	7	8
Variable	DIST1	DIST2	DIST3	DIST4	DIST5	DIST6	DIST7	DIST8
Type	F	F	F	F	F	F	F	F
Default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

VARIABLE		DESCRIPTION						
DIST1, . . .		Distance of each plane from (XO, YO,ZO)						

DEFINE_DE_TEMP**DEFINE*****DEFINE_DE_TEMP**

Purpose: Change the radius of discrete element sphere (DES) particles due to a change in temperature. Note that *DEFINE_DE_TEMP does not transfer heat from particle to particle.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	STYPE	LCID	SFT	ALPHA	INI_TEMP		
Type	I	I	I	F	F	F		
Default	none	0	0	0.0	0.0	0.0		

VARIABLE	DESCRIPTION
SID	Node set, part set, or part ID for which the temperature effect applies
STYPE	SID type: EQ.0: DES node set EQ.2: DES part set EQ.3: DES part
LCID	Load curve ID for the temperature as a function of time during the dynamic process
SFT	Scale factor on the temperature given with LCID
ALPHA	Coefficient of thermal expansion ($1/\text{ }^{\circ}\text{C}$ or $1/\text{ }^{\circ}\text{F}$), α
INI_TEMP	Initial temperature of the DES particles ($^{\circ}\text{C}$ or $^{\circ}\text{F}$)

Remarks:

1. **Change in radius.** This feature changes the radius of a particle due to a change in temperature. The change in radius is given by:

$$\Delta r = r \times \alpha \times \Delta T$$

Here, Δr is the change in radius of the particle, α is the coefficient of thermal expansion, r is the radius of the particle, and ΔT is the change in temperature.

DEFINE**DEFINE_DE_TO_BEAM_COUPLING*****DEFINE_DE_TO_BEAM_COUPLING**

Purpose: To define a coupling interface between discrete element spheres (DES) and beams.

Card 1	1	2	3	4	5	6	7	8
Variable	DESID	BEAMID	DESTYP	BEAMTYP				
Type	I	I	I	I				
Default	0	0	0	0				

Card 2	1	2	3	4	5	6	7	8
Variable	FRICS	FRICD	DAMP	BSORT				
Type	F	F	F	I				
Default	0	0	0	100				

VARIABLE	DESCRIPTION
DESID	Node set ID, node ID, part set ID or part ID specifying the DES in the coupling. DESTYP below indicates the ID type.
BEAMID	Part set ID or part ID specifying the beams in the coupling. BEAMTYP below indicates the ID type.
DESTYP	Type for DESID: EQ.0: Node set EQ.1: Node EQ.2: Part set EQ.3: Part
BEAMTYP	Type for BEAMID: EQ.0: Part set EQ.1: Part

VARIABLE	DESCRIPTION
FRICS	Friction coefficient
FRICD	Rolling friction coefficient
DAMP	Damping coefficient
BSORT	Number of cycles between bucket sorts

*DEFINE

*DEFINE_DE_TO_CPM_COUPLING

*DEFINE_DE_TO_CPM_COUPLING

Purpose: Define a coupling interface between discrete element spheres (DES) and a CPM airbag (*AIRBAG_PARTICLE_ID).

Card 1	1	2	3	4	5	6	7	8
Variable	DESID	BAGID	DESTYP					
Type	I	I	I					
Default	0	0	0					

VARIABLE	DESCRIPTION
DESID	Part set ID or part ID specifying the DES in the coupling. DESTYP below indicates the ID type.
BAGID	*AIRBAG_PARTICLE ID specifying the CPM airbag in the coupling
DESTYP	Type for DESID: EQ.0: Part set EQ.1: Part

***DEFINE_DE_TO_SURFACE_COUPLING_{OPTION}**

Purpose: To define a non-tied coupling interface between discrete element spheres (DES) and a surface defined by shell part(s) or solid part(s). This coupling is currently not implemented for tshell part(s).

Available options include:

<BLANK>

TRANSDUCER

The TRANSDUCER keyword option creates a force transducer for measuring the coupling interface forces. With this option, LS-DYNA outputs the translational forces from the active surface coupling interfaces to dem_rcforc. The *only* fields read when using this option are SURFID, SURFTYP, and CID_RCF. See [Remark 5](#).

Card 1	1	2	3	4	5	6	7	8
Variable	DESID	SURFID	DESTYP	SURFTYP	ISOFT			SBOX
Type	I	I	I	I	I			I
Default	none	none	0	0	0			0

Card 2	1	2	3	4	5	6	7	8
Variable	FRICS	FRICD	DAMP	BSORT	LCVX	LCVY	LCVZ	WEARC
Type	F	F	F	I	I	I	I	F
Default	0.0	0.0	0.0	100	0	0	0	0.

DEFINE**DEFINE_DE_TO_SURFACE_COUPLING**

User-Defined Wear Parameter Cards. Additional card for WEARC < 0.

Card 3	1	2	3	4	5	6	7	8
Variable	W1	W2	W3	W4	W5	W6	W7	W8
Type	F	F	F	F	F	F	F	F
Default	none							

This card is optional.

Card 4	1	2	3	4	5	6	7	8
Variable	SFP	SFT			IDEACT	CID_RCF	BT	DT
Type	F	F			I	I	F	F
Default	1.0	1.0			0	0	0.0	10^{20}

VARIABLE	DESCRIPTION
DESID	Node set ID, node ID, part set ID, or part ID specifying DES in the coupling. DESTYP below indicates the ID type.
SURFID	Part set ID or part ID specifying the surface. SURFTYP below indicates the ID type.
DESTYP	DESID type: EQ.0: Node set EQ.1: Node EQ.2: Part set EQ.3: Part
SURFTYP	SURFID type: EQ.0: Part set EQ.1: Part EQ.2: Segment set (TRANSDUCER keyword option only; see Remark 5)

VARIABLE	DESCRIPTION
ISOFT	Contact stiffness evaluation: EQ.0: Default is based on DES properties only. $K_a = K_{\text{des}}$ EQ.1: Geometric mean is based on both DES and contact segment's properties. $K_b = \sqrt{K_{\text{des}} \times \text{SFP} \times \text{bulk modulus of the surface}}$ EQ.2: Soft option is based on mass of the system and time step size. $K_c = \max(K_b, \frac{\text{SFP} \times \max(M_{\text{des}}, M_{n_1}, M_{n_2}, M_{n_3}, M_{n_4})}{dt^2})$
SBOX	BOX ID. Exclude segments belonging to SURFID that are outside of the box during initialization.
FRICS	Friction coefficient
FRICD	Rolling friction coefficient
DAMP	Damping coefficient (unitless). If a discrete element sphere impacts a rigid surface with a velocity v_{impact} , the rebound velocity is $v_{\text{rebound}} = (1 - \text{DAMP})v_{\text{impact}}$
BSORT	Number of cycles between bucket sorts; the default value is 100. For blast simulations with very high DEM particle velocity, we suggest setting BSORT = 20 or smaller. LT.0: $ \text{BSORT} $ is the minimum number of cycles between bucket sorts. This value can be increased during runtime by tracking the velocity of potential coupling pairs. This feature only works with MPP currently.
LVCX	Load curve defining surface velocity in the x -direction
LVCY	Load curve defining surface velocity in the y -direction
LVCZ	Load curve defining surface velocity in the z -direction
WEARC	WEARC is the wear coefficient: GT.0: Archard's Wear Law; see Remark 1 .

DEFINE**DEFINE_DE_TO_SURFACE_COUPLING**

VARIABLE	DESCRIPTION
	EQ.-1: Finnie Wear Law; an additional card is required
	LE.-100: User-defined wear model; an additional card is required, see Remark 6 .
W1-W8	For WEARC = -1, W1 is the yield stress of the target material. For WEARC \leq -100, Wi are user-defined wear parameters.
SFP	Scale factor on contact stiffness. By default, SFP = 1.0. The contact stiffness is calculated as $K_n = \text{SFP} \times \begin{cases} K \times r \times \text{NormK} & \text{if NormK} > 0 \\ \text{NormK} & \text{if NormK} < 0 \end{cases}$ where K is the bulk modulus, r is the discrete element radius, and NormK is the scale factor of the spring constant defined in *CONTROL_DISCRETE_ELEMENT.
SFT	Scale factor for surface thickness (scales true thickness). True thickness is the element thickness of the shell elements. This option applies only to contact with shell elements.
IDEACT	DES particles will be automatically deactivated after contacting the surface when IDEACT = 1.
CID_RCF	Coordinate system ID. Force resultants are output to the demforc file in a local system.
BT	Birth time
DT	Death time

Remarks:

1. **Archard's wear law.** If WEARC > 0, then the wear on the shell surface is calculated using Archard's wear law

$$h = \frac{\text{WEARC} \times f_n \times v_t}{A}$$

where,

h = wear depth

f_n = normal contact force from DE

v_t = tangential sliding velocity of the DE on shell

A = area of contact segment

The wear depth is output to the interface force file.

2. **Finnie's wear law.** If WEARC = -1, Finnie's wear law determines the wear on the shell surface. The model of Finnie relates the rate of wear to the rate of kinetic energy of particle impact on a surface as:

$$Q = \begin{cases} \frac{mv^2}{8p} (\sin 2\alpha - 3 \sin^2 \alpha) & \text{if } \tan \alpha < \frac{1}{3} \\ \frac{mv^2}{24p} \cos^2 \alpha & \text{if } \tan \alpha > \frac{1}{3} \end{cases}$$

where Q is the volume of the material removed from the surface, m is particle mass, α is the impact angle, and p is the yield stress of the target material input in field W1. The wear depth is output to the interface force file.

3. **Interface force file.** *DATABASE_BINARY_DEMFOR controls the output interval of the coupling forces to the DEM interface force file. This interface force file is activated by the command line option “dem=”, for example,

```
lsdyna i=inputfilename.k ... dem=interfaceforce_filename
```

The DEM interface force file can be read into LS-PrePost to plot coupling pressure and forces on the surface segments.

4. **Coupling force output.** *DATABASE_RCFORC controls the output interval of the coupling forces to the ASCII demrcf file. This output file is analogous to the rcfrcf file for *CONTACT.
5. **TRANSDUCER.** The TRANSDUCER keyword option causes LS-DYNA to collect and record the translational forces from a section of the surface side of an active surface coupling interface. The collected forces are from surface nodes instead of surface segments. Therefore, total force may be a little different from the parent interface. With this option, SURFID, SURFTYP, and CID_RCF are the only input not ignored.
6. **User-defined wear model.** When WEARC ≤ -100 , the subroutine userdewear in the file dyn21cnt.f in the usermat library defines the wear model; see comments within the subroutine for instructions. LS-DYNA passes parameters W1-W8 to the subroutine. Setting $-108 \leq \text{WEARC} < -100$ activates passing information between time steps with history variables. A maximum of 8 history variables can be defined. The $\min(|\text{WEARC}| - 100, 8)$ gives the number of available history variables. The interface force file contains these history variables for post-processing.

DEFINE**DEFINE_DE_TO_SURFACE_TIED*****DEFINE_DE_TO_SURFACE_TIED**

Purpose: To define a tied-with-failure coupling interface between discrete element spheres (DES) and a surface defined by shell part(s) or solid part(s). This coupling is currently not implemented for tshell part(s).

Card 1	1	2	3	4	5	6	7	8
Variable	DESID	SURFID	DESTYP	SURFTYP				
Type	I	I	I	I				
Default	0	0	0	0				

Card 2	1	2	3	4	5	6	7	8
Variable	NFLF	SFLF	NEN	MES	LCID	NSORT	MAXGAP	
Type	F	F	F	F	I	I	F	
Default	none	none	2.	2.	0	100	0.0	

VARIABLE	DESCRIPTION
DESID	Node set ID, node ID, part set ID or part ID specifying DES in the tied interface. DESTYP below indicates the ID type.
SURFID	Part set ID or part ID specifying the surface. SURFTYP below indicates the ID type.
DESTYP	DESID type: EQ.0: Node set EQ.1: Node EQ.2: Part set EQ.3: Part
SURFTYP	SURFID type: EQ.0: Part set

VARIABLE	DESCRIPTION
	EQ.1: Part
NFLF	Normal failure force. Only tensile failure, that is, tensile normal forces, will be considered in the failure criterion
SFLF	Shear failure force
NEN	Exponent for normal force
MES	Exponent for shear force. Failure criterion: $\left(\frac{ f_n }{NFLF}\right)^{NEN} + \left(\frac{ f_s }{SFLF}\right)^{MES} \geq 1.$ Failure is assumed if the left side is larger than 1. f_n and f_s are the normal and shear interface force.
LCID	Optional curve ID defining a scale factor as a function of time. The scale factor is applied to NFLF and SFLF, making the failure forces time-dependent.
NSORT	Number of cycles between bucket sorts
MAXGAP	Determines maximum gap between DES and the surface: GT.0.0: The maximum gap is determined by scaling the DES radius by MAXGAP, that is, $\text{MAXGAP} \times r_{DES}$ LT.0.0: $ \text{MAXGAP} $ is used as the maximum gap.

Remarks:

Both NFLF and SFLF must be defined. If failure in only tension or shear is required, then set the other failure force to a large value (10^{10}).

*DEFINE

*DEFINE_DEATH_TIMES

*DEFINE_DEATH_TIMES_OPTION

Available options include:

NODES

SET

RIGID

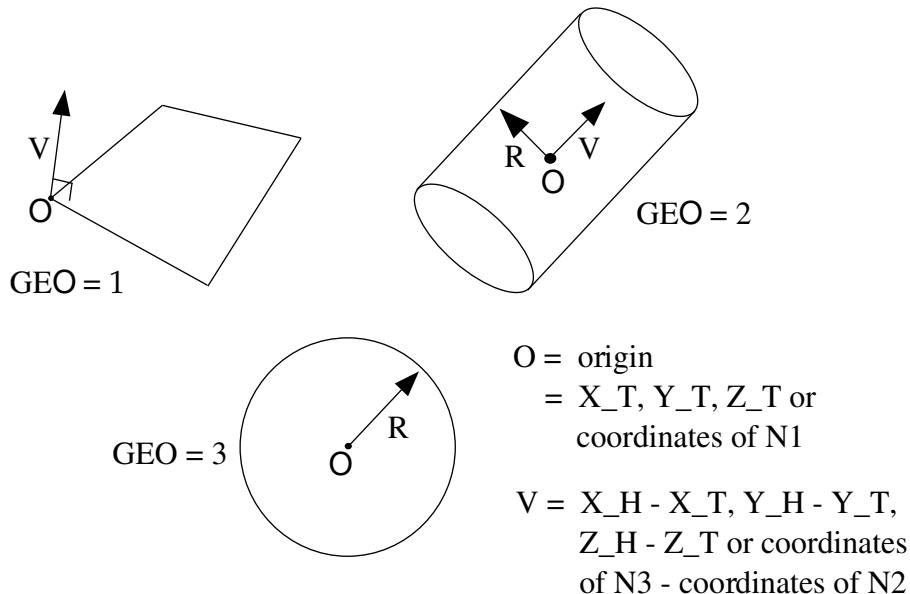
Purpose: To dynamically define the death times for *BOUNDARY_PRESCRIBED_MOTION based on the locations of nodes and rigid bodies. Once a node or rigid body moves past a plane or a geometric entity, the death time is set to the current time. The input in this section continues until the next keyword ("*") card is detected.

Card 1	1	2	3	4	5	6	7	8
Variable	GEO	N1	N2	N3				
Type	I	I	I	I				
Default	none	0	0	0				

Card 2	1	2	3	4	5	6	7	8
Variable	X_T	Y_T	Z_T	X_H	Y_H	Z_H	R	FLAG
Type	F	F	F	F	F	F	F	I
Default	Rem 1	Rem 1	Rem 1	Rem 2	Rem 2	Rem 2	none	1

ID Cards. Set the list of nodes and rigid bodies affected by this keyword. This input terminates at the next keyword ("*") card.

Card 3	1	2	3	4	5	6	7	8
Variable	NSID1	NSID2	NSID3	NSID4	NSID5	NSID6	NSID7	NSID8
Type	I	I	I	I	I	I	I	I

**Figure 17-39.** Geometry types.

VARIABLE	DESCRIPTION
GEO	Geometric entity type: EQ.1: Plane EQ.2: Infinite cylinder EQ.3: Sphere
N1	Node defining the origin of the geometric entity. See Remark 1 .
N2	Node defining the tail of the orientation vector. See Remark 2 .
N3	Node defining the head of the orientation vector. See Remark 2 .
X_T	x-coordinate of the origin of the geometric entity and the tail of the orientation vector. See Remark 1 .
Y_T	y-coordinate of the origin of the geometric entity and the tail of the orientation vector. See Remark 1 .
Z_T	z-coordinate of the origin of the geometric entity and the tail of the orientation vector. See Remark 1 .
X_H	x-coordinate of the head of the orientation vector. See Remark 2 .
Y_H	y-coordinate of the head of the orientation vector. See Remark 2 .

DEFINE**DEFINE_DEATH_TIMES**

VARIABLE	DESCRIPTION
Z_H	z-coordinate of the head of the orientation vector. See Remark 2 .
R	Radius of cylinder or sphere.
FLAG	Flag for where the node or rigid body must be with respect to the geometric entity for the death time to be set to the current time: EQ.1: The node or rigid body is outside of the geometric entity or on the positive side of the plane as defined by the normal direction. EQ.-1: Node or rigid body is inside the geometric entity or on the negative side of the plane.
NSID i	i^{th} node, node set, or rigid body

Remarks:

1. **Origin.** Either N1 or X_T, Y_T, and Z_T should be specified, but not both.
2. **Orientation Vector.** Either N2 and N3 or X_H, Y_H, and Z_H should be specified, but not both. Specifying N2 and N3 is equivalent to setting the head of the vector equal to the tail of the vector (X_T,Y_T,Z_T) plus the vector from N2 to N3.

DEFINE_DRIFT_REMOVE**DEFINE*****DEFINE_DRIFT_REMOVE**

Purpose: Minimize the drift of a structure over a loop by modifying the curves for specifying accelerations and forces.

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

Curve Card. Include as many of this card as needed. Input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	ID1	ID2	ID3	ID4	ID5	ID6	ID7	ID8
Type	I	I	I	I	I	I	I	I

VARIABLE	DESCRIPTION
OPTION	Specifies how curves are handled. EQ.0: no changes are made to the curves. EQ.1: modify to minimize the drift.
ID[N]	ID of N th curve to be modified

Remarks:

When data from an accelerometer is integrated to obtain velocities and displacements numerical noise leads to drift. For signals that are necessarily periodic, like data collected as a car drives around a track, the resulting quadrature loses its periodicity. This keyword activates an algorithm that approximately restores periodicity.

Because its chief property is *not* being periodic we approximate noise, a_{noise} , by a function that is not periodic, namely, a parabola (or possibly a straight line)

$$a_{\text{noise}}(t) = a_0 + a_1 t + a_2 t^2 .$$

DEFINE**DEFINE_DRIFT_REMOVE**

Since a_{noise} cannot be allowed in the context of periodicity we seek to find an $a_{\text{corrected}}$

$$a_{\text{corrected}} = a_{\text{data}} - a_{\text{noise}}$$

that removes as much “noise” as possible. We therefore find the a_i that minimize the norm of the corrected velocity,

$$\int_{t_0}^{t_1} [v_{\text{corrected}}(t)]^2 dt ,$$

where the corrected velocity is the quadrature of the corrected noise,

$$v_{\text{corrected}}(t) = \int_{t_0}^t [a_{\text{data}}(\tau) - a_{\text{noise}}(\tau)] d\tau = v_{\text{data}}(t) - \left(a_0 t + \frac{1}{2} a_1 t^2 + \frac{1}{3} a_2 t^3 \right).$$

For each curve listed on the data cards LS-DYNA approximately removes the drift by numerically solving the above minimization problem.

DEFINE_ELEMENT_DEATH**DEFINE*****DEFINE_ELEMENT_DEATH_OPTION**

Available options include:

SOLID

SOLID_SET

BEAM

BEAM_SET

SHELL

SHELL_SET

THICK_SHELL

THICK_SHELL_SET

Purpose: Set a time or space condition for deleting an element or element set during a simulation. This keyword is only for deformable elements, not rigid body elements.

Card 1	1	2	3	4	5	6	7	8
Variable	EID/SID	TIME	BOXID	INOUT	IDGRP	CID	PERCENT	
Type	I	F	I	I	I	I	F	
Default	none	0.0	{Ø}	0	0	global	0.0	

VARIABLE**DESCRIPTION**

EID/SID

Element ID or element set ID identifying elements that are considered for deletion, either by meeting the BOXID/INOUT criterion or the independent TIME/IDGRP/PERCENT criterion.

TIME

Deletion time for elimination of the element or element set. If BOX-ID is nonzero, a TIME value of zero is reset to 10^{16} .

BOXID

An optional box ID; see CID below and *DEFINE_BOX. An element is immediately deleted upon meeting the condition of being inside the box (or outside the box, depending on INOUT), without regard to TIME, IDGRP, or PERCENT. For an element set, only the element (or elements) in the set meeting the condition is deleted at

*DEFINE

*DEFINE_ELEMENT_DEATH

VARIABLE	DESCRIPTION
	each time step. Before R11, the elements in a single IDGRP were all deleted when one element in the group met the condition despite the value of PERCENT. As of R11, all the elements in the group are deleted when PERCENT of the elements in the group meet the condition.
INOUT	Flag pertaining to BOXID: EQ.0: Elements inside the box EQ.1: Elements outside the box
IDGRP	Group ID. Elements sharing the same positive value of IDGRP are considered to be in the same group. All elements in a group will be simultaneously deleted one cycle after a percentage of the elements (specified in PERCENT) fail. There is no requirement that each *DEFINE_ELEMENT_DEATH command have a unique IDGRP. In other words, elements in a single group can come from multiple *DEFINE_ELEMENT_-DEATH commands.
	Elements in which IDGRP = 0 are not assigned to a group and thus deletion of one element does not cause deletion of the other elements.
CID	Coordinate ID for transforming box BOXID. If CID is not specified, the box is in the global coordinate system. The box rotates and translates with the coordinate system only if the coordinate system is flagged for an update every time step.
PERCENT	Deletion percentage. EQ.0.0: When one element fails, all elements in the group will be deleted (default). GT.0.0: Percentage of elements failed before elements in group IDGRP are deleted.

DEFINE_ELEMENT_EROSION**DEFINE*****DEFINE_ELEMENT_EROSION_OPTION**

Available options include:

IGA

SHELL

TSHELL

Purpose: Defines the number of failed integration points per layer that fails a layer and the number of failed layers that triggers element deletion (erosion) during the simulation. This keyword is only for deformable shells and deformable thick shells. This keyword must be used in conjunction with a material model that has a failure option, or else with *MAT_ADD_EROSION. The criteria specified on this card will override other criteria related to the number of failed integration points for element deletion, such as NUMFIP on *MAT_ADD_EROSION or NUMINT on other *MAT cards.

Either the SHELL option or the TSHELL option is required, except for in the case of isogeometric analysis. Isogeometric analysis requires the IGA option and currently only supports isogeometric shell elements for STYP = 3 or 4.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	STYP	NUMFIP	NIPF				
Type	I	I	F	I				
Default	none	none	-100.	1				

VARIABLE	DESCRIPTION
SID	Element ID, element set ID, part ID, or part set ID; see *PART, *SET_PART or *SET_T/SHELL_OPTION.
STYP	ID type of SID: EQ.1: Shell or tshell element ID EQ.2: Shell or tshell element set ID EQ.3: Part ID EQ.4: Part set ID

DEFINE**DEFINE_ELEMENT_EROSION**

VARIABLE	DESCRIPTION
NUMFIP	Number of layers which must fail prior to element deletion. LT.0.0: $ NUMFIP $ is the percentage of layers which must fail prior to element deletion.
NIFP	Number of integration points within one layer that need to fail to indicate a failed layer

***DEFINE_ELEMENT_GENERALIZED_SHELL**

Purpose: Define a general 3D shell formulation to be used in combination with *ELEMENT_GENERALIZED_SHELL. The objective of this feature is to allow the rapid prototyping of new shell element formulations by adding them through the keyword input file.

All necessary information, like the values of the shape functions and their derivatives at various locations (at the integration points and at the nodal points), must be defined using this keyword. An example for a 9-noded generalized shell element with 4 integration points in the plane is given in [Figure 17-40](#) to illustrate the procedure. The element formulation ID (called ELFORM) used in this keyword needs to be greater or equal than 1000 and will be referenced through *SECTION_SHELL (see [Figure 19-4](#) in *ELEMENT_GENERALIZED_SHELL).

Card 1	1	2	3	4	5	6	7	8
Variable	ELFORM	NIPP	NMNP	IMASS	FORM			
Type	I	I	I	I	I			
Default	none	none	none	none	none			

Weights and Shape Function Values/Derivatives at Gauss Points:

These cards are read according to the following pseudo code:

```

for i = 1 to NIPP {
    read Card 2(i)
    for k = 1 to NMNP {
        read Card 3(i,k)
    }
} // comment: Read in NIPP × (1 + NMNP) cards

```

Weight Cards. Provide weight for integration point *i*.

(Card 2) _i	1	2	3	4	5	6	7	8
Variable	WI							
Type	F							

DEFINE**DEFINE_ELEMENT_GENERALIZED_SHELL**

Integration Point Shape Function Value/Derivatives Cards. Provide the value of the k^{th} shape function and its derivative at the i^{th} integration point.

(Card 3) $_{ik}$	1	2	3	4	5	6	7	8
Variable	NKI		DNKIDR			DNKIDS		
Type	F		F			F		

For FORM = 0 or FORM = 1, Shape Function Derivatives at Nodes:

These cards are read according to the following pseudo code:

```
for I = 1 to NMNP {
    for k = 1 to NMNP {
        read Card 4a(I,k)
    }
} // comment: Read in NMNP × NMNP cards
```

Nodal Shape Function Derivative Cards. The value of the k^{th} shape function's derivative at the I^{th} nodal point.

(Card 4a) $_{Ik}$	1	2	3	4	5	6	7	8
Variable	DNKldr		DNKLds					
Type	F		F					

For FORM = 2 or FORM = 3, Shape Function 2nd derivative at Gauss Points:

NOTE: For FORM = 2 and FORM = 3 it is assumed that the shape functions are at least C1 continuous (having a continuous derivative).

The cards for this method are read according to the following pseudo code:

```

for i = 1 to NIPP {
    for k = 1 to NMNP {
        read Card 4b(i,k)
    }
} // comment: Read in NGP × NMNP cards

```

Nodal Shape Function Second Derivative Cards. The value of the k^{th} shape function's second derivative at the i^{th} integration point.

(Card 4b) $_{ik}$	1	2	3	4	5	6	7	8
Variable	D2NKIDR2		D2NKIDRDS			D2NKIDS2		
Type	F		F			F		

VARIABLE	DESCRIPTION
ELFORM	Element Formulation ID which is referenced by *SECTION_SHELL to connect *ELEMENT_GENERALIZED_SHELL with the appropriate shell formulation. The chosen number needs to be greater or equal than 1000.
NIPP	Number of in-plane integration points.
NMNP	Number of nodes for this element formulation.
IMASS	Option for lumping of mass matrix: EQ.0: row sum EQ.1: diagonal weighting.
FORM	Shell formulation to be used EQ.0: shear deformable shell theory with rotational DOFs (shell normal evaluated at the nodes) EQ.1: shear deformable shell theory without rotational DOFs (shell normal evaluated at the nodes) EQ.2: thin shell theory without rotational DOFs (shell normal evaluated at the integration points) EQ.3: thin shell theory with rotational DOFs (shell normal evaluated at the integration points)
WI	Integration weight at integration point i .

DEFINE**DEFINE_ELEMENT_GENERALIZED_SHELL**

VARIABLE	DESCRIPTION
NKI	Value of the shape function N_k evaluated at integration point i .
DNKIDR	Value of the derivative of the shape function N_k with respect to the local coordinate r at the integration point i $\left(\frac{\partial N_k^i}{\partial r}\right)$.
DNKIDS	Value of the derivative of the shape function N_k with respect to the local coordinate s at the integration point i $\left(\frac{\partial N_k^i}{\partial s}\right)$.
DNKLDR	Value of the derivative of the shape function N_k with respect to the local coordinate r at the nodal point l $\left(\frac{\partial N_k^l}{\partial r}\right)$.
DNKLDS	Value of the derivative of the shape function N_k with respect to the local coordinate s at the nodal point l $\left(\frac{\partial N_k^l}{\partial s}\right)$.
D2NKIDR2	Value of the second derivative of the shape function N_k with respect to the local coordinate r at the integration point i $\left(\frac{\partial^2 N_k^i}{\partial r^2}\right)$.
D2NKIDRDS	Value of the second derivative of the shape function N_k with respect to the local coordinates r and s at the integration point i $\left(\frac{\partial^2 N_k^i}{\partial r \partial s}\right)$.
D2NKIDS2	Value of the second derivative of the shape function N_k with respect to the local coordinate s at the integration point i $\left(\frac{\partial^2 N_k^i}{\partial s^2}\right)$.

Remarks:

- Interpolation Shell Elements.** For post-processing and the treatment of contact boundary conditions, the use of interpolation shell elements (see *ELEMENT_-INTERPOLATION_SHELL and *CONSTRAINED_NODE_INTERPOLATION) is necessary.
- Input Order.** Data input order for the NMNP nodal points must match the definition of the connectivity of the element in *ELEMENT_GENERALIZED_-SHELL.

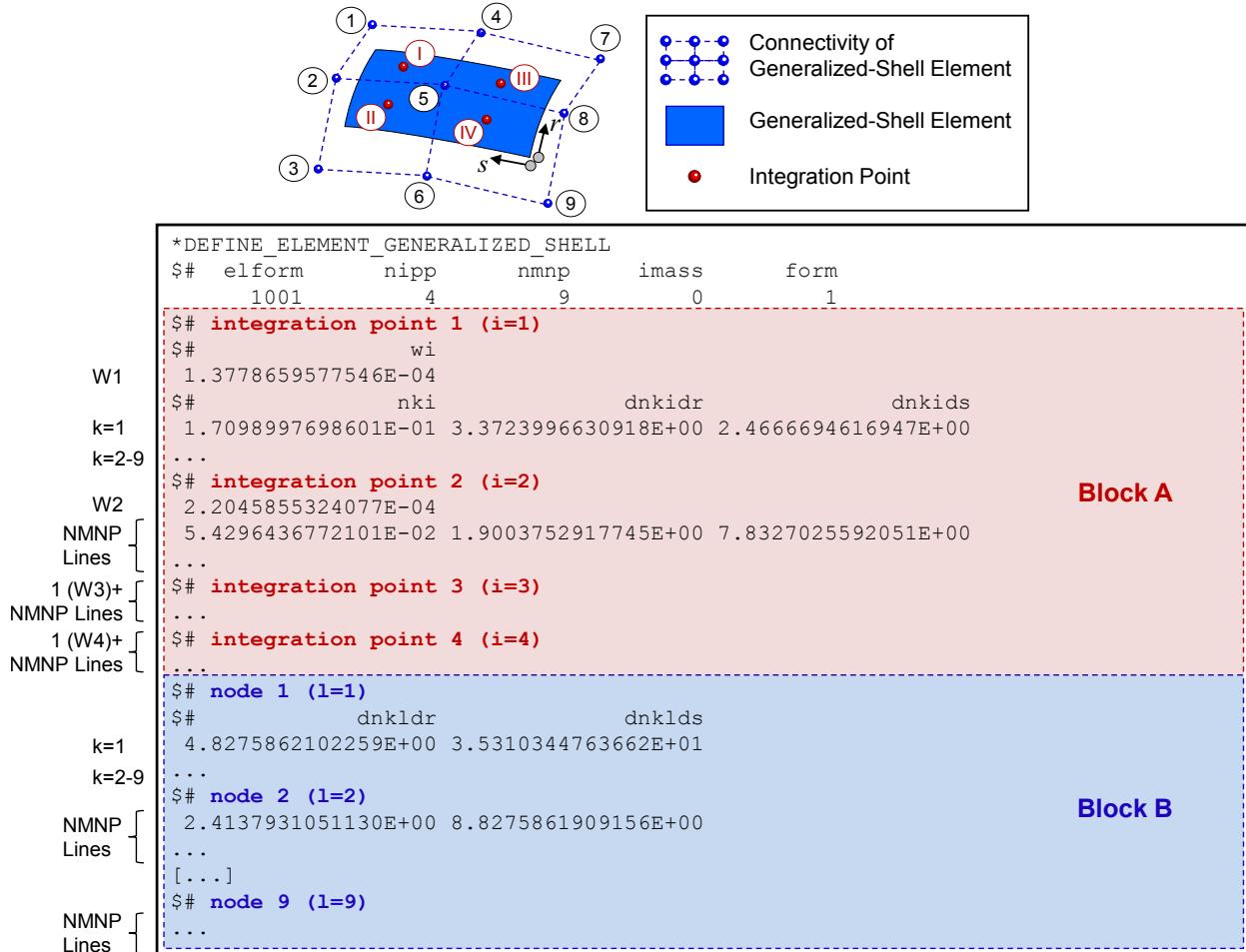
Example:

Figure 17-40. Example of a generalized shell formulation with *DEFINE_ELEMENT_GENERALIZED_SHELL.

*DEFINE

*DEFINE_ELEMENT_GENERALIZED_SOLID

*DEFINE_ELEMENT_GENERALIZED_SOLID

Purpose: Define a general 3D solid formulation to be used in combination with *ELEMENT_GENERALIZED_SOLID. New solid element formulations may be rapidly prototyped in the input deck using this feature.

All necessary information, like the values of the shape functions and their derivatives at all integration points, must be defined using this keyword. The example shown in [Figure 17-41](#) illustrates this procedure by prototyping an 18-noded generalized solid element with 8 integration points. The element formulation ID (called EFORM) used in this keyword needs to be greater or equal than 1000 and will be referenced through *SECTION_SOLID (see [Figure 19-5](#) in *ELEMENT_GENERALIZED_SOLID).

Card 1	1	2	3	4	5	6	7	8
Variable	EFORM	NIP	NMNP	IMASS				
Type	I	I	I	I				
Default	none	none	none	none				

Weights and Shape Function Values/Derivative at Gauss Points:

These cards are read according to the following pseudo code:

```
for i = 1 to NIP {  
    read Card 2(i)  
    for k = 1 to NMNP {  
        read Card 3(i,k)  
    }  
} // comment: Read in NIP × (1 + NMNP) cards
```

Weight Cards. Provide weight for integration point *i*.

(Card 2) <i>i</i>	1	2	3	4	5	6	7	8
Variable	WI							
Type	F							

DEFINE_ELEMENT_GENERALIZED_SOLID**DEFINE**

Integration Point Shape Function Value/Derivatives Cards. Provide the value of the k^{th} shape function and its derivative at the i^{th} integration point.

(Card 3) $_{ik}$	1	2	3	4	5	6	7	8
Variable	NKI		DNKIDR		DNKIDS		DNKIDT	
Type	F		F		F		F	

VARIABLE	DESCRIPTION
ELFORM	Element Formulation ID referenced by *SECTION_SOLID to connect *ELEMENT_GENERALIZED_SOLID with the appropriate solid formulation. The chosen number needs to be greater or equal than 1000.
NIP	Number of integration points.
NMNP	Number of nodes for this element formulation.
IMASS	Option for lumping of mass matrix: EQ.0: row sum EQ.1: diagonal weighting.
WI	Integration weight at integration point i .
NKI	Value of the shape function N_k evaluated at integration point i .
DNKIDR	Value of the derivative of the shape function N_k with respect to the local coordinate r at the integration point i $\left(\frac{\partial N_k^i}{\partial r}\right)$.
DNKIDS	Value of the derivative of the shape function N_k with respect to the local coordinate s at the integration point i $\left(\frac{\partial N_k^i}{\partial s}\right)$.
DNKIDT	Value of the derivative of the shape function N_k with respect to the local coordinate t at the integration point i $\left(\frac{\partial N_k^i}{\partial t}\right)$.

*DEFINE

*DEFINE_ELEMENT_GENERALIZED_SOLID

Remarks:

1. **Post-processing.** For post-processing the use of interpolation solid elements (see *ELEMENT_INTERPOLATION_SOLID and *CONSTRAINED_NODE_INTERPOLATION) is necessary.
2. **Input order.** Data input order for the NMNP nodal points must correlate with the definition of the connectivity of the element in *ELEMENT_GENERALIZED_SOLID.

Example:

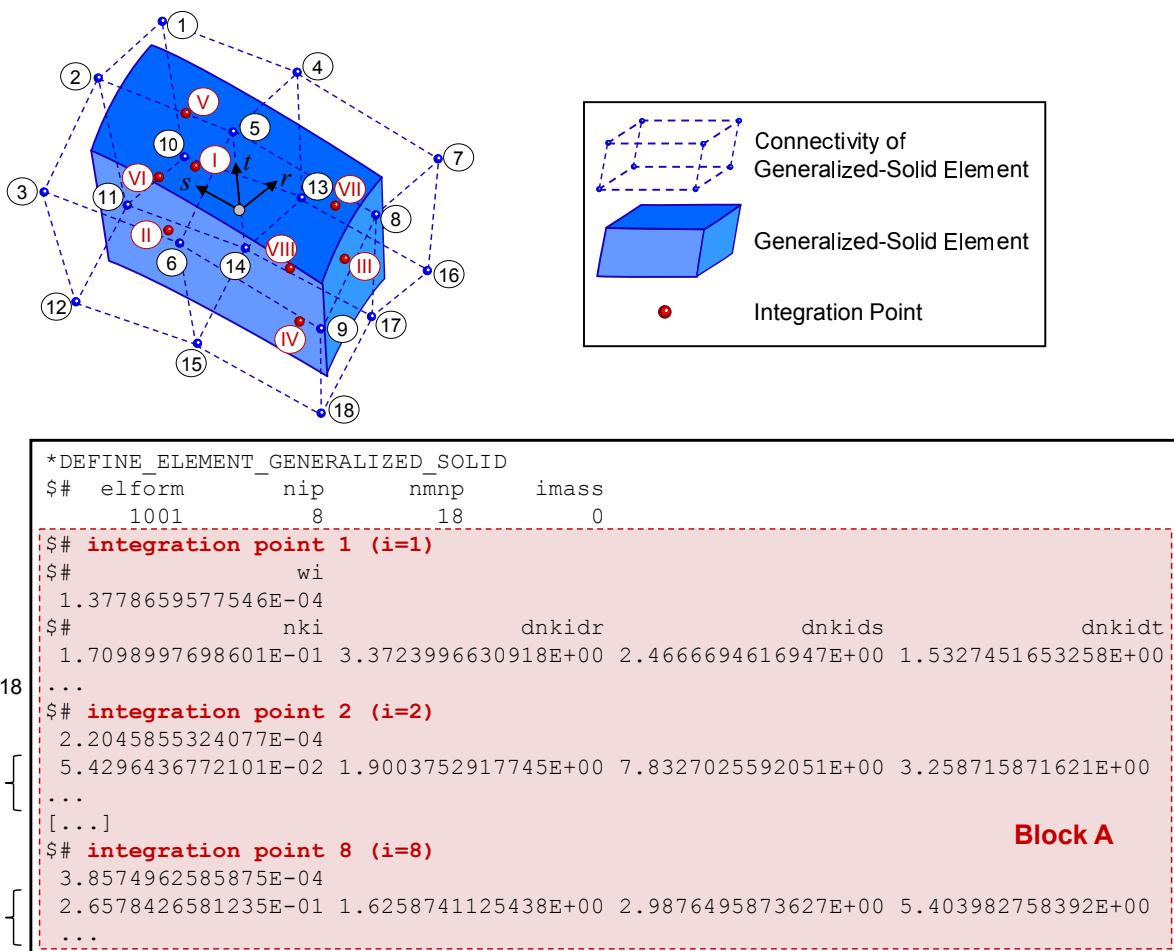


Figure 17-41. Example of a generalized solid formulation with *DEFINE_ELEMENT_GENERALIZED_SOLID

***DEFINE_FABRIC_ASSEMBLIES**

Purpose: Define lists of part sets to properly treat fabric bending between parts.

Define as many cards as needed for the assemblies, using at most 8 part sets per card. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	SPID1	SPID2	SPID3	SPID4	SPID5	SPID6	SPID7	SPID8
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

VARIABLE**DESCRIPTION**

SPID n Part set ID that comprises an assembly.

Remarks:

The materials *MAT_FABRIC and *MAT_FABRIC_MAP are equipped with an optional coating feature to model the fabric's bending resistance. See the related parameters ECOAT, SCOAT and TCOAT on these material model manual entries.

The default behavior for these coatings, *which this keyword changes*, excludes T-intersections, and, furthermore requires that all fabric elements must have a consistently oriented normal vector. In [Figure 17-42](#), the left connection of fabric elements is permitted by the default functionality while the right one is not. However, with using this keyword the proper bending treatment for the right connectivity can be activated by adding the following input to the deck

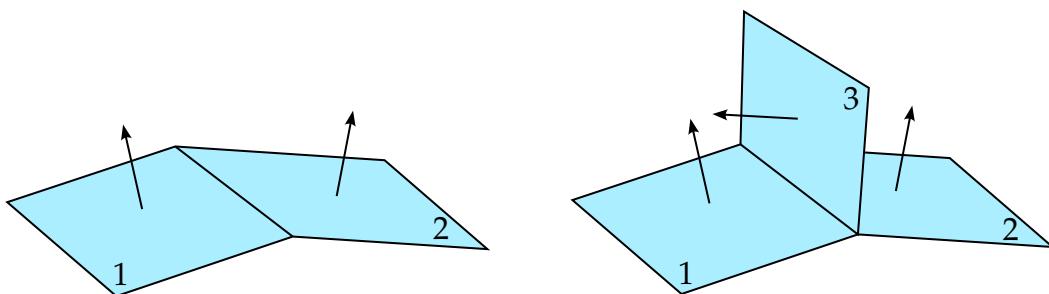


Figure 17-42. Bending in fabric, intended use of *DEFINE_FABRIC_ASSEMBLIES. The numbers indicate the parts the elements belong to.

```
*SET_PART_LIST  
1  
1,2  
*SET_PART_LIST  
2  
3  
*DEFINE_FABRIC_ASSEMBLIES  
1,2
```

which decouples part 3 from the other two parts in terms of bending, thus creating a moment free hinge along the edge between part sets 1 and 2. Bending between parts 1 and 2 is unaffected since these are contained in the same fabric assembly.

For several instances of this keyword in an input deck, the list of assemblies is appended. If assemblies are defined and there happens to be fabric parts that do not belong to any of the specified assemblies, then these parts are collected in a separate unlisted assembly. The restriction on consistent normal vectors and on having no T-intersections applies to all elements within an assembly.

DEFINE_FIBERS**DEFINE*****DEFINE_FIBERS**

Purpose: Define carbon fibers and their related properties in a matrix for a one-step inverse forming simulation. This keyword works *only* with the keyword *CONTROL_FORMING_ONESTEP. . This keyword requires a double precision executable. We developed this feature with Ford Motor Company.

Fiber ID and Property Card.

Card 1	1	2	3	4	5	6	7	8
Variable	IDF	IDP	NUMF	N1	N2	EFB	SHR	HRGLS
Type	I	I	I	I	I	F	F/I	F
Default	none	none	none	optional	optional	none	none	1.0

Fiber Orientation Card.

Card 2	1	2	3	4	5	6	7	8
Variable	ALPHA1	ALPHA2	ALPHA3					
Type	F	F	F					
Default	none	none	none					

Reference Fiber Orientation Card. Define this card if N1 and N2 in Card 1 are undefined or zero.

Card 2	1	2	3	4	5	6	7	8
Variable	X1	Y1	Z1	X2	Y2	Z2		
Type	F	F	F	F	F	F		
Default	none	none	none	none	none	none		

*DEFINE

*DEFINE_FIBERS

VARIABLE	DESCRIPTION
IDF	ID of a fiber set to be defined. It must be unique number; see Figure 17-43 .
IDP	Part ID of the matrix material associated with the fiber set; see Figure 17-43 .
NUMF	Number of fiber orientations; see Figure 17-43 .
N1, N2	Direction from Node 1 (N1) to Node 2 (N2) defines the reference fiber orientation; see Figure 17-43 . If not defined or zero, Card 3 is required.
EFB	Effective stiffness of the fiber in its orientation which typically equals Young's Modulus times fiber cross sectional area fraction. Fiber cross sectional area fraction is typically between 0.25 and 0.5.
SHR	Shear stiffness of the fiber: GT.0: shear stiffness, LT.0: SHR is the load curve ID defining shear stiffness as a function of shear strain.
HRGLS	Hourglass coefficient for stiffness type hourglass control.
ALPHA1, ALPHA2, ALPHA3	Initial orientation angles of the first, second and third fibers relative to reference fiber orientation defined by N2 – N1, respectively.
X1, Y1, Z1, X2, Y2, Z2	If N1 and N2 are undefined, or zero, Card 3 is required to define the coordinates for N1 and N2. The Z1 and Z2 coordinate values must be defined close to the part. Based on the coordinate inputs, LS-DYNA will find the nearest nodes to define N1 and N2 from the model. See Figure 17-43 .

Remarks:

- Matrix Material.** *MAT_024 is supported to define the matrix material.
- Predicted Flat Blank Orientation.** The method of orienting the unfolded blank with three nodes defined in *CONTROL_FORMING_ONESTEP_AUTO_CONSTRAINT is disabled when *DEFINE_FIBERS is used. Use the dynain (not re-positioned.k) file for the predicted flat blank.

3. **Coulomb Friction.** Coulomb friction can be specified with *CONTROL_FORMING_ONESTEP_FRICTION.
4. **Draw Bead Forces.** Use of *CONTROL_FORMING_ONESTEP_DRAWBEAD is not recommended.
5. **Matrix Element Direction.** Element direction of the matrix must be consistent. The directions can be displayed in LS-PrePost4.6, under *EleTol* → *EleEdit* → *Direction* → *Shell* → *Vector* → *Apply*. To align all element directions with an existing element direction, use *EleTol* → *EleEdit* → *Direction* → *Shell* → *Orient* → *Pick seed* → *Apply*.
6. **Output for Visualization.** *ELEMENT_FIBER_INFO and *ELEMENT_BEAM (new part IDs) are automatically created and output in onestepresult file to display the fiber orientations after forming in LS-PrePost; see [Figure 17-43](#).

Additional history variables are written in the onestepresult file to display various angles in color contours (see [Figure 17-43](#)):

- a) History variable #1: the angle between two fibers,
- b) History variable #2: the angle between the first fiber and the element direction,
- c) History variable #3: the angle between the second fiber and the element direction.

These color contour fringe plots can be overlaid with beam elements to verify the fiber orientations. To request history variable output, set NEIPS in *DATABASE_EXTENT_BINARY to a minimum of 3.

Example:

A partial keyword example below defines a fiber set ID 1 in a matrix material with part ID 121. The first and second fiber orientations are at 45° and 135°, respectively, from the reference fiber direction.

```
$-----1-----2-----3-----4-----5-----6-----7-----8
*DEFINE_FIBERS
$      IDF      IDP      NUMF      N1      N2      Efb      SHR      HRGLS
$      1        121      2          1200000.0    -1011     1.0
$      ALPHA1    ALPHA2    ALPHA3
$      45.0     135.0
$      X1       Y1       Z1       X2       Y2       Z2
$      2650.0   800.0    808.0   2940.0   800.0   800.0
*DEFINE_CURVE
1011
0.0,0.0
0.1,1.0
0.5,5.0
```

*DEFINE

*DEFINE_FIBERS

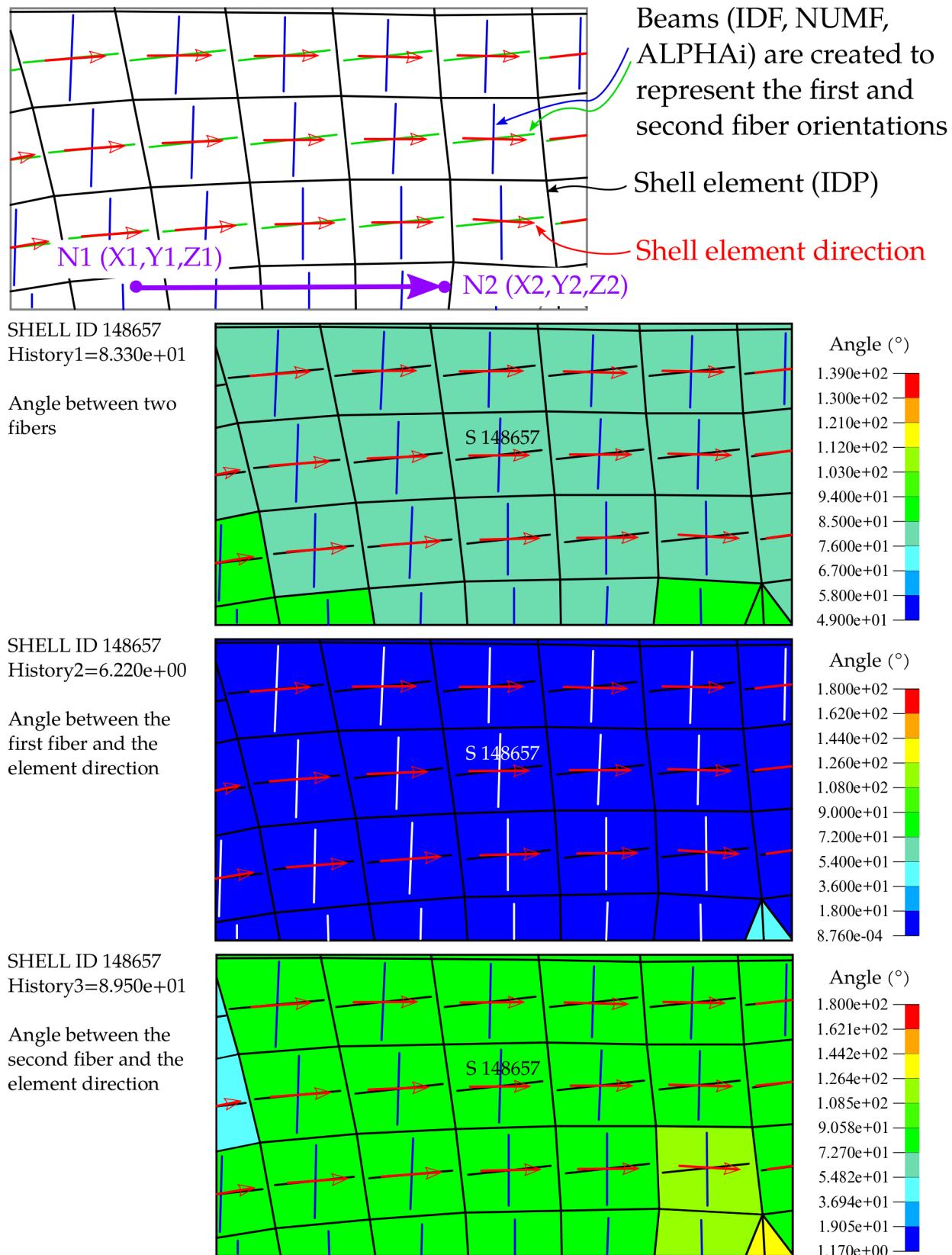


Figure 17-43. *DEFINE_FIBERS inputs and outputs.

***DEFINE_FIELD_{OPTION}**

Available options include:

<BLANK>

TITLE

Purpose: Define a field associated with a point cloud; see [*DEFINE_POINT_CLOUD](#). Field data, defined on a point cloud, are automatically interpolated at required locations for the purpose of analysis. The interpolation is performed through radial basis functions. Parameters defined on *DEFINE_FIELD control the interpolation.

Card Summary:

Card Title. This card is included if the TITLE keyword option is used.

TITLE							
-------	--	--	--	--	--	--	--

Card 1. This card is required.

FID	PCID	NV	LMTSRC	SRCRAD	REFLEN		
-----	------	----	--------	--------	--------	--	--

Card 2. This card is required. Include ceil[NV × number of points/8] cards. This input ends if an empty field is found OR at the next keyword ("*") card.

V1	V2	V3	V4	V5	V6	V7	V8
----	----	----	----	----	----	----	----

Data Card Definitions:

Title Card. Additional card for the TITLE keyword option.

Card Title	1	2	3	4	5	6	7	8
Variable	TITLE							
Type	A80							

VARIABLE	DESCRIPTION
TITLE	Name or description of the field defined in this keyword.

DEFINE**DEFINE_FIELD**

Card 1	1	2	3	4	5	6	7	8
Variable	FID	PCID	NV	LMTSRC	SRCRAD	REFLEN		
Type	I	I	I	I	F	F		
Default	none	none	none	0	0	0		

VARIABLE	DESCRIPTION
FID	Field ID. A unique ID number must be used.
PCID	ID of the point cloud referenced by this field. See *DEFINE_POINT_CLOUD.
NV	Number of data specified for each point of the point cloud PCID. See Remark 1 .
LMTSRC	<p>Flag to restrict the interpolation domain to a subset of data points of the point cloud PCID. This flag activates a search algorithm that restricts the interpolation domain based on a point-to-element distance criterion. For any element associated with a *PART_FIELD that references this field FID:</p> <ul style="list-style-type: none"> EQ.0: Perform global interpolation by interpolating field data from all data points of point cloud PCID. EQ.1: Restrict the interpolation domain and search for all data points of point cloud PCID whose distance from centroid of the element is \leq a distance determined by SRCRAD.
SRCRAD	<p>Radius of the interpolation domain. If LMTSRC = 0, this parameter is disregarded. If LMTSRC = 1, this parameter restricts the interpolation domain. Based on the value of SRCRAD, the interpolation domain is restricted to all data points whose distance from the element centroid is:</p> <ul style="list-style-type: none"> LT.0: Less than or equal to $SRCRAD \times REFLEN$ EQ.0: Less than or equal to REFLEN GT.0: Less than or equal to SRCRAD
REFLEN	Reference length used for scaling of radial basis functions, and, if required, to restrict the interpolation domain.

VARIABLE	DESCRIPTION
	EQ.0: For any element, the solver automatically computes a reference length based on the maximum distance of all its integration points.
	GT.0: User-defined reference length. Any value can be specified.

Field Data Cards. Include $\text{ceil}[\text{NV} \times \text{number of points}/8]$ cards. This input ends at the first missing field or the next keyword ("**") card.

Card 2	1	2	3	4	5	6	7	8
Variable	V1	V2	V3	V4	V5	V6	V7	V8
Type	F	F	F	F	F	F	F	F

VARIABLE	DESCRIPTION
v_i	Field data specified at points of the point cloud PCID. See Remark 2 . $\text{NV} \times \text{number of points}$ defined in PCID should be defined. The data should be input such that the first NV data are for the first point, the second NV data are for the second point, etc.

Remarks:

1. **Field dimension.** The format of *DEFINE_FIELD is general in the sense that no distinction is made between scalar, vector, or tensor data at the keyword level.
2. **Data type.** A field represents a consistent set of data associated with one physical property or parameter.
3. **Precision.** To specify more significant digits for the field data, invoke the "long format" for this keyword, meaning *DEFINE_FIELD_{OPTION}+. For the "long format," the format of Card 1 is (4I20,2F20), while the format of Card(s) 2 is (8F20). The precision of the executable may also limit the actual number of significant digits of the input field data used for the analysis. See [Long Format Input](#) in the section [General Card Format](#) of [Getting Started](#) in this manual for more details regarding the number of significant digits and the "long format."

*DEFINE

*DEFINE_FILTER

*DEFINE_FILTER

Purpose: Define a general purpose filter, currently used by this option:

SENSOR_SWITCH

Card 1	1	2	3	4	5	6	7	8
Variable	ID			TITLE				
Type	I			A70				

Card 2	1	2	3	4	5	6	7	8
Variable	TYPE	DATA1	DATA2	DATA3	DATA4	DATA5	DATA6	DATA7
Type	A10							

VARIABLE	DESCRIPTION
ID	Identification number.
TITLE	Title for this filter.
TYPE	One of the 3 currently defined filter types: DISCRETE, CONTINUOUS, or CHAIN
DATA1-7	Filter type specific data, which determines what the filter does.

Filter Types:

FILTER	DESCRIPTION
DISCRETE	The discrete filter operates on a fixed number of values of the input data. The first data field is an A10 character field, which gives the type of operation the filter performs: MIN, MAX, and AVG are the available options. The second data field is an I10 field, giving the number of input values over which the minimum, maximum, or average is computed.

FILTER	DESCRIPTION
CONTINUOUS	Like the DISCRETE filter, except that it operates over a fixed time interval. The first data field is exactly the same as for the DISCRETE option. The second data field is an F10 field, indicating the duration of the filter. For example, if AVG is given, and the duration is set to 0.1, a running timestep weighted average is computed over the last 0.1 time of the simulation.
CHAIN	Here, data fields 1-7 are all I10 fields, and give the IDs of a list of other filters (including other CHAIN filters, if desired), each of which will be applied in order. So the raw data is fed to the filter indicated by DATA1. The output of that is fed to the next filter, and so on, with up to 7 filters in the chain. List only as many filters as you need.

DEFINE**DEFINE_FORMING_ADAPTIVE_PART_INITIAL*****DEFINE_FORMING_ADAPTIVE_PART_INITIAL**

Purpose: Refine the mesh of a deformable part set in local regions at time zero. This keyword is intended for refining the mesh of blank parts in small regions for spot welding simulations or where contact occurs between the blank and rigid pins in gravity loading processes. In the case of rigid pins, this keyword enables contact to occur between the pins and the blank because, otherwise, the pins are too small compared to the blank's initial coarse mesh. Thus, contact is missed without this refinement. The local regions for refinement on the blank are determined by projecting the parts of local regions onto the blank.

Card 1	1	2	3	4	5	6	7	8
Variable	PSTID	PSTBID	VX	VY	VZ	SMIN	IMOVE	
Type	I	I	F	F	F	F	I	
Default	none	none	0.0	0.0	0.0	0.0	0	

VARIABLE	DESCRIPTION
PSTID	Part set of local regions that will be projected onto the parts of the blank (PSTBID) to determine where the blank will be refined
PSTBID	Part set of the deformable body (blank) to be refined
VX, VY, VZ	Vector for projecting the parts of PSTID onto the parts of PSTBID
SMIN	Minimum element size after refinement
IMOVE	Flag for treatment of the vector (VX, VY, VZ): EQ.0: Use (VX, VY, VZ) as the projection vector. EQ.1: Ignore (VX, VY, VZ) and use the movement vector defined with *PART_MOVE instead for the projection. This option may be ideal when including this keyword with *INCLUDE_TRANSFORM because the vector changes from the transformation. This option enables determining the vector only once.

DEFINE_FORMING_BLANKMESH**DEFINE*****DEFINE_FORMING_BLANKMESH**

Purpose: A rectangular sheet metal blank can be defined with square elements and an arbitrary mesh orientation. This blank can, then, be trimmed with *ELEMENT_BLANKING and *DEFINE_CURVE_TRIM_2D into a developed blank with a complex periphery and inner hole cutouts. This keyword is renamed from the previous keyword *CONTROL_FORMING_BLANKMESH.

Card 1	1	2	3	4	5	6	7	8
Variable	IDMSH	ELENG	LENG1	LENG2	ANGLE	NPLANE	CID	
Type	I	F	F	F	F	I	I	
Default	none	0.0	0.0	0.0	0.0	1	0	

Card 2	1	2	3	4	5	6	7	8
Variable	PIDBK	NID	EID	XCENT	YCENT	ZCENT	SHIFT1	SHIFT2
Type	I	I	I	F	F	F	F	F
Default	1	1	1	0.0	0.0	0.0	0.0	0.0

VARIABLE	DESCRIPTION
IDMSH	A unique ID for the blank mesh (not the blank PID) to be generated
ELENG	The edge length of the square element to be generated.
LENG1	First edge length of the rectangular blank along the following axis in the coordinate system (CID) defined: NPLANE.EQ.1: X-axis NPLANE.EQ.2: Y-axis NPLANE.EQ.3: Z-axis
LENG2	Second edge length of the rectangular blank along the following axis in the coordinate system (CID) defined: NPLANE.EQ.1: Y-axis

DEFINE**DEFINE_FORMING_BLANKMESH**

VARIABLE	DESCRIPTION
	NPLANE.EQ.2: Z-axis NPLANE.EQ.3: X-axis
ANGLE	An angle about the following axis of the CID defined, starting from its respective axis, to rotate the blank (the orientation of the mesh) to be generated. The sign of the rotation angle follows the right-hand rule. NPLANE.EQ.1: about Z-axis, starting from the positive X-axis NPLANE.EQ.2: about X-axis, starting from the positive Y-axis NPLANE.EQ.3: about Y-axis, starting from the positive Z-axis
NPLANE	A plane in which a flat blank to be generated, in reference to the coordinate system defined (CID): EQ.1: XY-plane (default) EQ.2: YZ-plane EQ.3: XZ-plane
CID	ID of a local coordinate system defined by *DEFINE_COORDINATE_SYSTEM. Default is 0 representing the global coordinate system.
PIDBK	Part ID of the blank, as defined by *PART
NID	Starting node ID of the blank to be generated
EID	Starting element ID of the blank to be generated
XCENT	X-coordinate of the center of the blank
YCENT	Y-coordinate of the center of the blank
ZCENT	Z-coordinate of the center of the blank
SHIFT1	First shift distance of the blank in the following axis in the coordinate system (CID) defined. NPLANE.EQ.1: X-axis NPLANE.EQ.2: Y-axis NPLANE.EQ.3: Z-axis
SHIFT2	Second shift distance of the blank in the following axis in the

VARIABLE	DESCRIPTION
	coordinate system (CID) defined:
	NPLANE.EQ.1: Y axis
	NPLANE.EQ.2: Z-axis
	NPLANE.EQ.3: X-axis

About the Keyword:

A flat and rectangular blank can be defined and meshed, which can be trimmed with IGES curves to a desired blank shape with complex periphery and inner cutouts. The mesh orientation is always parallel to the rectangular blank edge and can be changed by defining the field ANGLE. The blank outlines and inner holes can be defined using IGES curves and included in *DEFINE_CURVE_TRIM_2D (not_3D). This feature is very useful for interactive input deck set up of metal forming simulation.

Example:

A complete keyword example of generating a flat blank with PID 1 is provided below. Referring to [Figure 17-44](#), the rectangular blank mesh is generated in the global XY-plane with its center at the global origin and a size of 1100.0×1050.0 mm. The square element edge length is 12 mm. The node and element ID numbers start at 8000 and 9000, respectively. An inner cut-out curve is provided by the IGES file innerholes.iges, and the blank outer line is defined with the IGES file outerlines.iges. Both files are included in the keyword *DEFINE_CURVE_TRIM_2D. A seed point is defined indicating the portion that remains after trimming.

```

*KEYWORD
*SET_PART_LIST
1
1
*CONTROL_TERMINATION
0.0
*ELEMENT_BLANKING
$# psid
1
*DEFINE_FORMING_BLANKMESH
$ IDMSH ELENG LENG1 LENG2 ANGLE NPLANE CID
      3     12.0   1100.00 1050.0
$ PIDBK NID EID XCENT YCENT ZCENT SHIFT1 SHIFT2
      1     8000  9000
*DEFINE_CURVE_TRIM_2D
$# tcid tctype TFLG TDIR TCTOL TOLN
      11111    2          0  0.250000 1.000000
innerholes.iges
*DEFINE_CURVE_TRIM_2D
$# tcid tctype TFLG TDIR TCTOL TOLN
      11112    2          0  0.250000 1.000000
outerlines.iges
*DEFINE_TRIM_SEED_POINT_COORDINATES
1,264.0,0.0,-200.0

```

```
*PART
Blank
$#    pid      secid      mid
      1          1          1
*SECTION_SHELL
$#    secid    elform    shrf      nip
      1          2  1.000000      1
$#    t1        t2        t3        t4
  0.500000  0.500000  0.500000  0.500000
*MAT_TRANSVERSELY_ANISOTROPIC_ELASTIC_PLASTIC
$      MID      RO      E       PR      SIGY      ETAN      R
      1 7.9000E-9 2.0700E+5  0.300000 179.80000   1.000     1.0
*INTERFACE_SPRINGBACK_LSDYNA
1
*END
```

The blank and mesh orientation can be rotated about a particular axis. Following the right-hand rule, the blank in this case is rotated about Z-axis for a positive 30° (ANGLE=30.0) starting from the positive X-axis, as shown in [Figure 17-45](#).

Revision information:

This feature is available in LS-DYNA starting in Revision 59165. This keyword name was changed from *CONTROL_FORMING_BLANKMESH to *DEFINE_FORMING_BLANKMESH in Revision 69074. The variable NPLANE is available in Revision 69128. Some important updates are available in Revision 123603.

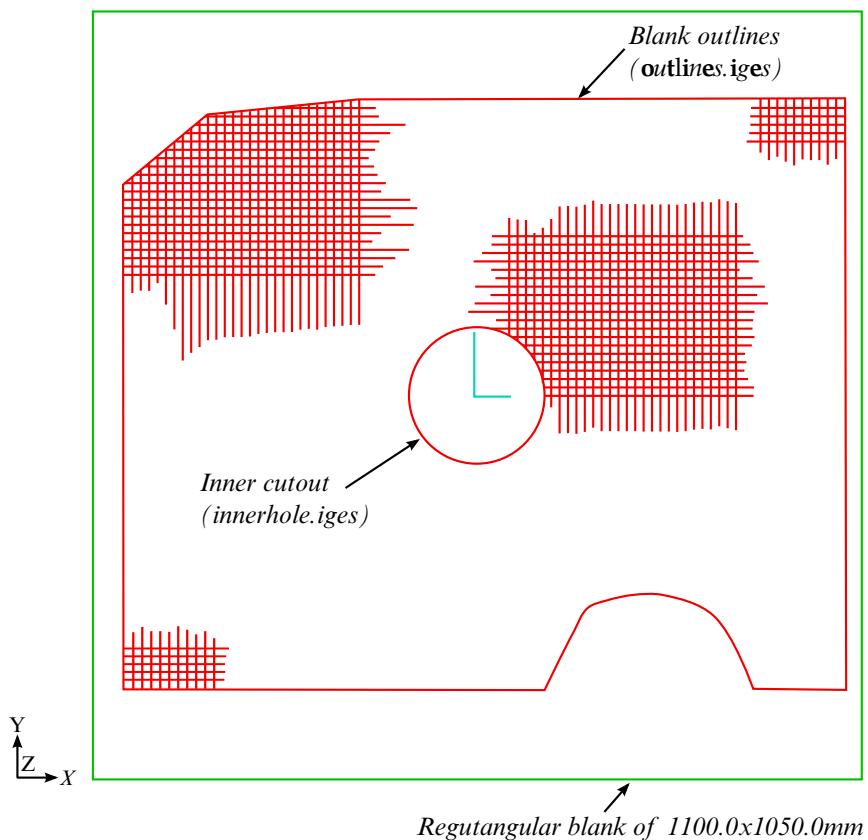


Figure 17-44. A rectangular blank mesh is generated and trimmed into a developed blank shape, with an inner cutout defined by `innerhole.iges` and outline defined by `outlines.iges`.

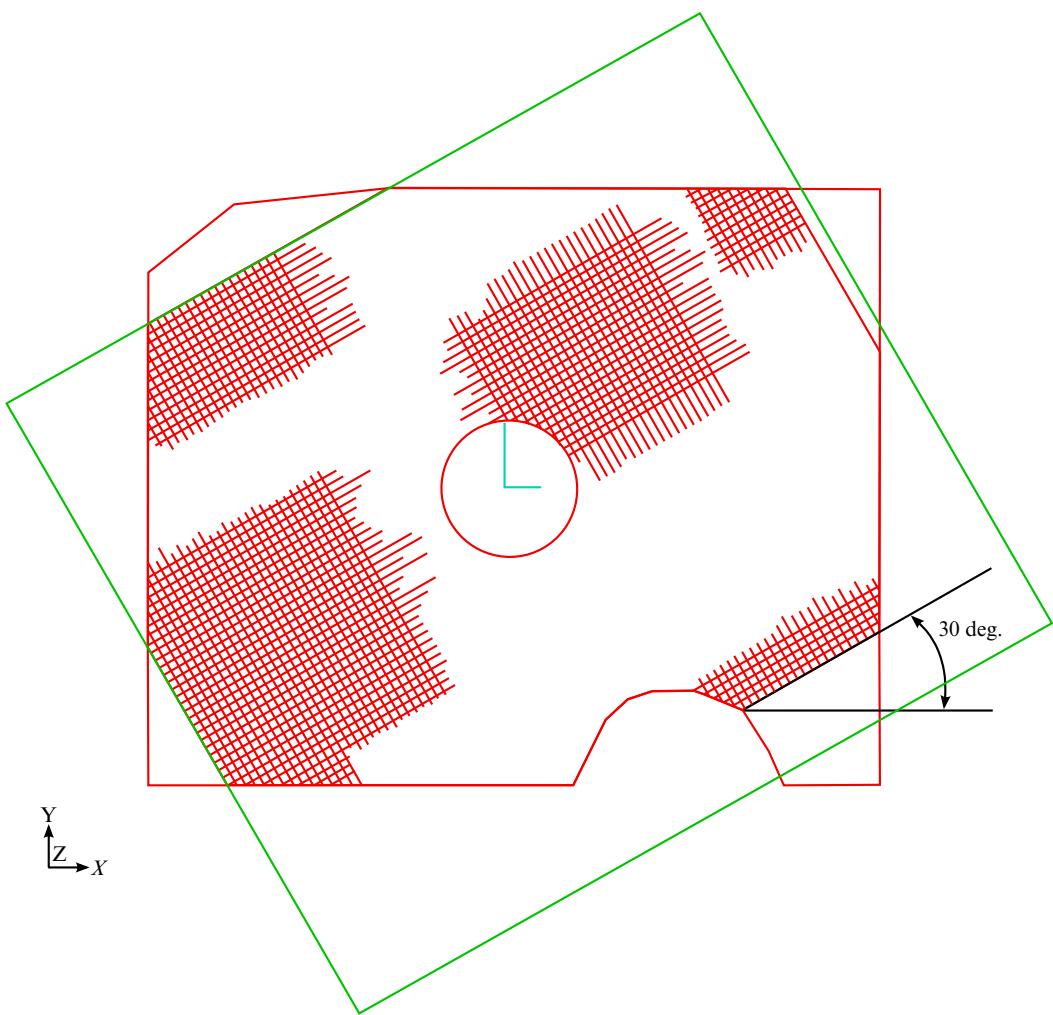


Figure 17-45. A rectangular blank mesh is generated in an angle (ANGLEX = 30) and trimmed into a developed blank shape, with an inner cutout defined by `innerhole.iges` and outline defined by `outlines.iges`. Note the blank mesh orientation is always parallel to the rectangular blank boundary.

***DEFINE_FORMING_CLAMP**

Purpose: This keyword acts as a macro that replaces the combination of cards required to model a clamping process. It is available for double precision executables with the implicit solver. A related keyword is *DEFINE_FORMING_CONTACT.

Define Clamp Card. Define one card for each clamp set. Include as many cards in the following format as desired. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	CLP1	CLP2	VID	GAP	AT	DT		
Type	I	I	I	F	F	F		
Default	none	none	none	0.0	0.0	0.0		

VARIABLE	DESCRIPTION
CLP1	Part ID of a moving rigid body clamp, defined by *PART and *MAT_020 (*MAT_RIGID).
CLP2	Part ID of a fixed rigid body clamp, defined by *PART and *MAT_020. This is sometimes called "net pad".
VID	Define CLP1 moving direction: GT.0: Vector ID from *DEFINE_VECTOR, specifying the moving direction of CLP1 LT.0: Absolute value is a node ID, whose normal vector will be used to define the moving direction of CLP1.
GAP	Final desired distance between CLP1 and CLP2 at the end of clamping.
AT	Begin time for CLP1's move.
DT	Duration of CLP1's move.

About the keyword:

One typical application of this keyword is to estimate springback during the clamping of a formed panel on a checking fixture. Net pads (lower fixed regular or square pads) of a

few millimeters thick are placed according to GD&T (*Geometry Dimensioning and Tolerancing*) requirements on a support platform, typically taking shape of the nominal product. Each net pad (CLP2, see [Figure 17-46](#)) has a corresponding moving clamp (CLP1).

The movable clamp (CLP1) is initially open so that the formed panel can be loaded onto the net pads. Four-way and two-way position gaging pins are used to initially locate and load the panel in the fixture before CLP1 is moved to close with the net pad (CLP2). A white light scan is then performed on the panel, and scan data is processed to ascertain the degree of panel conformance to the required nominal shape. Even with the clamps fully closed, some severely distorted panels will significantly deviate from the nominal shape when the residual stresses from the forming process are too great.

Although, unrelated to this keyword another common method to determine the panel springback amount is the *free state* check which involves a white light scan on the panel secured on a platform but with no additional forces (clamps – CLP1s) applied to deform the panel (to the net pads CLP2s, for example). LS-DYNA can model both methods and job setups can be easily done by selecting *Implicit Static Flanging* process in LS-PrePost 4.2's *Metal Forming Application* → *eZ-Setup*. Furthermore, once springback has been determined, die compensation (*INTERFACE_COMPENSATION_3D) can then be performed to minimize or eliminate the springback; the resulting compensated die shapes can be surfaced, re-machined to produce panels that are within dimensional tolerance.

Since the clamp is, typically, modelled with a much coarser mesh than that on a blank, *CONTACT_FORMING_SURFACE_TO_SURFACE should be used. Rotating-type of clamps are currently not supported.

Application example:

A partial keyword example of using the feature is listed below. Referring to [Figure 17-46](#), the drawn and trimmed blank is positioned between the clamps CLP1 and CLP2. The implicit termination "time" is set at 1.0, with a stepping size of 0.25, for a total of four steps – two steps each for the two CLP1. With the original blank thickness of 1.0 mm, the CLP1s are set to close with the lower CLP2s at "time" of 1.0, leaving a total GAP of 1.02 mm. Note the VIDs are defined as "-46980", indicating that the moving clamps (CLP1) will move in the normal direction defined by Node #46980. The contact definition between the panel and the clamps are defined using *DEFINE_FORMING_CONTACT.

```
*KEYWORD
*INCLUDE
./trimmed.dynain
./nets.k
*CONTROL_TERMINATION
1.0
*CONTROL_IMPLICIT_forming
1
*control_implicit_general
1, 0.25
*CONTROL_SHELL
```

```
:  
*DATABASE_EXTEND_BINARY  
:  
*PART  
Blank  
$ PID SECID MID  
    1      1      1  
Clamp1  
2,2,2  
Clamp2  
3,2,3  
Clamp3  
4,2,2  
Clamp4  
5,2,3  
*MAT_TRANSVERSELY_ANISOTROPIC_ELASTIC_PLASTIC  
$ MID RO E PR SIGY ETAN R HLCID  
    1 2.700E-09 12.00E+04 0.28 0.0 0.0 0.672 2  
*MAT_RIGID  
$# mid ro e pr n couple m alias  
  2 7.8500E-9 2.1000E+5 0.300000  
$# cmo con1 con2  
 1.000000    7    7  
$# lco or a1 a2 a3 v1 v2 v3  
 0.000 0.000 0.000 0.000 0.000 0.000  
*MAT_RIGID  
$# mid ro e pr n couple m alias  
  3 7.8500E-9 2.1000E+5 0.300000  
$# cmo con1 con2  
 1.000000    4    7  
$# lco or a1 a2 a3 v1 v2 v3  
 9997          1.0  
*SETION_SHELL  
1,16,,7  
1.0,1.0,1.0,1.0  
*LOAD_BODY_Z  
 9997          1.0  
*DEFINE_CURVE  
 9997  
    0.0000      9810.0000  
    1.0000      9810.0000  
*DEFINE_FORMING_CLAMP  
$---+---1---+---2---+---3---+---4---+---5---+---6---+---7---+---8  
$ CLP1 CLP2 VID GAP AT DT  
  3     2   -46980  1.02  0.0   0.5  
  5     4   -46980  1.02  0.5   0.5  
$---+---1---+---2---+---3---+---4---+---5---+---6---+---7---+---8  
*DEFINE_FORMING_CONTACT  
$ IPS IPM FS ONEWAY  
  1     2   0.125  1  
  1     3   0.125  1  
  1     4   0.125  1  
  1     5   0.125  1  
*END
```

Note with this keyword, the formed panel needs to be pre-positioned properly with respect to the clamps by users, and auto-position (*CONTROL_FORMING_AUTOPOSITION) cannot be used. Furthermore, prescribed motions and clamp motion curves do not need to be defined for the clamps.

*DEFINE

*DEFINE_FORMING_CLAMP

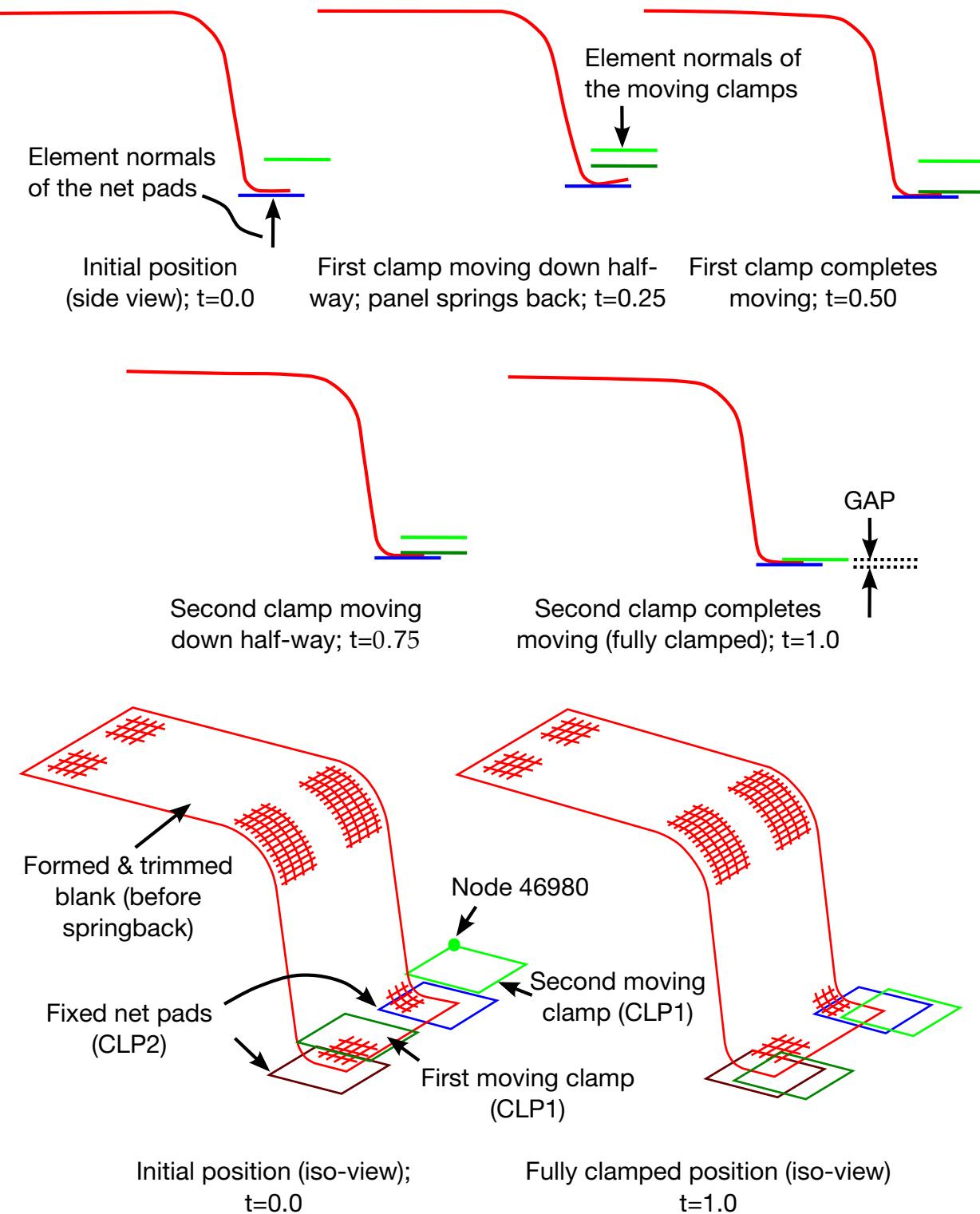


Figure 17-46. Variable definitions and an example of using the negative VID.

***DEFINE_FORMING_CONTACT**

Purpose: This keyword works as a macro for the FORMING_(ONE_WAY)_SURFACE_TO_SURFACE keyword. It adds one contact definition to the model per data card. Each data card consists of a reduced set of fields compared with the full *CONTACT keyword. The omitted fields take their default values. A related keyword is *DEFINE_FORMING_CLAMP.

Define Contact Card. Define one card for each contact interface. Define as many cards in the following format as desired. The input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	IPSA	IPSB	FS	ONEWAY				
Type	I	I	F	I				
Default	none	none	none	0				

VARIABLE	DESCRIPTION
IPSA	Part ID of a SURFA sliding member, typically a deformable sheet metal blank.
IPSB	Part ID of a SURFB sliding member, typically a tool or die defined as a rigid body.
FS	Coulomb friction coefficient
ONEWAY	Define FORMING contact type: EQ.0: The contact is FORMING_ONE WAY_SURFACE_TO_SURFACE. EQ.1: The contact is FORMING_SURFACE_TO_SURFACE.

Application example:

A partial keyword example that defines contact between a deformable part and two pairs of clamps is given below. In [Figure 17-47](#), a blank (PID 1) is defined to have FORMING_SURFACE_TO_SURFACE contact with rigid body clamps (PIDs 2, 3, 4, and 5) with coefficient of frictions for each interface as 0.125, 0.100, 0.125, and 0.100, respectively. Only a total of four lines are needed to define four contact interfaces, as opposed to at least three cards for each interface using the contact keywords.

*DEFINE

*DEFINE_FORMING_CONTACT

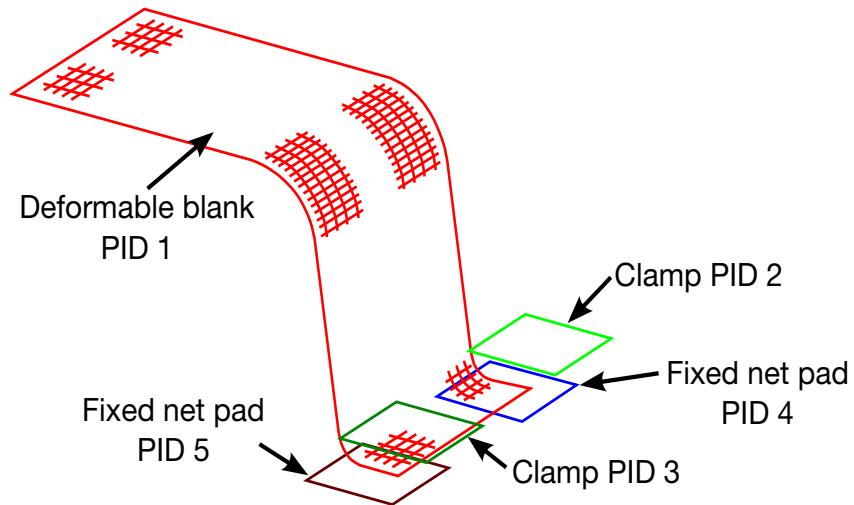


Figure 17-47. Define contact interfaces.

```
$-----1-----2-----3-----4-----5-----6-----7-----8
*DEFINE_FORMING_CONTACT
$    IPSA      IPSB      FS      ONEWAY
     1          2      0.125      1
     1          3      0.100      1
     1          4      0.125      1
     1          5      0.100      1
```

***DEFINE_FORMING_ONESTEP_PRIMARY**

NOTE: This keyword was previously named *DEFINE_FORMING_ONESTEP_MASTER. The old name is supported for backward compatibility. *DEFINE_FORMING_ONESTEP_PRIMARY is the preferred name starting with R13.

Purpose: Define a primary blank to which a second (constrained) blank is connected using *CONSTRAINED_SPOTWELD. Such a configuration is often called “patch-welded blanks.” Note this keyword must be used together with *CONTROL_FORMING_ONESTEP.

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

VARIABLE	DESCRIPTION
SLPID	Part ID of the primary blank to which a constrained blank is welded using *CONSTRAINED_SPOTWELD

Example:

A partial keyword example is given below. Two patches, with PIDs 2 and 3, are spot-welded onto the primary blank (PID 1) using multiple *CONSTRAINED_SPOTWELD keywords. The primary blank is specified using *DEFINE_FORMING_ONESTEP_PRIMARY. The spot welds can be generated using LS-PrePost.

```

*KEYWORD
*TITLE
Patch welded blanks - One Step Simulation
*PARAMETER
$-----1-----2-----3-----4-----5-----6-----7-----8
I nip          3
R dt           0.25
R autobd       1.50
R tsclmax      1.00
R tsclmin      0.80
R epsmax       0.20
I nslovr       2
I ilimit        1
R dctl          0.01

```

DEFINE**DEFINE_FORMING_ONESTEP_PRIMARY**

```
R deltau      0.00
I lsolvr      5
$ Reposition Nodes
I nid1       6344
I nid2       1950
I nid3       4113
$-----1-----2-----3-----4-----5-----6-----7-----8
*PART
Part to be unfolded - 3
$#    pid      secid      mid
     1          1          1
     2          1          1
     3          1          1
*SECTION_SHELL
$#    secid      elform      shrf      nip      propt      qr/irid      icomp      setyp
     1          16      0.833333  &nip      1          0          0          1
$#    t1          t2          t3          t4
     1.000000  1.000000  1.000000  1.000000
$-----1-----2-----3-----4-----5-----6-----7-----8
*INCLUDE
2patches.k
*INCLUDE
sim_Mat.k
*INCLUDE
EZone_step.k
*CONSTRAINED_SPOTWELD_ID
$#    wid
     1
$#    n1          n2          sn          ss          n          m          tf          ep
     8396        2019        0.000      0.000      0.000  0.0001.0000E+201.0000E+20
*CONSTRAINED_SPOTWELD_ID
     3
     8289        1894        0.000      0.000      0.000  0.0001.0000E+201.0000E+20
*CONSTRAINED_SPOTWELD_ID
     13
     8216        1817        0.000      0.000      0.000  0.0001.0000E+201.0000E+20
*CONSTRAINED_SPOTWELD_ID
     14
     1966        8355        0.000      0.000      0.000  0.0001.0000E+201.0000E+20
*CONTROL_FORMING_ONESTEP_QUAD2
$#    option      TSCLMAX      autobd      TSCLMIN      EPSMAX
     7      &tsclmax      &autobd      &tsclmin      &epsmax
*DEFINE_FORMING_ONESTEP_PRIMARY
$    SLPID
     1
*END
```

Revision Information:

This feature is available starting in LS-DYNA Dev Revision 134771 for SMP and double precision only.

DEFINE_FP_TO_SURFACE_COUPLING**DEFINE*****DEFINE_FP_TO_SURFACE_COUPLING**

Purpose: Define a tied coupling interface between fluid particles modeled with implicit smoothed particle Galerkin (ISPG) and a surface.

Card 1	1	2	3	4	5	6	7	8
Variable	FP	SURF	FPTYPE	SURFTYPE				
Type	I	I	I	I				
Default	none	none	1	0				

Card 2	1	2	3	4	5	6	7	8
Variable	SBC	SCA				SFP		
Type	I	F				F		
Default	0	0.5				0.1		

VARIABLE	DESCRIPTION
FP	Part/part set ID for the fluid particles
SURF	Segment set ID specifying the surface. Currently the segment set should be generated from the 8-noded hexahedral elements.
FPTYPE	Type for FP: EQ.0: Part set ID EQ.1: Part ID
SURFTYPE	Type for SURF: EQ.0: Segment set ID
SBC	Type of boundary condition. EQ.0: Free-slip boundary EQ.1: Non-slip boundary

DEFINE**DEFINE_FP_TO_SURFACE_COUPLING**

VARIABLE	DESCRIPTION
SCA	Static (equilibrium) contact angle in radians
SFP	Stiffness coefficient along the normal direction of the contact interface. SFP should be less than 1.0. If SFP is too small, large penetrations can occur.

***DEFINE_FRICTION**

Purpose: Define friction coefficients between parts for use with the contact options:

SINGLE_SURFACE

AIRBAG_SINGLE_SURFACE

AUTOMATIC_GENERAL

AUTOMATIC_SINGLE_SURFACE

AUTOMATIC_SINGLE_SURFACE_MORTAR

AUTOMATIC_NODES_TO_SURFACE

AUTOMATIC_SURFACE_TO_SURFACE

AUTOMATIC_SURFACE_TO_SURFACE_MORTAR

AUTOMATIC_ONE WAY_SURFACE_TO_SURFACE

ERODING_SINGLE_SURFACE

The input ends at the next keyword ("*"). Default friction values are used for any part ID pair that is not defined. Contact interfaces that are defined with SOFT = 2 on optional card A may use this part data for any keyword that is supported by the segment-to-segment contact.

The coefficient tables specified by the following cards are activated when [FS](#) he (see t card of *CONTACT) is set to -2.0. This feature overrides the coefficients defined in [*PART_CONTACT](#) (which are turned on only when FS is set to -1.0).

When only *one* friction table is defined, it is used for *all* contacts having FS set to -2.0. Otherwise, for each contact with FS equal to -2.0, the keyword reader assigns a table to each *CONTACT by matching the value of [FD](#) from *CONTACT with an ID from Card 1 below. Failure to match FD to an ID causes error termination.

DEFINE**DEFINE_FRICTION**

Card 1	1	2	3	4	5	6	7	8
Variable	ID	FS_D	FD_D	DC_D	VC_D	ICNEP		
Type	I	F	F	F	F	I		
Default	none	0.0	0.0	0.0	0.0	0		

Friction *ij* card. Sets the friction coefficients between parts (or part sets) *i* and *j*. Add as many of these cards to the deck as necessary. The next keyword ("*") card terminates the friction definition.

Card 2	1	2	3	4	5	6	7	8
Variable	PID <i>i</i>	PID <i>j</i>	FS <i>ij</i>	FD <i>ij</i>	DC <i>ij</i>	VC <i>ij</i>	PTYPE <i>i</i>	PTYPE <i>j</i>
Type	I	I	F	F	F	F	A	A
Default	none	none	0.0	0.0	0.0	0.0		

VARIABLE**DESCRIPTION**

ID

Identification number.

FS_D

Default value of the static coefficient of friction. The frictional coefficient is assumed to depend on the relative velocity v_{rel} of the surfaces in contact,

$$\mu_c = \text{FD} + (\text{FS} - \text{FD})e^{-\text{DC}|v_{\text{rel}}|}.$$

Default values are used when the part pair is undefined. For Mortar contact, $\mu_c = \text{FS}$, that is, dynamic effects are ignored.

FD_D

Default value of the dynamic coefficient of friction. The frictional coefficient is assumed to depend on the relative velocity v_{rel} of the surfaces in contact,

$$\mu_c = \text{FD} + (\text{FS} - \text{FD})e^{-\text{DC}|v_{\text{rel}}|}.$$

Default values are used when the part pair is undefined. For Mortar contact, $\mu_c = \text{FS}$, that is, dynamic effects are ignored.

VARIABLE	DESCRIPTION
DC_D	Default value of the exponential decay coefficient. The frictional coefficient is assumed to depend on the relative velocity v_{rel} of the surfaces in contact, $\mu_c = \text{FD} + (\text{FS} - \text{FD})e^{-\text{DC} v_{\text{rel}} }.$ <p>Default values are used when the part pair is undefined. For Mortar contact, $\mu_c = \text{FS}$, that is, dynamic effects are ignored.</p>
VC_D	Default value of the coefficient for viscous friction. This is necessary to limit the friction force to a maximum. A limiting force is computed $F_{\text{lim}} = \text{VC} \times A_{\text{cont}},$ <p>where A_{cont} is the area of the segment contacted by the node in contact. The suggested value for VC is to use the yield stress in shear $\text{VC} = \sigma_o / \sqrt{3}$ where σ_o is the yield stress of the contacted material. Default values are used when the part pair is undefined.</p>
ICNEP	Flag to check for non-existing parts, or part sets, (PID_i , PID_j) on Card 2: <p>EQ.0: The existence of parts or part sets is checked, and an error occurs when any is missing (default).</p> <p>EQ.1: The existence of parts or part sets is checked, and lines with non-existent parts will be ignored.</p>
PID_i	Part or part set ID i
PID_j	Part or part set ID j
FS_{ij}	Static coefficient of friction between parts i and j
FD_{ij}	Dynamic coefficient of friction between parts i and j
DC_{ij}	Exponential decay coefficient between parts i and j
VC_{ij}	Viscous friction between parts i and j
PTYPE_i , PTYPE_j	EQ.“PSET”: when PTYPE_i or PTYPE_j refers to a *SET_PART.

*DEFINE

*DEFINE_FRICTION_ORIENTATION

*DEFINE_FRICTION_ORIENTATION

Purpose: Define different coefficients of friction (COF) for specific directions, specified using a vector and angles in degree. In addition, COF can be scaled according to the amount of pressure generated in the contact interface. This keyword works in SMP only with *CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE, including the SMOOTH keyword option. In MPP it works with *CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ORTHO_FRICTION, including the SMOOTH keyword option.

This feature is developed jointly with the Ford Motor Company.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	LCID	LCIDP	V1	V2	V3		
Type	I	I	I	F	F	F		
Default	none	0	0	0.0	0.0	0.0		

VARIABLE

DESCRIPTION

PID Part ID to which directional and pressure-sensitive COF is to be applied. See *PART.

LCID ID of the load curve defining COF as a function of orientation in degrees

LCIDP ID of the load curve defining COF scale factor as a function of pressure

V1, V2, V3 Vector components of vector V defining zero-degree (rolling) direction

Remarks:

Load curves LCID and LCIDP are not extrapolated beyond what are defined. It is recommended that the definition is specified for the complete range of angle and pressure values expected. One edge of all elements on the sheet metal blank must align initially with the vector defined by V1, V2, and V3.

This feature was initially intended for use with *CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE in SMP. It later was further enhanced to be used in

combination with *CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ORTHOFRICTION for MPP applications. With this contact, the fields LCID and LCIDP are overridden by the friction factor input in ORTHO_FRICTION, while the vector **V** defines the first orthogonal direction. It furthermore allows the convenience of defining SURFA and SURFB in *CONTACT as part sets in which case the segment sets necessary for ORTHO_FRICTION are generated automatically with orientation according to vector **V**. A detailed keyword example is shown in [Figure 17-54](#).

Example:

The following is a partial keyword input of using this feature to define directional frictions and pressure-sensitive COF scale factor.

```
*DEFINE_FRICTION_ORIENTATION
$      PID      LCID      LCIDP      V1      V2      V3
      1       15       16      1.0      0.        0.
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$ define COF vs. orientation angle
*DEFINE_CURVE
15
0.0, 0.3
90.0, 0.0
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$ define COF scale factor vs. pressure
*DEFINE_CURVE
16
0.1, 0.3
0.2, 0.3
0.3, 0.3
0.4, 0.2
0.5, 0.5
0.6, 0.4
```

Referring to [Figure 17-48](#), a deformable blank is clamped with 1000 N of force between two rigid plates and is pulled 90 mm in the X-direction using displacement control. Initial and final positions of the blank are shown in [Figure 17-49](#). The normal force is recovered from the RCFORC file, as shown in [Figure 17-50](#), which agrees with what is applied. The frictional force (pulling force) in the X-direction is 89 N, as shown in [Figure 17-51](#). A hand calculation from the input verifies this result:

$$\frac{[\text{clamping force}]}{1000\text{N}} \times \frac{[\text{X-dir coefficient}]}{0.3} \times \frac{[\text{coefficient scale factor at 0.27 pressure}]}{0.3} = 90\text{N}$$

The interface pressure can be output from an LS-DYNA run when 'S=filename' is included in the command line. The binary output can be viewed from LS-PrePost.

The element directions are automatically aligned with the vector **V**. The left side of [Figure 17-52](#) shows the element directions of the incoming sheet blank. The keyword will re-orient the element directions based on the vector **V** specified, which has the component of (1.0,0.0,0.0) in this case. The re-oriented element directions for the blank are shown on the right side of the figure.

*DEFINE

*DEFINE_FRICTION_ORIENTATION

Following the numeric directions provided in [Figure 17-53](#), LS-PrePost4.0 can be used to check the element directions of a sheet blank.

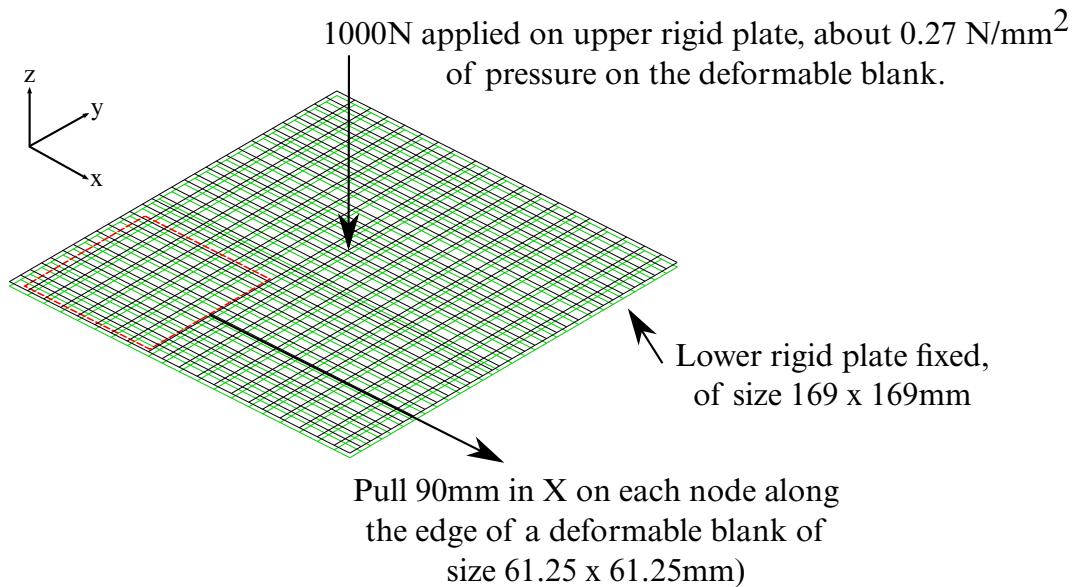


Figure 17-48. Boundary and loading conditions of a small test model.

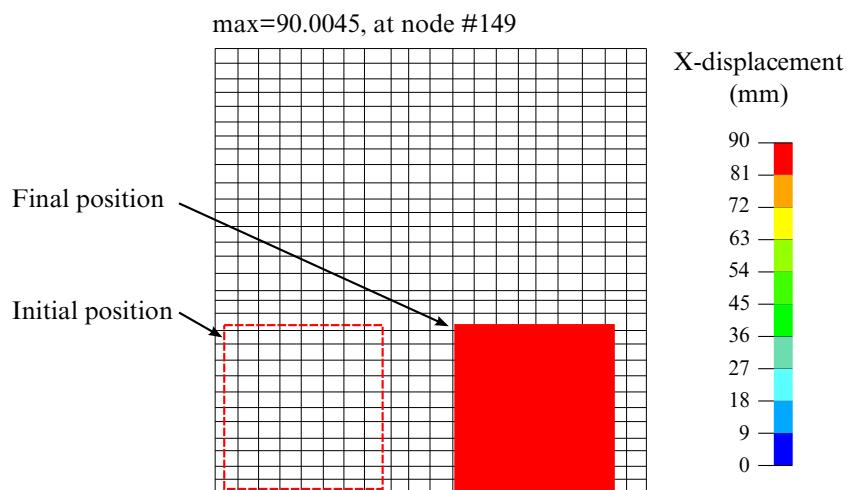


Figure 17-49. Initial and final position of the blank.

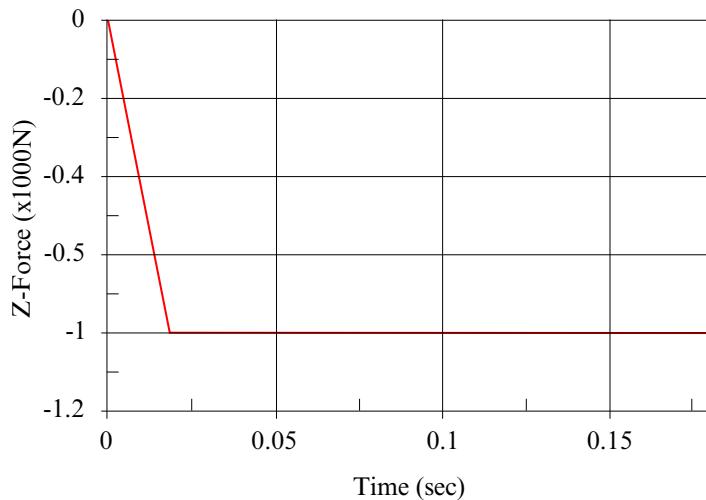


Figure 17-50. Normal force from RCFORC file.

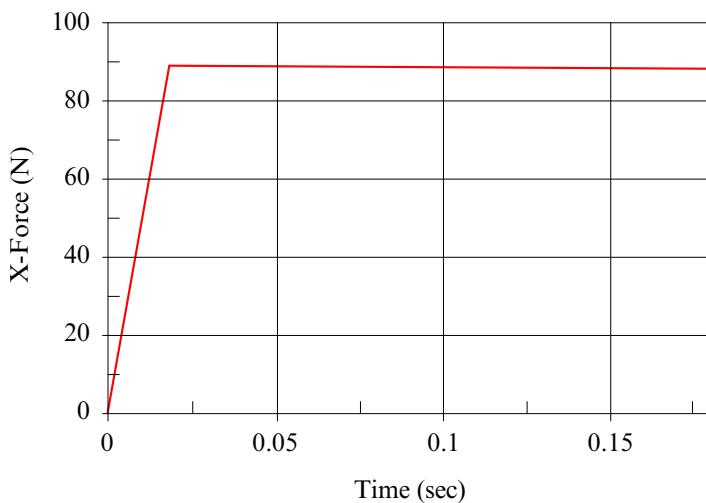


Figure 17-51. Pulling force (frictional force) from RCFORC file.

*DEFINE

*DEFINE_FRICTION_ORIENTATION

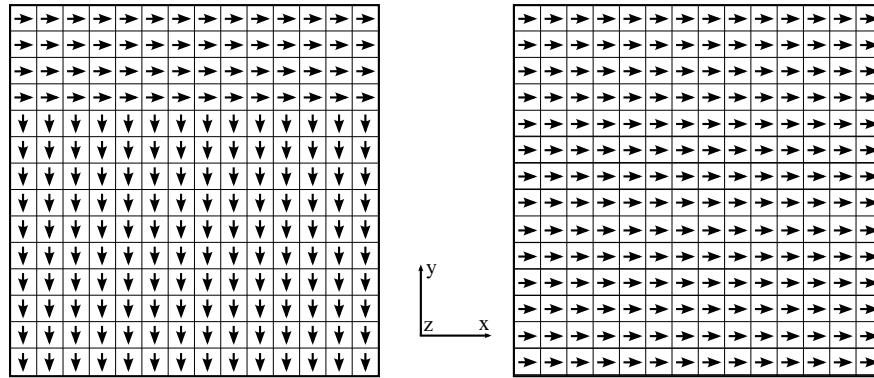


Figure 17-52. Element directions (N1-N2) of an incoming sheet blank (left) and directions after re-orientation.

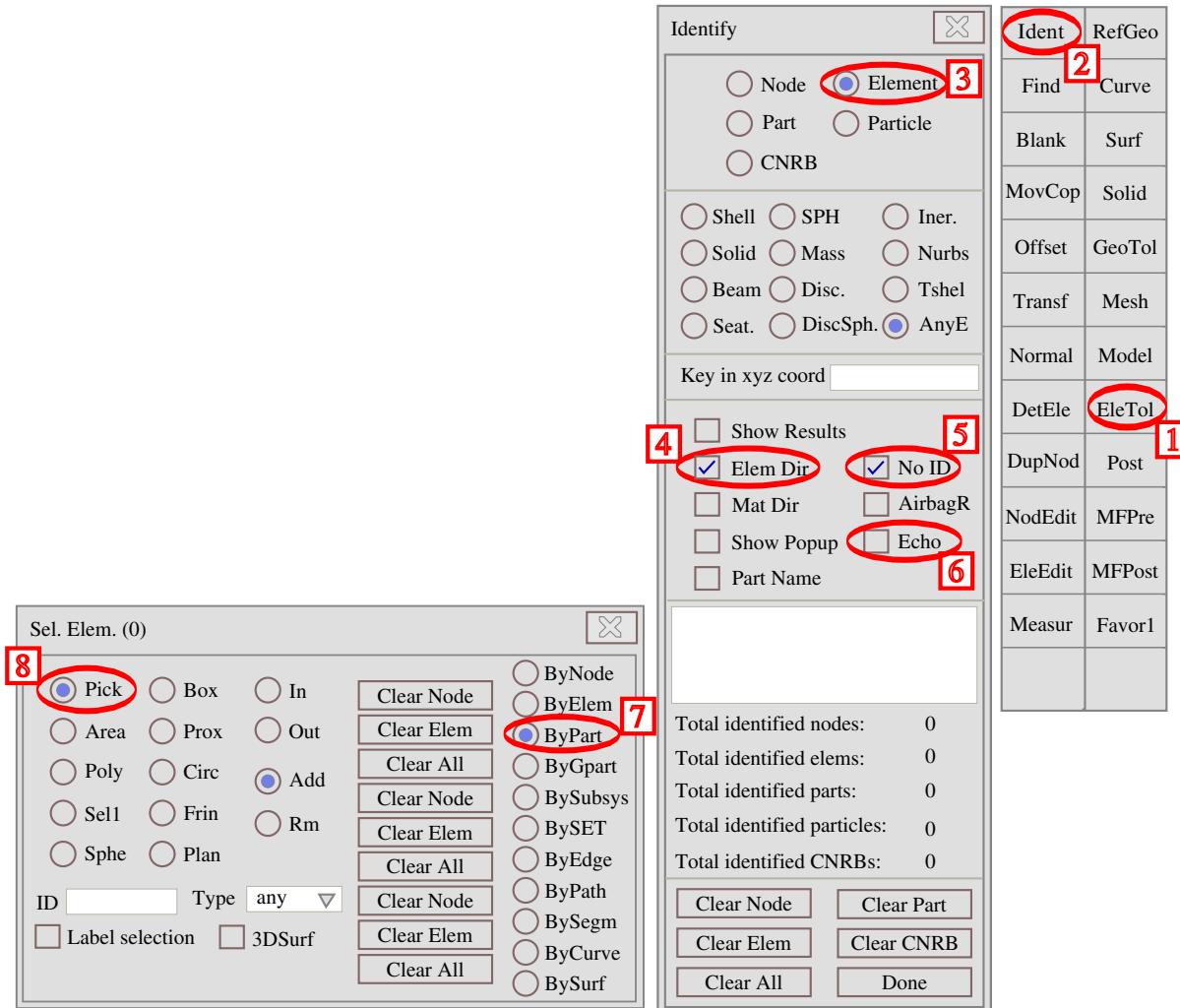


Figure 17-53. Checking element directions (N1-N2) by part using LS-PrePost

```

*DEFINE_FRICTION_ORIENTATION
$ PID    LCID    LCIDP
      1,,,           V1     V2     V3
      .1.0   0.0   0.0
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ORTHO_FRICTION
$ SURFA    SURFB    SURFATYP    SURFBTYP
      1,       3,       2,       2
$ FS      FD      DC      VC
      1.25,  0.0,   , 20.0
$ SFSA    SFSB
      0.0,   0.0
$FS1_SA, FD1_SA, DC1_SA, VC1_SA, LC1_SA, OACS_SA, LCFSA, LCPSA
      0.3,   0.0,   0.0,   0.0,   , 1,   15,   16
$FS2_SA, FD2_SA, DC2_SA, VC2_SA, LC2_SA
      0.1,   0.0,   0.0,   0.0
$FS1_SB, FD1_SB, DC1_SB, VC1_SB, LC1_SB, OACS_SB, LCFSB, LCPSB
      0.3,   0.0,   0.0,   0.0,   , 0,   15,   16
$FS2_SB, FD2_SB, DC2_SB, VC2_SB, LC2_SB
      0.1,   0.0,   0.0,   0.0
*DEFINE_CURVE
$ LCFSA, define COF vs. angle based on 1st orthogonal direction
15
0.00,0.3
45.0,0.2
90.0,0.1
*DEFINE_CURVE
$ LCPSA, define COF scale factor vs. pressure
16
0.0,0.0
0.3,0.3
0.5,0.5

```

Use this keyword/vector to define rolling direction

Use *Set_part_list

FS ignored if ORTHO_FRICTION is present

FS1_SA, LC1_SA ignored if LCFSA, LCPSA are defined:
LCFSA: COF vs. Angle;
LCPSA: COF scale factor vs. Pressure.

LCFSB, LCPSB
FS1_SB, LC1_SB ignored if LCFSB, LCPSB are defined

1st Orthogonal direction follows SURFA segment orientation, as defined by 'a1' in *SET_SEGMENT;
Ignored when defined with *DEFINE_FRICTION_ORIENTATION.

1st Orthogonal direction follows SURFA segment orientation, as defined by 'a1' in *SET_SEGMENT;
Ignored when defined with *DEFINE_FRICTION_ORIENTATION.

Figure 17-54. Use of this keyword with _ORTHO_FRICTION for MPP.

DEFINE**DEFINE_FRICTION*****DEFINE_FRICTION_SCALING**

Purpose: Define scale factors for contact friction for the inner and outer surfaces of shell contact segments. This option is currently only available for contact interfaces that have SOFT set to 2 on optional card A of the *CONTACT input data.

Include as many of Card 1 as desired. The next keyword ("*") card terminates this input. Default friction values are used for any part ID pair that is not defined.

Card 1	1	2	3	4	5	6	7	8
Variable	FSID	CID	PSID	SCALEI	SCALEO			
Type	I	F	F	F	F			
Default	none	0.0	0.0	1.0	1.0			

VARIABLE	DESCRIPTION
FSID	Friction scaling ID number. Each friction scaling definition should have a unique ID which is used for output messages only.
CID	Contact ID. Optional input to limit friction scaling to one contact interface with this ID number.
PSID	Part set ID. Optional input to limit friction scaling to parts in the set.
SCALEI	Friction scale factor for the inner surface of shell segments
SCALEO	Friction scale factor for the outer surface of shell segments

Remarks:

Some materials, such as fabrics, have friction coefficients on one side that differ from the other side. To enable the contact friction to mimic this behavior, separate scale factors can be defined for the inner and outer surface of contact segments that are attached to thin shell elements. The inner and outer surfaces are defined by using the right hand rule with node numbering of the segments. Segment node numbering is consistent with shell element numbering if the contact surface is defined by part or part set.

When modeling an airbag, the inner and outer surface may not be consistent with element or segment numbering, but will depend on how the airbag is defined. When an airbag is

defined using the particle method (see *AIRBAG_PARTICLE), then the SCALEO parameter will scale the friction on the inner surface of the bag, and SCALEI will scale the outer surface. When defined by control volumes (see *AIRBAG), the SCALEI parameter will scale friction on the inner surface of the bag and SCALEO the scale the outer surface.

This option is compatible with all other options to define the friction coefficients as the scale factors are applied after the Coulomb coefficient, μ_c , has been calculated.

*DEFINE

*DEFINE_FUNCTION

*DEFINE_FUNCTION

Purpose: Define a function that can be referenced by a limited number of keyword options. The function arguments are different for each keyword that references *DEFINE_FUNCTION. Unless stated otherwise, all the listed argument(s) in their correct order must be included in the argument list. Some usages of *DEFINE_FUNCTION allow random ordering of arguments and argument dropouts. See the individual keywords for the correct format. Some examples are shown below.

NOTE: Unless the type of the function is explicitly given (as in [Example 2](#) below), the function will return an integer value if the name of the function starts with a letter in the range of i - n. Otherwise it will return a real value. For example, if a function is defined as “ifunc(x) = sqrt(x),” then ifunc(2.0) will return 1, not 1.414.

The TITLE option is not allowed with *DEFINE_FUNCTION.

Card 1	1	2	3	4	5	6	7	8
Variable	FID				HEADING			
Type	I				A70			

Function Cards. Insert as many cards as needed. These cards are combined to form a single line of input. The next keyword (“**”) card terminates this input.

Card	1	2	3	4	5	6	7	8
Variable				FUNCTION				
Type				A80				

VARIABLE	DESCRIPTION
FID	Function ID. Functions, tables (see *DEFINE_TABLE), and load curves may not share common IDs. A unique number must be defined.
HEADING	An optional descriptive heading
FUNCTION	Arithmetic expression involving a combination of independent variables and other functions, that is, “f(a,b,c)=a*2+b*c+sqrt(a*c),” where a, b, and c are the independent variables. The function name, “f(a,b,c),” must be unique since other functions can then use

VARIABLE	DESCRIPTION
	and reference this function. For example, “g(a,b,c,d)=f(a,b,c)**2+d.” In this example, two *DEFINE_FUNCTION definitions are needed to define functions f and g.

Remarks:

1. **Reserved Names for Variables and Constants.** Certain useful constants are intrinsic and available through reserved names. For example, PI is the proportionality constant relating the circumference of a circle to its diameter, while DTOR and RTOD are, respectively, used to convert from degrees to radians and from radians to degrees. Also, TIME stores the current simulation time. See Appendix U for a complete list of variable names reserved internally by LS-DYNA.
2. **Trigonometric and Other Intrinsic Functions.** Most trigonometric and other mathematical functions available in the Fortran and C programming languages are valid in *DEFINE_FUNCTION, such as SIN, COS, ABS, and MAX. Please see a more complete list under “Intrinsic Functions” in *DEFINE_CURVE_FUNCTION.
3. **Units of Angular Measures.** Unless otherwise noted units of radians are always used for the arguments and output of functions involving angular measures.
4. **Dynamic Relaxation.** Unlike *DEFINE_CURVE and *DEFINE_CURVE_FUNCTION, *DEFINE_FUNCTION is always active in dynamic relaxation phase.

The following examples serve only as an illustration of syntax.

Example 1:

Prescribe sinusoidal x-velocity and z-velocity for some nodes.

```
*BOUNDARY_PRESCRIBED_MOTION_SET
$#    nsid      dof      vad      lcid      sf
      1          1        0          1
      1          3        0          2
*DEFINE_FUNCTION
1,x-velo
x(t)=1000*sin(100*t)
*DEFINE_FUNCTION
2,z-velo
a(t)=x(t)+200
```

Example 2:

Ramp up a hydrostatic pressure on a submerged surface.

```
*comment
units: mks

Apply a hydrostatic pressure ramped up over a finite time = trise.

pressure on segment = rho * grav * depth of water
where depth of water is refy - y-coordinate of segment
and refy is the y-coordinate of the water surface

*DEFINE_FUNCTION
10
float hpres(float t, float x, float y, float z, float x0, float y0, float
           z0)
{
    float fac, trise, refy, rho, grav;
    trise = 0.1; refy = 0.5; rho = 1000.; grav = 9.81;
    fac = 1.0;
    if(t<=trise) fac = t/trise;
    return fac*rho*grav*(refy-y);
}
*LOAD_SEGMENT_SET
1,10
```

Example 2 illustrates that a programming language resembling C can be used to define a function. Before a variable or function is used, its type must be declared. For example, variable t is real, so it has “float” preceding it in the function declaration. The braces indicate the beginning and end of the function being programmed. Semicolons must appear after each statement but several statements may appear on a single line. Please refer to a C programming guide for more detailed information.

***DEFINE_FUNCTION_TABULATED**

Purpose: Define a function of one variable using two columns of input data (in the manner of *DEFINE_CURVE) that can be referenced by a limited number of keyword options or by other functions defined via *DEFINE_FUNCTION. This command must appear in the keyword deck before the function it defines is used.

The TITLE option is not allowed with *DEFINE_FUNCTION_TABULATED.

Card 1	1	2	3	4	5	6	7	8
Variable	FID				HEADING			
Type	I				A70			

Card 2	1	2	3	4	5	6	7	8
Variable				FUNCTION				
Type				A80				

Point Cards. Put one pair of points per card (2E20.0). Add as many cards as necessary. Input is terminated when a keyword ("*") card is found.

Cards 3	1	2	3	4	5	6	7	8
Variable	A1		01					
Type	F		F					
Default	0.0		0.0					

VARIABLE**DESCRIPTION**

FID

Function ID. Functions, tables (see *DEFINE_TABLE), and load curves may not share common ID's. A unique number has to be defined.

HEADING

An optional descriptive heading.

*DEFINE

*DEFINE_FUNCTION_TABULATED

VARIABLE	DESCRIPTION
FUNCTION	Function name.
A1, A2, ...	Abscissa values.
O1, O2, ...	Ordinate (function) values.

Example:

```
*BOUNDARY_PRESCRIBED_MOTION_SET
$ function 300 prescribes z-acceleration of node set 1000
1000,3,1,300
*DEFINE_FUNCTION_TABULATED
201
tabfunc
0., 200
0.03, 2000.
1.0, 2000.
*DEFINE_FUNCTION
300
a(t)=tabfunc(t)*t
$ for t < 0.03, function 300 is equivalent to a(t)=(200. + 60000.*t)*t
$ for t >= 0.03, function 300 is equivalent to a(t)=(2000.)*t
```

***DEFINE_GROUND_MOTION**

Purpose: Define an earthquake ground motion history using ground motion records provided as load curves, for use in conjunction with *LOAD_SEISMIC_SSI for dynamic earthquake analysis including nonlinear soil-structure interaction.

Card 1	1	2	3	4	5	6	7	8
Variable	GMID	ALCID	VLCID					
Type	I	I	I					
Default	none	none	0					

VARIABLE	DESCRIPTION
GMID	Ground motion ID. A unique number has to be defined.
ALCID	Load curve ID of ground acceleration history.
VLCID	Load curve ID of ground velocity history.

Remarks:

- Earthquake Ground Motion Data.** Earthquake ground motion data is typically available either only as ground accelerations, or as a triple of ground accelerations, velocities and displacements. Usually, the velocities and the displacements are computed from the accelerations using specialized filtering and baseline correction techniques. Either input is accepted, with each quantity specified as a load curve. Only the acceleration and the velocity are required in the latter case; LS-DYNA does not require the ground displacement.
- Generated Velocity Load Curve.** If only the ground acceleration data is provided for a particular ground motion, LS-DYNA generates a corresponding load curve for the velocity by integrating the acceleration numerically. The generated load curves are printed out to the d3hsp file. It is up to the user to ensure that these generated load curves are satisfactory for the analysis.

*DEFINE

*DEFINE_HAZ_PROPERTIES

*DEFINE_HAZ_PROPERTIES

Purpose: To model the heat affected zone in a welded structure, the yield stress and failure strain are scaled in shell models as a function of their distance from spot welds and the nodes specified in *DEFINE_HAZ_TAILOR_WELDED_BLANK. *DEFINE_HAZ_PROPERTIES currently supports spot welds defined with *CONSTRAINED_SPR2 and *CONSTRAINED_INTERPOLATION_SPOTWELD. It also supports spot welds defined with beam or solid elements using *MAT_SPOTWELD. *CONSTRAINED_SPOTWELD is not currently supported.

Card 1	1	2	3	4	5	6	7	8
Variable	ID_HAZ	IOP	PID	PID_TYP				
Type	I	I	I	I				
Default	0	0	0	0				

Card 2	1	2	3	4	5	6	7	8
Variable	ISS	IFS	ISB	IFB	ISC	IFC	ISW	IFW
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

VARIABLE	DESCRIPTION
ID_HAZ	Property set ID. A unique ID number must be used.
IOP	Activate scaling flag: EQ.0: Scaling not activated. EQ.1: Scaling activated.
PID	Part or part set ID
PID_TYP	PID type: EQ.0: Part ID EQ.1: Part set ID

VARIABLE	DESCRIPTION
ISS	Curve ID for scaling the yield stress based on the distance to the closest solid element spot weld.
IFS	Curve ID for scaling the failure strain based on the distance to the closest solid element spot weld.
ISB	Curve ID for scaling the yield stress based on the distance to the closest beam element spot weld. Use a negative ID for curves normalized by the spot weld diameter as described in Remark 2 .
IFB	Curve ID for scaling the failure strain based on the distance to the closest beam element spot weld. Use a negative ID for curves normalized by the spot weld diameter as described in Remark 2 .
ISC	Curve ID for scaling the yield stress based on the distance to the closest constrained spot weld.
IFC	Curve ID for scaling the failure strain based on the distance to the closest constrained spot weld.
ISW	Curve ID for scaling the yield stress based on the distance to the closest tailor-welded blank node.
IFW	Curve ID for scaling the failure strain based on the distance to the tailor-welded blank node.

Remarks:

1. **Scaling.** The yield stress and failure strain are assumed to vary radially as a function of the distance of a point to its neighboring spot welds. Since larger spot welds may have a larger radius of influence, the smallest scale factor for the yield stress from all the neighboring spot welds is chosen to scale the yield stress at a particular point. The failure strain uses the scaling curve for the same weld.
2. **Weld diameter normalization.** Curve IDs may be input as negative values to indicate that they are normalized by the diameter of the spot weld to compensate for the effects of the spot weld size. When this option is used, the scale factor is calculated based on the distance divided by the spot weld diameter for the spot weld that is closest to the element.
3. **Distance measurement.** The distance from a spot weld (or node for the blank) is measured along the surface of the parts in the part set. This prevents the heat-softening effects of a weld from jumping across empty space.

4. **Material restrictions.** The HAZ capability only works with parts with materials using the STOCHASTIC option. It may optionally be simultaneously used with *DEFINE_STOCHASTIC_VARIATION to also account for the spatial variations in the material properties. See *DEFINE_STOCHASTIC_VARIATION for more details.
5. **Extra history variables.** The scale factors on yield stress and failure strain may be stored as extra history variables as follows:

Material Model	History Variable # for Scaling Factor on Yield Stress	History Variable # for Scaling Factor on Failure Strain
10	5	6
15	7	8
24	6	7
81	6	7
98	7	8
123 (shells only)	6	7
GISSMO	n/a	ND+20

When the HAZ capability is used in combination with *DEFINE_STOCHASTIC_VARIATION, the history variables shown in the table above represent the net scale factors due to HAZ and stochastic variation.

DEFINE_HAZ_TAILOR_WELDED_BLANK**DEFINE*****DEFINE_HAZ_TAILOR_WELDED_BLANK**

Purpose: Specify nodes of a line weld such as in a Tailor Welded Blank. The yield stress and failure strain of the shell elements in the heat affected zone (HAZ) of this weld are scaled according to *DEFINE_HAZ_PROPERTIES.

Card 1	1	2	3	4	5	6	7	8
Variable	IDTWB	IDNS	IDP	IPFLAG	IMONFLAG			
Type	I	I	I	I	I			
Default	0	0	0	0	0			

VARIABLE	DESCRIPTION
IDTWB	Tailor Welded Blank ID
IDNS	Node Set ID defining the location of the line weld.
IDP	Part or part set ID. Applies to all HAZ parts if IDP = 0 (default).
IPFLAG	IPFLAG type: EQ.0: part ID (default) EQ.1: part set ID
IMONFLAG	Monotonicity flag for load curves ISW and IFW on *DEFINE_HAZ_PROPERTIES: EQ.0: ISW and IFW increase monotonically. EQ.1: ISW and IFW are allowed to be arbitrary load curves.

*DEFINE

*DEFINE_HEX_SPOTWELD_ASSEMBLY

*DEFINE_HEX_SPOTWELD_ASSEMBLY_{OPTION}

Purpose: Define a list of hexahedral elements that make up a single spot weld for computing the force and moment resultants that are written into the swforc output file. A maximum of 16 elements may be used to define an assembly representing a single spot weld. See [Figure 17-55](#). This table of element IDs is generated automatically when beam elements are converted to solid elements. See the input parameter RPBHX associated with the keyword *CONTROL_SPOTWELD_BEAM.

Available options for this command are:

<BLANK>

N

For the <BLANK> option, all solid elements specified on Card 2 make up the spot weld and no additional card is read. For the N option, N is an integer representing the total number of solid elements making up the spot weld. If N is greater than 8, the additional card beyond Card 2 is read. N may not exceed 16.

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

Card 2	1	2	3	4	5	6	7	8
Variable	EID1	EID2	EID3	EID4	EID5	EID6	EID7	EID8
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

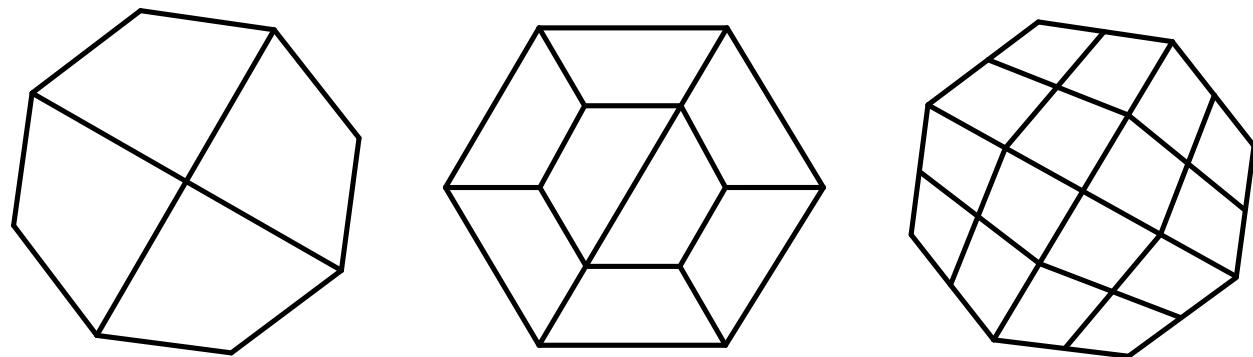


Figure 17-55. Illustration of four, eight, and sixteen element assemblies of solid hexahedron elements forming a single spot weld.

Additional card for $N > 8$.

Optional	1	2	3	4	5	6	7	8
Variable	EID9	EID10	EID11	EID12	EID13	EID14	EID15	EID16
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

VARIABLE
DESCRIPTION

ID_SW

Spot weld ID. A unique ID number must be used.

EID n

Element ID n for up to 16 solid hexahedral elements.

Remarks:

The elements comprising a spot weld assembly may share a part ID (PID) with elements in other spot weld assemblies defined using *DEFINE_HEX_SPOTWELD_ASSEMBLY but may not share a PID or even a material ID (MID) with elements that are not included in a *DEFINE_HEX_SPOTWELD_ASSEMBLY.

*DEFINE

*DEFINE_ISPG_TO_SURFACE_COUPLING

*DEFINE_ISPG_TO_SURFACE_COUPLING

Purpose: Define a coupling interface between fluid particles modeled with incompressible smoothed particle Galerkin (ISPG) and a surface. The coupling can be tied except on the edges, entirely tied, or free slip.

NOTE: This keyword is only available for ISPG parts with sections specified with *SECTION_ISPG.

Card 1	1	2	3	4	5	6	7	8
Variable	FP	SURF	FPTYPE	SURFTYPE	CPL_ID			
Type	I	I	I	I	I			
Default	none	none	none	none	none			

Card 2	1	2	3	4	5	6	7	8
Variable	SBC	SCA	ACA	RCA	VCA	SFPN	THK	
Type	I	F	F	F	F	F	F	
Default	0	none	SCA	SCA	1.0	1.0	0.5	

Card 3	1	2	3	4	5	6	7	8
Variable	HTC							
Type	F							
Default	2.0E-5							

VARIABLE

DESCRIPTION

FP

Part or part set ID for the fluid particles

VARIABLE	DESCRIPTION
SURF	Segment set ID specifying the surface. Currently, the segment set should be generated from 8-node hexahedral elements.
FPTYPE	Type for FP: EQ.0: Part set ID EQ.1: Part ID
SURFTYPE	Type for SURF: EQ.0: Segment set ID
CPL_ID	ID of the coupling
SBC	Boundary condition type for the contact edges (see Figure 17-56) and contact treatment for the non-contact-edge ISPG nodes: EQ.0: Contact edges can move on the structure's surface, while non-contact-edge ISPG nodes in contact with the structure are tied to the structure, modeling a non-slip boundary condition. EQ.1: All ISPG nodes in contact with the structure, including those on the contact edges, are tied to the structure. EQ.2: A free slip condition applies to all ISPG nodes in contact with the structure, including those on the contact edges.
SCA	Static (equilibrium) contact angle in radians. See Remark 1 .
ACA	Advancing contact angle in radians. If set to zero, then the value of SCA is used. See Remark 1 .
RCA	Receding contact angle in radians. If set to zero, then the value of SCA is used. See Remark 1 .
VCA	Velocity of the fluid contact edge for measurement of the contact angles. See Remark 1 .
SFPN	Scale factor for the thickness of the solid elements coupled with the ISPG fluid. This parameter can increase the contact detection thickness if the length of the solid element along the contact normal direction is much less than the length in the in-plane direction.
THK	Scale factor on the average ISPG element edge length for contact detection

*DEFINE

*DEFINE_ISPG_TO_SURFACE_COUPLING

VARIABLE	DESCRIPTION
HTC	Heat transfer coefficient for the thermal boundary condition between the ISPG nodes and the structural surface if performing a coupled thermal-flow analysis. The default value is 2.0×10^5 kW/(mm ² K) with its base units assumed to be kg, mm, ms, and K.

(a)

(b)

Figure 17-56. Example showing contact between an ISPG fluid and structural segments. When the fluid moves as shown in (a), the fluid moves “forward” or advances along the structural segment. If the fluid moves as shown in (b), the fluid moves “backward” or recedes along the structure.

Remarks:

1. **Adhesion force and contact angles.** For $SBC \neq 1$, the wall adhesion force, $F_{adhesion}$, acts on the contact edge of the ISPG fluid. $F_{adhesion}$ is towards the structure's interior, and the contact angle determines the direction of the $F_{adhesion}$, as illustrated in [Figure 17-56](#).

The contact angle depends on the ISPG fluid edge velocity relative to the structural segment. When the ISPG fluid moves forward, as shown in the portion of [Figure 17-56](#) labeled (a), the contact angle is:

$$\begin{cases} ACA & \text{if } |v_{edge}| > VCA \\ ACA + \frac{(SCA - ACA)(VCA - |v_{edge}|)}{VCA} & \text{if } |v_{edge}| \leq VCA \end{cases}$$

When it moves backward, as illustrated by (b) in [Figure 17-56](#), the contact angle is:

$$\begin{cases} RCA + \frac{(SCA - RCA)(VCA - |v_{edge}|)}{VCA} & \text{if } |v_{edge}| \leq VCA \\ RCA & \text{if } |v_{edge}| > VCA \end{cases}$$

***DEFINE_LANCE_SEED_POINT_COORDINATES**

Purpose: Define seed points that determine the portion of a part(s) that is not removed (not scraps) during the trimming portion of lancing. This keyword is to be used in conjunction with *ELEMENT_LANCING. See *ELEMENT_LANCING.

Card 1	1	2	3	4	5	6	7	8
Variable	NSEED	X1	Y1	Z1	X2	Y2	Z2	
Type	I	F	F	F	F	F	F	
Default	none	0.0	0.0	0.0	0.0	0.0	0.0	

VARIABLE	DESCRIPTION
NSEED	Number of seed points. Maximum value of “2” is allowed.
X1, Y1, Z1	Location coordinates of seed point #1.
X2, Y2, Z2	Location coordinates of seed point #2.

Remarks:

- Trimming Restrictions.** This keyword will remove all scraps during or after lancing, depending on how the parameter AT is defined in *ELEMENT_LANCING. Lancing curves must form a closed loop, meaning the first and last point coordinates must be coincident. Scraps are the portions that are exclusive of the portions whose seed points are defined by this keyword.
- Example.** The following input defines two sets of seed point coordinates, where a double-attached part may be lanced and trimmed:

```
*DEFINE_LANCE_SEED_POINT_COORDINATES
$   NSEED      X1      Y1      Z1      X2      Y2      Z2
    2       -289.4    98.13   2354.679   -889.4    91.13   255.679
```

Revision Information

This feature is available starting with LS-DYNA Revision 107262.

*DEFINE

*DEFINE_MATERIAL_HISTORIES

*DEFINE_MATERIAL_HISTORIES_{OPTION}

The available options include:

<BLANK>

NAMES

Purpose: Control the content of the history variables in the d3plot and d3part databases. This feature is supported for solid, shell, and integrated beam elements. See [Motivation](#) below for the rationale behind this keyword and [How this Keyword Works](#) for a description of how the history variables are output. Note that all mentions of d3plot also apply to d3part which is excluded for clarity.

The NAMES keyword option enables giving each specified history variable a corresponding name. These names are reflected in d3hsp and a post-processing file containing the history variable names if HISNOUT > 0 on [*CONTROL_OUTPUT](#).

NOTE: Whenever the input file includes this keyword, the effective plastic strain output is zero for all materials for which there is no plastic strain (see [Remark 3](#) for more details).

Card Summary:

Card Sets. Define as many instantiations of Card 1 or sets of Card 1 and Card 2 for the NAMES keyword option as needed to define the extra history variables. This input ends at the next keyword ("*") card. This keyword can be defined without any data cards; see [Remark 3](#) below for details of when this should be used.

Card 1. Include as many of this card or sets of this card with Card 2 for the names keyword option as needed.

LABEL	A1	A2	A3	A4

Card 2. Include this card in sets with Card 1 when the NAMES keyword option is used.

NAME	

Data Card Definitions:

History Variable Card. Define as many cards as needed to define the extra history variables. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable		LABEL			A1	A2	A3	A4
Type		A40			F	F	F	F
Default		none			0.0	0.0	0.0	0.0

VARIABLE	DESCRIPTION
LABEL	String identifying history variable type. Labels are <i>case-sensitive</i> . See section LABEL below and Remark 4 . By prefixing the label with (capital letter) 'X', the current history variable will not be printed to the binary database. See Example 4 .
An	Attributes. See section LABEL below.

User-Defined Name Card. Additional card for the NAMES keyword option. Define this card in sets with Card 1. See [Example 2](#).

Card 2	1	2	3	4	5	6	7	8
Variable				NAME				
Type				A70				

VARIABLE	DESCRIPTION
NAME	User-defined name for history variable

LABEL:

The permitted LABELs (*case-sensitive*) are

Instability	A number between 0 and 1 that indicates how close an element or integration point is to failure or to initiate damage, whatever happens first (see Remark 1). Attributes apply to the DIEM
-------------	---

damage model (see *MAT_ADD_DAMAGE_DIEM) and add erosion model (see *MAT_ADD_EROSION) only. In those cases, the following apply:

A1 value	Description
-12	Instability value for FAILTM on *MAT_ADD_EROSION. FAILTM is assumed nonzero.
-10	Instability value for SIGTH and IMPULSE on *MAT_ADD_EROSION. SIGTH and IMPULSE are assumed nonzero.
-9	Instability value for EPSSH on *MAT_ADD_EROSION. EPSSH is assumed nonzero.
-8	Instability value for MXEPS on *MAT_ADD_EROSION. MXEPS is assumed nonzero.
-7	Instability value for SIGVM on *MAT_ADD_EROSION. SIGVM is assumed nonzero.
-6	Instability value for SIGP1 on *MAT_ADD_EROSION. SIGP1 is assumed nonzero.
-5	Instability value for MNPRES on *MAT_ADD_EROSION. MNPRES is assumed nonzero.
-4	Instability value for VOLEPS on *MAT_ADD_EROSION. VOLEPS is assumed nonzero.
-3	Instability value for EFFEPPS on *MAT_ADD_EROSION. EFFEPPS is assumed nonzero.
-2	Instability value for MNEPS on *MAT_ADD_EROSION. MNEPS is assumed nonzero.
-1	Instability value for MXPRES on *MAT_ADD_EROSION. MXPRES is assumed nonzero.
1	Max instability value in any integration point and any criterion for DIEM
2	Location (isoparametric z-coordinate between -1 and 1) of max instability value for DIEM
3	Criterion attaining max instability value for DIEM (integer value)
4	Stress triaxiality η for evaluating instability criterion value for the ductile (DITYP = 0) criterion in DIEM

5	Shear influence θ for evaluating instability criterion value for the shear (DITYP = 1) criterion in DIEM
6	Ratio of principal strain rates α for evaluating instability criterion value for the MSFLD (DITYP = 2) criterion in DIEM
7	Ratio of principal strain rates α for evaluating instability criterion value for the FLD (DITYP = 3) criterion in DIEM
8	Stress state parameter β for evaluating instability criterion value for the weighted ductile (DITYP = 4) criterion in DIEM

Damage

A number between 0 and 1 indicating the damage level of the element or integration point (see [Remark 1](#)). Attributes apply to the GISSMO damage model (see [*MAT_ADD_DAMAGE_GISSMO](#)) only. In those cases, the following apply:

A1 value	Description
0	GISSMO's effective damage value \tilde{D} (ND+18)
1	GISSMO's damage value D (ND)

Plastic Strain Rate

Effective plastic strain rate, calculated generically for “all” plastic materials from the evolution of effective plastic strain. No attributes apply to this label.

Effective Stress

Effective, or equivalent, stress. If not specifically stated in the material section for the material in question, this will be the von Mises stress. For selected anisotropic materials (such as the Barlat models), it is the stress value that is conjugate to the effective plastic strain rate with respect to energy density rate.

Effective Tresca Stress

Effective, or equivalent, stress according to Tresca (maximum shear stress):

$$\sigma_e^T = \max[|\sigma_1 - \sigma_2|, |\sigma_2 - \sigma_3|, |\sigma_3 - \sigma_1|] ,$$

where σ_1 , σ_2 and σ_3 are the principal stresses.

Effective Strain

Effective, or equivalent, (von Mises) strain:

$$\epsilon_e = \sqrt{\frac{2}{3}(\epsilon_{dev,11}^2 + \epsilon_{dev,22}^2 + \epsilon_{dev,33}^2 + 2\epsilon_{12}^2 + 2\epsilon_{23}^2 + 2\epsilon_{31}^2)} ,$$

where ε_{ij} are the standard LS-DYNA output strains (similar to logarithmic strain, see Chapter 22.10 in the Theory Manual) and ε_{dev} denotes the deviatoric component.

Note, using this label activates (integration point) strain output in ***DATABASE_EXTENT_BINARY** (i.e. STRFLG(L) = 1 and NINTSLD = 8).

Plastic Energy Density Plastic work per unit volume. It is only nonzero for materials deemed as plastic materials. This is evaluated as $\int_0^t \sigma \dot{\varepsilon}_{\text{eff}}^p$, where σ is the von Mises effective stress and $\dot{\varepsilon}_{\text{eff}}^p$ is the work equivalent plastic strain. Thus, the result will not be entirely correct for anisotropic materials. Upon request, we will improve the calculation for the asked for anisotropic materials.

Effective Creep Strain Effective creep strain for material models with creep. In particular, it will provide the effective creep strain for *MAT_ADD_INELASTICITY if creep is one of the selected inelasticity laws; see this keyword for more details.

History Fetch a history variable at a known place for a given material type, element type and part set (see [Example 1](#)). The following attributes apply:

Attribute	Description
A1	Mandatory (no defaults) GT.0: History variable location LT.0: Load curve ID = (-A1); see below for a complete description
A2	Material type. By default, it applies to all materials.
A3	Element type: 0 for all element types, 1 for solids, 2 for shells and 3 for beams
A4	Part set. History will only be fetched from the parts in the given part set. The default is all parts in the model.

When A1 < 0, then |A1| refers to a [*DEFINE_CURVE](#). In this curve definition you specify an ‘operation’ that you would like to perform over a certain list of history variables within an element.

Operation	Description
1	Maximum value
2	Layer where the maximum value occurs
3	History variable that attains the max value
4	Sum the history variables

Only the ordinate values of the load curve are used, but we suggest defining the abscissa value using increasing integer numbers starting with 0. Furthermore, you should use DATTYP = 6 in the curve definition to make sure that the data is not transformed in any way. Here is an example:

```
*DEFINE_CURVE
ID, 0, 1., 1., 0., 0., 6, 0
0, 1 (operation)
1, 12 (location of 1st history variable)
2, 17 (location of 2nd history variable)
3, 22 (location of 3rd history variable)
```

In this particular case, you will receive the maximum value of the history variables 12, 17 and 22 at any integration point within one element.

Operator

Perform an operation on a given history variable or quantity. The following attributes apply:

Attribute	Description
A1	Type of operator: EQ.0: Maximum value over time EQ.1: Point in time when maximum value over time was attained. EQ.2: Minimum value over time EQ.3: Point in time when minimum value over time was attained. EQ.5: Absolute value, A2 EQ.6: Addition, A2 + A3 EQ.7: Subtraction, A2 - A3 EQ.8: Multiplication, A2 × A3 EQ.9: Division, A2 / A3 (assuming A3 > 0) EQ.10: Exponentiation, A2 ^{A3} EQ.11: Maximum, max(A2, A3) EQ.12: Minimum, min(A2, A3)

	EQ.13: Natural logarithm, $\ln A2 $
A2 / A3	First and second argument to operator: EQ.-16: Real number given by A4 EQ.-9: x -stress EQ.-8: y -stress EQ.-7: z -stress EQ.-6: xy -stress EQ.-5: yz -stress EQ.-4: zx -stress EQ.-3: Mid principal stress EQ.-2: Minor principal stress EQ.-1: Major principal stress EQ.0: Effective plastic strain GT.0: History variable # in list specified here
A4	Parameter for operator A1: If A2 or A3 are EQ.-16 then Real number (see Example 4) Else If 0.LE.A1.LE.3 then GT.0: Time at which min/max will be reset LT.0: $ A4 $ is ID to load curve whose ordinate contains time points when min/max will be reset. Even though only the ordinate values are used, DATTYP = 6 is recommended. Else LT.0: Shell stress will be rotated to global coordinates. Endif Endif

The stress components used as arguments (A2 and A3) are for solid elements in global coordinates and for shell elements in element-local (co-rotated) coordinates. However, by setting A4 to a negative value, shell element stresses will be rotated to global coordinates.

See [Examples 3](#) and [4](#).

Principal Stress Range Compute the maximum (principal) stress amplitude. This feature is a scanning tool intended for fatigue analysis. Currently, only shell elements are supported. See [Remark 5](#) for details about how it is determined.

The following attributes apply:

Attribute	Description
A1	GT.0: Reset time LT.0: A1 is ID to load curve whose ordinate contains time points when min/max will be reset. Even though only the ordinate values are used, DAT-TYP = 6 is recommended.
A2	Number of buckets (NBKTS) each 90° sector is divided into. Default value is 65 buckets.
A3	(not used)
A4	Part set. Stress range will only be computed in the parts in the given part set. The default is all parts in the model.

Max Stress Range

As label *Principal Stress Range* above but not specifically considering the principal stresses; see [Remark 5](#). The drawback of this approach is that close to twice the amount of storage is needed. Therefore, the default number of buckets (NBKTS) is 45.

The same attributes apply as for label *Principal Stress Range*.

Motivation:

Material models in LS-DYNA have history variables that are specific to the constitutive model being used. For most materials, 6 are reserved for the Cauchy stress components and 1 for the effective plastic strain, but many models have more than that. The history variables may include physical quantities like material damage, material phase compositions, strain energy density and strain rate, as well as nonphysical quantities like material direction cosines and scale factors.

By using NEIPS, NEIPB and NEIPH on [*DATABASE_EXTENT_BINARY](#), these extra history variables can be exported to the d3plot database *in the order that they are stored*, and in LS-PrePost the variables may then be plotted (Hist button) or fringed (Misc menu). This approach has a few drawbacks. You must, for instance, have knowledge of the storage location of a certain history variable for a given material model and element type.

While this information can be retrieved either in the LS-DYNA manual, on LS-DYNA support sites or in LS-PrePost itself, it is not always convenient.

Furthermore, the same physical quantity may be stored in different locations for different materials and different element types, meaning that history variable #1 will correspond to different things in different parts which complicates post-processing of large models. You may also be interested in a certain material specific quantity that is not necessarily stored as a history variable; this quantity is not retrievable using this approach. Finally, if the history variable of interest happens to be stored in a bad location, that is, among the last ones in a long list, it would be necessary to set NEIPS, NEIPB and/or NEIPH large enough to access this variable in LS-PrePost. This could result in unnecessarily large binary plot files.

How this Keyword Works:

This keyword attempts to organize the extra history variables with the goal of obtaining an output that is reasonably small and easy to interpret. The input is very simple: use the keyword *DEFINE_MATERIAL_HISTORIES in the keyword input deck, followed by lines that specify the history variables of interest using predetermined labels and attributes. *NEIPS, NEIPB and NEIPH on *DATABASE_EXTENT_BINARY are then overridden by the number of history variables, that is, number of lines, requested on this card.* As an example:

```
*DEFINE_MATERIAL_HISTORIES
Instability
Damage
```

means that two extra history variables are output to the d3plot database, so NEIPS, NEIPB and NEIPH are internally set to 2 regardless of the user input. History variable #1 corresponds to an instability measure (between 0 and 1) and history variable #2 corresponds to a material damage (between 0 and 1). Thus, the history variables are output in the order they are listed. If there are several instances of this keyword in an input deck, the order of the history variables follows the order that the cards are read by the keyword reader.

In the d3hsp file, search for the string “Material History List” to find the complete list. For a material that does not store or calculate an “instability” or “damage” history variable, the output is zero and thus the output is not cluttered by unwanted data. Note that this keyword does not necessarily require that the history variable be stored, as long as it can be calculated when LS-DYNA outputs a plot state (see [Remark 7](#) regarding storage). Thus, you can possibly request quantities that are not available by just using NEIPS, NEIPB and/or NEIPH on [*DATABASE_EXTENT_BINARY](#).

Remarks:

1. **Damage and instability.** For large models with many different parts and materials, the *Instability* and *Damage* variables should provide a comprehensive overview and understanding of the critical areas in terms of failure that otherwise may not be assessable. Note that *Damage* can only be nonzero after *Instability* reaches 1, and thereafter the integration point fails when *Damage* reaches 1. If the material has no damage evolution law (which is the case in many of the *MAT_ADD_EROSION features for which failure of the integration point occurs immediately upon reaching the failure criterion without any stress softening), then in general *Damage* will jump from 0 to 1 exactly when *Instability* reaches 1.
2. **History variable requests from *DATABASE_EXTENT_BINARY.** As discussed in [How this Keyword Works](#), NEIPS, NEIPB, and NEIPH set in *DATABASE_EXTENT_BINARY are ignored when the input deck includes *DEFINE_MATERIAL_HISTORIES. The number of extra history variables output to d3plot is determined by how many instantiations of Card 1 are included in the input deck. For instance, if only instability and damage are specified with this keyword, then two extra history variables are output.
3. **Effective plastic strain.** Whenever *DEFINE_MATERIAL_HISTORIES is present in the input file, the effective plastic strain output will be zero for all materials for which there is no plastic strain. Thus, this slot in the d3plot database is in this case *not* taken over by something else, but rather occupied by zero data. For instance, for *MAT_ELASTIC, it will be a null field while it will be the effective plastic strain field for *MAT_PIECEWISE_LINEAR_PLASTICITY. For a part that uses *MAT_ADD_INELASTICITY with a plastic model, the effective plastic strain will be the one pertaining to this inelasticity law. Thus, *DEFINE_MATERIAL_HISTORIES may be used without any cards if this behavior is desired for the effective plastic strain.
4. **Requestable history variables types.** At the end of the remarks for a material model in Volume II, the history variable types (other than *History*) available are listed in a table similar to the one below. *History* is available for all materials, but retrievable ones depend on the model. Also, whether mentioned in such a table or not, *Plastic Strain Rate* and *Plastic Energy Density* available for any material model that calculates plastic strain as the 7th standard history variable.

<i>*DEFINE_MATERIAL_HISTORIES Properties</i>					
Label	Attributes				Description
Instability	-	-	-	-	Failure indicator $\epsilon_{\text{eff}}^p / \epsilon_{\text{fail}}^p$, see FAIL
Plastic Strain Rate	-	-	-	-	Effective plastic strain rate $\dot{\epsilon}_{\text{eff}}^p$

<i>*DEFINE_MATERIAL_HISTORIES Properties</i>		
Label	Attributes	Description
Plastic Energy Density	- - - - -	Plastic energy density $\int_0^t \sigma \dot{\varepsilon}_{\text{eff}}^p$

Label in the table states what the string *LABEL* on ***DEFINE_MATERIAL_HISTORIES** must be, *a1* to *a4* will list attributes *A1* to *A4* if necessary, and *Description* will be a short description of what is output with this option, including possible restrictions.

5. **Calculating principal stress range.** Assuming plane stress ($\sigma_z = 0$), the normal stress at an angle φ to the local element *x*-axis is

$$\sigma(\varphi) = \frac{\sigma_x + \sigma_y}{2} + \frac{\sigma_x - \sigma_y}{2} \cos 2\varphi + \tau_{xy} \sin 2\varphi.$$

Specifically, the major principal stress and its direction are

$$\sigma_1 = \frac{\sigma_x + \sigma_y}{2} \pm \sqrt{\left(\frac{\sigma_x - \sigma_y}{2}\right)^2 + \tau_{xy}^2}$$

$$\varphi_1 = \frac{1}{2} \tan^{-1} \frac{2\tau_{xy}}{\sigma_x - \sigma_y}$$

Using the equations above, the range of angles $\varphi \in [-90^\circ, 90^\circ]$ is divided into $2 \times \text{NBKTS}$ buckets with each bucket covering $90^\circ/\text{NBKTS}$ degrees. In each bucket the minimum stress over time is stored. At the same time the over-time maximum major principal stress and its direction is stored. The stress range is then computed as the difference between the major principal stress and the minimum stress in the corresponding direction (i.e., bucket). Output to the binary database is the over-time maximum stress range.

The computed principal stress range is exact in the case when the angle of the minimum minor principal stress is the same as the angle of the maximum major principal stress. In most other cases the computed stress range is an overestimate. It is estimated that the worst overestimate would be close to $2\cos(90^\circ - 90^\circ/\text{NBKTS}) \times 100\%$ of the minimum minor principal stress in size.

Please note that besides the maximum stress range, $2 \times \text{NBKTS} + 2$ values will be stored for each integration point. Therefore, we recommend limiting the number of elements for which this label is used.

The alternative label *Max Stress Range* does not consider the direction of the major principal stress. Instead, in each bucket the stress range is computed as the difference of the over-time maximum and minimum normal stress. Output to the binary database is the over-time maximum stress range of all buckets. For

this label, besides the maximum stress range, $4 \times \text{NBKTS}$ values will be stored for each integration point.

6. **Future developments and feature requests.** Further development of this keyword will mainly be driven by customer requests submitted to LstSuggestions@ansys.com. Currently only solid, shell, and integrated beam elements are supported for the binary d3plot format. Future developments may include adding output for thick shells to d3plot and supporting ASCII/binout.
7. **Storage allocation.** In many cases (e.g., *Instability*, *History*, *Effective Stress*), no additional storage (by the element or material) beyond the already occupied slots in the internal data array is needed. However, some requested quantities on this card will require additional storage. In most cases only one slot, but in other cases more. For example, requesting the *Operator* ‘Point in time when maximum value over time was attained’ ($A1 = 1$) will require two slots: one containing the maximum value and one for the time it occurred. Note that only one value (the time) will be output to the d3plot database. Below is a list of the *additional* data storage required for each label.

Label	Additional storage slots
<i>Instability</i>	0
<i>Damage</i>	0
<i>Plastic Strain Rate</i>	1
<i>Effective Stress</i>	0
<i>Effective Tresca Stress</i>	0
<i>Plastic Energy Density</i>	1
<i>Effective Creep Strain</i>	0
<i>History</i>	1
<i>Operator</i>	2 if $A1 = 1$ or 3 1 otherwise
<i>Principal Stress Range</i>	$3 + \text{NBKTS} \times 2$
<i>Max Stress Range</i>	$1 + \text{NBKTS} \times 4$

Examples:

1. **History label.** The History label is for when you know where history variables of interest are stored and want to use this to compress or simplify the output. For example, the following input

```
*DEFINE_MATERIAL_HISTORIES
History,4,272,1,23
History,1,81
*SET_PART_LIST
23
2,3
```

makes a list of two history variables in the output. History variable #1 fetches the 4th history variable and outputs it only for the RHT concrete model (material 272) solid elements in parts 2 and 3. History variable #2 fetches the 1st history variable and outputs it only for the plasticity with damage model (material 81) but for any element and part. Both of these requested variables happen to be the damage in the respective materials, so an alternative to do something similar would be to use

```
*DEFINE_MATERIAL_HISTORIES
Damage
```

for which the damage for all materials will be displayed in history variable #1.

2. **History variable names.** The NAMES keyword option allows you to give the history variables labels in d3hsp and the post-processing file (if HISNOUT > 0 on [*CONTROL_OUTPUT](#)). For example, the following

```
*DEFINE_MATERIAL_HISTORIES_NAMES
History,1,24
Effective strain rate
History,4,47
my_own_history
```

gives the first history variable corresponding to history variable 1 of *MAT_024 the label “Effective strain rate” and the second history variable corresponding to the fourth history variable of user material 47 the label “my_own_history”.

3. **Operator label.** The Operator label is for performing specific operations on stress or a history variable listed in *DEFINE_MATERIAL_HISTORIES. For example,

```
*DEFINE_MATERIAL_HISTORIES
Effective Stress
Operator,0,1,0,0.1
Operator,1,1,0,0.1
Operator,0,-1,0,-99
Operator,1,-1,0,-99
Operator,2,-2
Operator,3,-2
*DEFINE_CURVE
-99
```

```
0.0,0.1  
0.0,0.3  
0.0,0.7
```

leads to the following history variables in d3plot:

History Variable #	Description
1	Effective stress
2	Maximum value over time of effective stress, reset at time 0.1.
3	Time when maximum value in time of effective stress was attained, reset at time 0.1.
4	Maximum value in time of major principal stress, reset at times 0.1, 0.3, and 0.7.
5	Time when maximum value in time of major principal stress was attained, reset at times 0.1, 0.3, and 0.7.
6	Minimum value in time of minor principal stress
7	Time when minimum value in time of minor principal stress was attained.

4. **Operator label for creating expressions.** The Operator label can be used to create user-defined expressions. For example,

```
*DEFINE_MATERIAL_HISTORIES  
$ 1 (1):  
Effective Tresca Stress  
$ 2 (-): s1-s2  
XOperator,7,-1,-3  
$ 3 (-): s2-s3  
XOperator,7,-3,-2  
$ 4 (-): s3-s1  
XOperator,7,-2,-1  
$ 5 (-): abs(s1-s2)  
XOperator,5,2  
$ 6 (-): abs(s2-s3)  
XOperator,5,3  
$ 7 (-): abs(s3-s1)  
XOperator,5,4  
$ 8 (-): max(abs(s1-s2),abs(s2-s3))  
XOperator,11,5,6  
$ 9 (2): max(abs(s1-s2),abs(s2-s3),abs(s3-s1))  
Operator,11,7,8  
$ 10 (-): hsv 1 + hsv 9  
XOperator,6,1,9
```

***DEFINE**

***DEFINE_MATERIAL_HISTORIES**

```
$ 11 (3): hsv 10 * 0.5  
Operator,8,10,-16,0.5
```

leads to the following history variables in d3plot:

History Variable #	Description
1	Effective Tresca stress
2	Effective Tresca stress (computed using operators)
3	Mean value of Tresca expressions above

DEFINE_MULTI_DRAWBEADS_IGES**DEFINE*****DEFINE_MULTI_DRAWBEADS_IGES**

Purpose: Simplify the definition and creation of draw beads, which previously required the use of many keywords.

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

Card 2	1	2	3	4	5	6	7	8
Variable	DBID	VID	PID	BLKID	NCUR			
Type	I	I	I	I	I			
Default	none	none	1	1	none			

IGES Curve ID Cards. For multiple draw bead curves include as many cards as necessary. Input is terminated at the next keyword ("*") card.

Card 3	1	2	3	4	5	6	7	8
Variable	CRVID	BFORCE						
Type	I	F						
Default	none	0.0						

VARIABLE	DESCRIPTION
FILENAME	IGES file that has the draw bead curve segment definitions
DBID	Draw bead set ID, which may consist of many draw bead segments.

DEFINE**DEFINE_MULTI_DRAWBEADS_IGES**

VARIABLE	DESCRIPTION
VID	Vector ID, as defined by *DEFINE_VECTOR. This vector is used to project the supplied curves to the rigid tool, defined by PID.
PID	Part ID of a rigid tool to which the curves are projected and attached.
BLKID	Part ID of the sheet blank with which created draw beads will contact.
NCUR	Number of draw bead curve segments (in the IGES file defined by FILENAME) to be defined.
CVRID	IGES curve ID for each segment.
BFORCE	Draw bead force for each segment.

Revision information:

This feature is available in LS-DYNA R5 Revision 62840 and later releases.

DEFINE_MULTI_SHEET_CONNECTORS**DEFINE*****DEFINE_MULTI_SHEET_CONNECTORS**

Purpose: Define a multi-sheet connection with up to four shell element sheets. The individual sheets can be joined with up to three connection elements. Currently, for the connection elements, only single hexahedron elements with *MAT_SPOTWELD_DAIMLER-CHRYSLER (*MAT_100_DA) are supported. Also, PRUL must be ≥ 2 in *DEFINE_CONNECTION_PROPERTIES. In addition to the standard definition of *MAT_100_DA between two sheets, it is possible with this keyword to define the material behavior of the joining elements as functions of the geometric and material properties of all sheets involved.

Card Summary:

Card Sets. Include as many sets of the following cards as desired. This input ends with the next keyword ("*") card.

Card 1. This card is required.

ID	ITYP	NSHEETS					
----	------	---------	--	--	--	--	--

Card 2. This card is required.

PID1	JNT12	PID2	JNT23	PID3	JNT34	PID4	
------	-------	------	-------	------	-------	------	--

Card 3. This card is required.

PARM1	PARM2	PARM3	PARM4	PARM5	PARM6	PARM7	PARM8
-------	-------	-------	-------	-------	-------	-------	-------

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	ID	ITYP	NSHEETS					
Type	I	I	I					
Default	none	1	none					

VARIABLE**DESCRIPTION**

ID Multi sheet connector ID

DEFINE**DEFINE_MULTI_SHEET_CONNECTORS**

VARIABLE		DESCRIPTION						
ITYP		Material model of joint elements: EQ.1: *MAT_SPOTWELD_DAIMLERCHRYSLER.						
NSHEETS		Number of sheets connected with this multi sheet connector						
Card 2	1	2	3	4	5	6	7	8
Variable	PID1	JNT12	PID2	JNT23	PID3	JNT34	PID4	
Type	I	I	I	I	I	I	I	
Default	none	none	none	optional	optional	optional	optional	

VARIABLE		DESCRIPTION						
PID1		Part ID of sheet number one						
JNT12		ID of joining element between sheet one and two						
PID2		Part ID of sheet number two						
JNT23		ID of joining element between sheet two and three (optional)						
PID3		Part ID of sheet number three (optional)						
JNT34		ID of joining element between sheet three and four (optional)						
PID4		Part ID of sheet number four (optional)						

Card 3	1	2	3	4	5	6	7	8
Variable	PARM1	PARM2	PARM3	PARM4	PARM5	PARM6	PARM7	PARM8
Type	F	F	F	F	F	F	F	F
Default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

VARIABLE	DESCRIPTION
PARMi	Set of user parameters additionally available in *DEFINE_FUNCTION. See Remark 1 .

Remarks:

1. **Failure Rule from *DEFINE_FUNCTION.** The elements referenced through JNT12, JNT23 and JNT34 should be hexahedral solid elements with material *MAT_100_DA and PRUL must be ≥ 2 on *DEFINE_CONNECTION_PROPERTIES. Therefore, the following eleven variables in the respective *DEFINE_CONNECTION_PROPERTIES must be defined as function IDs: DSIGY, DETAN, DDGPR, DSN, DSB, DSS, DEXSN, DEXSB, DEXSS, DGFAD, and DSCLMR.

These functions depend on:

$(t1, t2, t3, t4)$ = thicknesses of the sheets
 $(sy1, sy2, sy3, sy4)$ = initial yield stresses at plastic strain
 $(sm1, sm2, sm3, sm4)$ = maximum engineering yield stresses
 r = strain rate of joining element
 a = area of joining element
 s = number of sheets
 n = internal number of joining element (1, 2, or 3)
 $(p1, p2, p3, p4, p5, p6, p7, p8)$ = parameters defined in card 3
 fn = normal term in failure function
 fb = bending term in failure function
 fs = shear term in failure function
 $(ym1, ym2, ym3, ym4)$ = Young's moduli of the sheets

For DSIGY = 100, such a function could look like:

```

*DEFINE_FUNCTION
 100
func(t1, t2, t3, t4, sy1, sy2, sy3, sy4, sm1, sm2, sm3, sm4, r, a, s, n, p1, p2, p3,
p4, p5, p6, p7, p8, fn, fb, fs, ym1, ym2, ym3, ym4)=0.5*(sy1+sy2)
  
```

All the listed arguments in their correct order must be included in the argument list, even if you are not using 4 sheets. Note that this argument list is longer than the one listed in the remarks for the manual page of *DEFINE_CONNECTION_PROPERTIES. Also, PRUL = 2 and 3 are the same for this keyword because the order of the sheets is determined by this keyword. Since material parameters must be identified from the weld partners during initialization, this feature is only available for a subset of material models now, namely material types 3, 24, 36, 81, 120, 123, 124, 133, 187, 224, 243, 251, and 258.

*DEFINE

*DEFINE_MULTISCALE

*DEFINE_MULTISCALE

Purpose: Associate beam sets with multiscale local model IDs for modeling detailed local model failure through the multiscale method.

Local Model/Beam Set and Coupling Type Association Cards. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	BSET	CTYPE					
Type	I	I	I					
Default	none	none	0					

VARIABLE	DESCRIPTION
ID	Multiscale local model ID to use. See *INCLUDE_MULTISCALE .
BSET	Beam set which uses this multiscale local model ID for failure modeling.
CTYPE	Coupling type (see Remark 2): EQ.0: Weak coupling (default) EQ.1: Strong coupling

Remarks:

1. **Explanation of capability.** See [*INCLUDE_MULTISCALE](#) for a detailed explanation of this capability.
2. **Coupling type.** When CTYPE is 0 or not specified, this multiscale feature generates the solid local models for the defined beams and uses a weak coupling algorithm (used in [*INCLUDE_MULTISCALE](#)) for the multiscale calculation. When CTYPE equals to 1, it generates the solid local models for the defined beams and uses a strong coupling algorithm (algorithm used in [*INCLUDE_COSIM](#); see [*INCLUDE_COSIM](#) for details) for the calculation.

DEFINE_NURBS_CURVE**DEFINE*****DEFINE_NURBS_CURVE**

Purpose: Define a NURBS curve using a univariate knot vector, a control polygon, and optionally a set of control weights. The knot vector defines the necessary shape functions and parameterizes the curve.

There is no limit on the size of the input data. Hence, the total number of keyword cards depends on the parameters defined on the first card. The total number of cards is $1 + \text{ceil}[(N + P + 1)/8] + N$, where N and P designate the number of control points forming the control polygon and the polynomial degree, respectively.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	N	P		TYPE	WFL		
Type	I	I	I		I	I		
Default	none	none	none		0	0		

Knot Vector Cards. The knot vector of length $N+P+1$ is given below, requiring $\text{ceil}[(N + P + 1)/8]$ cards in total. The knot vector must be normalized to the $[0,1]$ interval.

Card 2	1	2	3	4	5	6	7	8
Variable	K1	K2	K3	K4	K5	K6	K7	K8
Type	F	F	F	F	F	F	F	F

Control Point Cards. The spatial coordinates of the control points and the control weights are listed on N cards. Control weight entries are disregarded unless $WFL = 1$ on Card 1.

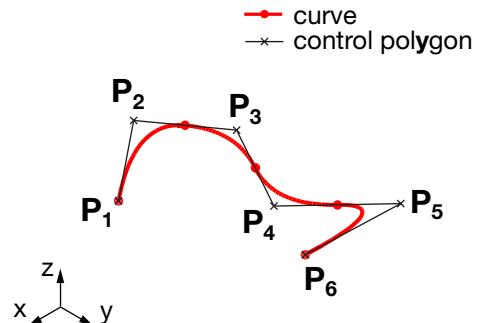
Card 3	1	2	3	4	5	6	7	8
Variable	X	Y	Z	W				
Type	F	F	F	F				

DEFINE**DEFINE_NURBS_CURVE**

VARIABLE	DESCRIPTION
ID	Curve ID. A unique number must be chosen.
N	Number of control points
P	Polynomial degree
TYPE	Coordinate type: EQ.0: Spatial EQ.1: Parametric
WFL	Flag for user defined control weights: EQ.0: Control weights are assumed to be uniform and positive, that is, the curve is a B-spline curve. The fourth column of Card 3 is disregarded. EQ.1: Control weights are defined on the fourth entry of Card 3.
Kn	Values of the univariate knot vector define in Card 2 with $n = 1, \dots, N + P + 1$.
Xk	Spatial coordinates in the global X-direction defined in Card 3 with $k = 1, \dots, N$.
Yk	Spatial coordinates in the global Y-direction defined in Card 3 with $k = 1, \dots, N$.
Zk	Spatial coordinates in the global Z-direction defined in Card 3 with $k = 1, \dots, N$.
Wk	Control weights defined in Card 3 with $k = 1, \dots, N$.

Remarks:

1. **Trimming Curves.** While the keyword is meant to store NURBS curves in two or three spatial dimensions, it is also employed to describe trimming curves that define trimmed NURBS elements/surfaces; see *ELEMENT_SHELL_NURBS_PATCH for further details. In the latter case, the control point coordinates are in fact parametric coordinates of the surface to be trimmed, that is, $(x, y, z, w) = (r, s, 0, w)$ and TYPE = 1 on Card 1.

**Figure 17-57.** An example quadratic NURBS curve.**Example:**See [Figure 17-57](#).

```
$ Define quadratic nurbs curve.  
*DEFINE_NURBS_CURVE  
$ CARD 1  
$-----+----N-----+----P-----PERI-----TYPE-----WFL  
      1           6           2           0           0           1  
$ CARD 2  
$-----+----1-----+----2-----+----3-----+----4-----+----5-----+----6-----+----7-----+----8  
      0.0          0.0          0.0         0.25          0.5          0.75         1.0          1.0  
      1.0  
$ CARD 3  
$-----+----X-----+----Y-----+----Z-----+----W  
      -0.353        0.471        0.850        1.000  
      0.050        0.600        1.000        1.000  
      0.444        0.444        0.900        0.995  
      0.125        0.000        0.800        1.000  
      0.222       -0.556        0.900        1.010  
      -0.353       -0.529        0.850        1.000
```

*DEFINE

*DEFINE_PART_FROM_LAYER

*DEFINE_PART_FROM_LAYER

Purpose: This keyword creates new blank(s) from an existing sheet blank. It is often used to model composite layers. This keyword applies to shell elements only. The new blanks cannot be adaptively refined.

Multiple layers can be defined. Input ends with the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	LAYER	PIDSRC	LAYOLD	MID	THICK		
Type	I	I	I	I	I	F		
Default	none	none	none	none	none	none		

VARIABLE	DESCRIPTION
PID	The part ID to be created.
LAYER	The layer ID of the PID, see Figure 17-58 .
PIDSRC	The part ID of the existing blank to be copied from.
LAYOLD	The layer ID of the existing blank.
MID	The material ID of the PID.
THICK	The thickness of the PID.

Remarks:

1. **Required Input Deck Cards.** *PART and *SECTION_SHELL do not need to be defined for the to-be created composite layer, but *MAT and contact cards still need to be defined.
2. **Contact.** Contact between outer layer of composites and the tools can be modeled using *CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE, while *CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_COMPOSITE, or *CONTACT_AUTOMATIC_SURFACE_TO_SURFACE can be used to model the interactions between two composite layers.

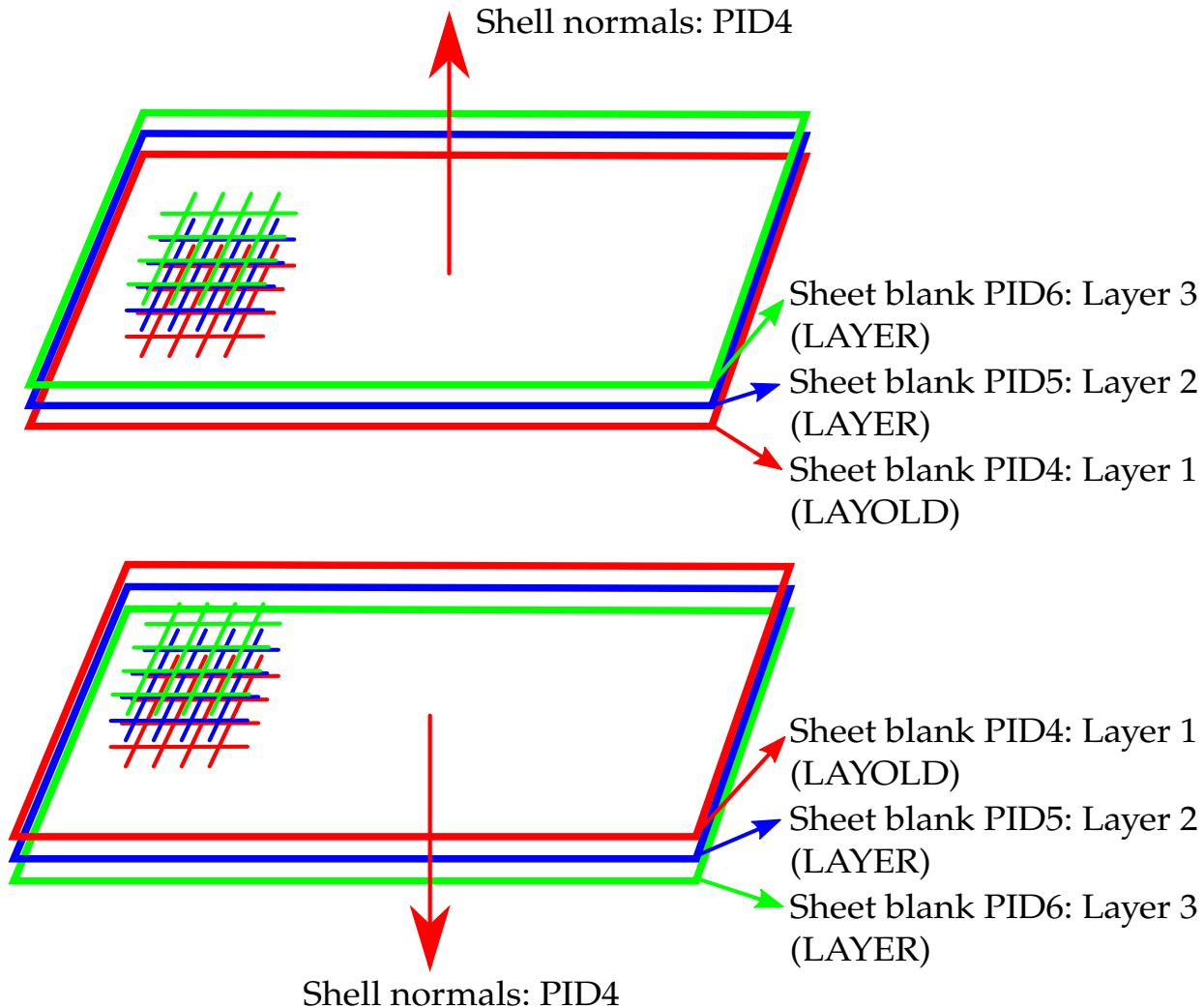


Figure 17-58. Defining the variables LAYER and LAYOLD.

Starting in Dev Revision 124300, contact definitions will be automatically defined between all adjacent layers using *CONTACT_SURFACE_TO_SURFACE during the simulation.

3. The following keyword defines new sheet metal blanks with part IDs 5 and 6 from an existing sheet blank with PID 4. It positions PID 5 on top of PID4 and PID6 on top of PID5.

```
*DEFINE_PART_FROM_LAYER
$-----1-----2-----3-----4-----5-----6-----7-----8
$     PID      LAYER    PIDSRCE   LAYOLD      MID     THICK
$      5        2         4          1        5     0.10
$      6        3         4          1        5     0.10
$-----1-----2-----3-----4-----5-----6-----7-----8
```

***DEFINE**

***DEFINE_PART_FROM_LAYER**

Revision Information:

This feature is available starting from Dev Revision 117590. Automatic contact is available in Dev Revision 124300.

DEFINE_PARTICLE_BLAST**DEFINE*****DEFINE_PARTICLE_BLAST**

Purpose: To define control parameters for particle based blast loading. This keyword was formerly called *PARTICLE_BLAST in versions R11 and earlier.

Card 1	1	2	3	4	5	6	7	8
Variable	LAGSID	LAGSTYPE	NODID	NODTYPE	HECID	HECTYPE	AIRCID	
Type	I	I	I	I	I	I	I	
Default	0	0	0	0	0	0	0	

Card 2	1	2	3	4	5	6	7	8
Variable	NPHE	NPAIR	IUNIT					
Type	I	I	I					
Default	0	0	0					

Card 3	1	2	3	4	5	6	7	8
Variable	IHETYPE	DENSITY	ENERGY	GAMMA	COVOL	DETO_V		
Type	I	F	F	F	F	F		
Default	0	0.	0.	0.	0.	0.		

DEFINE**DEFINE_PARTICLE_BLAST**

Card 4	1	2	3	4	5	6	7	8
Variable	DET _X	DET _Y	DET _Z	TDET	BTEND	NID		
Type	F	F	F	F	F	I		
Default	0.	0.	0.	0.	0.	0		

Card 5	1	2	3	4	5	6	7	8
Variable	BCX ₀	BCX ₁	BCY ₀	BCY ₁	BCZ ₀	BCZ ₁		
Type	F	F	F	F	F	F		
Default	0.	0.	0.	0.	0.	0.		

Card 6	1	2	3	4	5	6	7	8
Variable	IBC _{X0}	IBC _{X1}	IBC _{Y0}	IBC _{Y1}	IBC _{Z0}	IBC _{Z1}	BC_P	
Type	I	I	I	I	I	I	I	
Default	0	0	0	0	0	0	0	

VARIABLE**DESCRIPTION**

LAGSID Structure ID for particle structure interaction

LAGSTYPE Structure type:

EQ.0: Part Set

EQ.1: Part

NODID Discrete element sphere (DES) or Smooth particle hydrodynamics (SPH) ID for the interaction between particles and nodes.

NODTYPE Nodal type:

DEFINE_PARTICLE_BLAST**DEFINE**

VARIABLE	DESCRIPTION
	EQ.0: Node Set EQ.1: Node EQ.2: Part Set EQ.3: Part
HECID	Initial container for high explosive particle
HECTYPE	Structure type: EQ.0: Part Set EQ.1: Part EQ.2: Geometry, see *DEFINE_PBLAST_GEOMETRY
AIRCID	Initial geometry for air particles: EQ.0: Filled air particles to entire domain defined by Card 5 GT.0: Reference to *DEFINE_PBLAST_AIRGEO ID
NPHE	Number of high explosive particles
NPAIR	Number of air particles
IUNIT	Unit System EQ.0: Kg-mm-ms-K EQ.1: SI Units EQ.2: Ton-mm-s-K EQ.3: g-cm-us-K EQ.4: blob-in-s-K, where [blob] = [lbf][s] ² /[in]
IHETYPE	High Explosive type (see Remark 1): EQ.0: User defined EQ.1: TNT EQ.2: C4
DENSITY	High Explosive density for user defined explosive (see Remark 1).
ENERGY	High Explosive energy per unit volume for user defined explosive (see Remark 1).

DEFINE**DEFINE_PARTICLE_BLAST**

VARIABLE	DESCRIPTION
GAMMA	High Explosive fraction between C_p and C_v for user defined explosive (see Remark 1).
COVOL	High Explosive co-volume for user defined explosive (see Remark 1).
DET_V	High Explosive detonation velocity for user define explosive (see Remark 1).
DETX	Detonation point x
DETY	Detonation point y
DETZ	Detonation point z
TDET	Detonation time
BTEND	Blast end time
NID	An optional node ID defining the position of the detonation point. If defined, its coordinates will overwrite the DETX, DETY, and DETZ defined above.
BCX0	Global domain x -min
BCX1	Global domain x -max
BCY0	Global domain y -min
BCY1	Global domain y -max
BCZ0	Global domain z -min
BCZ1	Global domain z -max
IBCX0	Boundary conditions for global domain x -min: EQ.0: Free EQ.1: Rigid reflecting boundary
IBCX1	Boundary conditions for global domain x -max: EQ.0: Free EQ.1: Rigid reflecting boundary

VARIABLE	DESCRIPTION
IBCY0	Boundary conditions for global domain y -min: EQ.0: Free EQ.1: Rigid reflecting boundary
IBCY1	Boundary conditions for global domain y -max: EQ.0: Free EQ.1: Rigid reflecting boundary
IBCZ0	Boundary conditions for global domain z -min: EQ.0: Free EQ.1: Rigid reflecting boundary
IBCZ1	Boundary conditions for global domain z -max: EQ.0: Free EQ.1: Rigid reflecting boundary
BC_P	Pressure ambient boundary condition for global domain: EQ.0: Off (Default) EQ.1: On (Remark 2)

Remarks:

1. **Material constants for commonly used high explosives.** If the explosive material is TNT or C4, then the parameters DENSITY (ρ), ENERGY (e_0), GAMMA (γ), COVOL (COV), and DET_V (D) are set automatically to the values in the table below by LS-DYNA. Otherwise, the user must provide those values to define the explosive material.

IHETYPE	ρ	e_0	γ	COV	D
TNT	1630 $\frac{\text{kg}}{\text{m}^3}$	7 $\frac{\text{GJ}}{\text{m}^3}$	1.35	0.6	6930 $\frac{\text{m}}{\text{s}}$
C4	1601 $\frac{\text{kg}}{\text{m}^3}$	9 $\frac{\text{GJ}}{\text{m}^3}$	1.32	0.6	8193 $\frac{\text{m}}{\text{s}}$

2. **Pressure boundary conditions.** If pressure boundary conditions are used, particles will not escape from the global domain when the pressure in the domain is lower than the ambient. The ambient is automatically assumed as 1 ATM by default inside LS-DYNA.

DEFINE**DEFINE_PBLAST_AIRGEO*****DEFINE_PBLAST_AIRGEO**

Purpose: To define a simple geometry for initial air domain.

Card 1	1	2	3	4	5	6	7	8
Variable	GID	GTYPE1	GTYPE2					
Type	I	I	I					
Default	0	0	0					

Card 2	1	2	3	4	5	6	7	8
Variable	XA	YA	ZA	XB	YB	ZB		
Type	F	F	F	F	F	F		
Default	0.	0.	0.	0.	0.	0.		

Card 3	1	2	3	4	5	6	7	8
Variable	X0	Y0	Z0	G1	G2	G3		
Type	F	F	F	F	F	F		
Default	0.	0.	0.	0.	0.	0.		

Card 4	1	2	3	4	5	6	7	8
Variable	XC	YC	ZC	G4	G5	G6		
Type	F	F	F	F	F	F		
Default	0.	0.	0.	0.	0.	0.		

VARIABLE	DESCRIPTION
GID	ID of a GEOMETRY defining initial air particle domain.
GTYPE1	Geometry type
GTYPE2	EQ.1: box EQ.2: sphere EQ.3: cylinder EQ.4: ellipsoid EQ.5: hemisphere (see Remark 1)
XA, YA, ZA	(XA, YA, ZA) defines a vector of the <i>x</i> -axis
XB, YB, ZB	(XB, YB, ZB) defines a vector of the <i>y</i> -axis
X0, Y0, Z0	Center coordinates of air domain
G1	Dimension value depending on GTYPE. GTYPE.EQ.1: length of <i>x</i> edge GTYPE.EQ.2: Radius of sphere GTYPE.EQ.3: Radius of cross section GTYPE.EQ.4: length of <i>x</i> -axes GTYPE.EQ.5: Radius of hemisphere
G2	Dimension value depending on GTYPE. GTYPE.EQ.1: length of <i>y</i> edge GTYPE.EQ.3: length of cylinder GTYPE.EQ.4: length of <i>y</i> -axes
G3	Dimension value depending on GTYPE. GTYPE.EQ.1: length of <i>z</i> edge GTYPE.EQ.4: length of <i>z</i> -axes
XC, YC, ZC	Center coordinates of domain excluded from the air domain
G4, G5, G6	See definition of G1, G2, G3

Remarks:

1. If GTYPE1/GTYPE2 is 5, the hemisphere is defined in negative z direction defined by the cross product of the y and x axis.

DEFINE_PBLAST_GEOMETRY**DEFINE*****DEFINE_PBLAST_GEOMETRY**

Purpose: To define a simple geometry for high explosives domain.

Card 1	1	2	3	4	5	6	7	8
Variable	GID	GTYPE						
Type	I	I						
Default	0	none						

Card 2	1	2	3	4	5	6	7	8
Variable	XA	YA	ZA	XB	YB	ZB	ITYPE	
Type	F	F	F	F	F	F	I	
Default	0.	0.	0.	0.	0.	0.	0	

Card 3	1	2	3	4	5	6	7	8
Variable	XC	YC	ZC					
Type	F	F	F					
Default	0.	0.	0.					

Card 4	1	2	3	4	5	6	7	8
Variable	G1	G2	G3					
Type	F	F	F					
Default	0.	0.	0.					

DEFINE**DEFINE_PBLAST_GEOMETRY**

VARIABLE	DESCRIPTION
GID	ID of a GEOMETRY defining high explosive particle domain.
GTYPE	Geometry type <ul style="list-style-type: none">EQ.1: BoxEQ.2: SphereEQ.3: CylinderEQ.4: EllipsoidEQ.5: Hemisphere (see Remark 1)
XA, YA, ZA	ITYPE.EQ.0: Local <i>x</i> -axis ITYPE.EQ.0: Starting coordinate of the charge used to internally calculate the local coordinate system
XB, YB, ZB	ITYPE.EQ.0: Local <i>y</i> -axis ITYPE.EQ.1: Ending coordinate of the charge used to internally calculate the local coordinate system
ITYPE	Flag for determining how the domain geometry of the high explosive particles is generated: <ul style="list-style-type: none">EQ.0: Particles are generated in the given geometry with the local coordinate system provided with XA, YA, ZA, XB, YB, and ZB (default).EQ.1: Particles are generated in the given geometry using starting and ending coordinates given with XA, YA, ZA, XB, YB, and ZB.
XC	X-coordinate of charge center
YC	Y-coordinate of charge center
ZC	Z-coordinate of charge center
G1	GTYP.EQ.1: Length of <i>x</i> edge GTYP.EQ.2: Radius of sphere GTYP.EQ.3: Radius of cross section GTYP.EQ.4: Length of <i>x</i> -axes

VARIABLE	DESCRIPTION
	GTYPE.EQ.5: Radius of hemisphere
G2	GTYPE.EQ.1: Length of <i>y</i> edge GTYPE.EQ.3: Length of cylinder GTYPE.EQ.4: Length of <i>y</i> -axes
G3	GTYPE.EQ.1: Length of <i>z</i> edge GTYPE.EQ.4: Length of <i>z</i> -axes

Remarks:

1. **Hemisphere.** If GTYPE is 5, the hemisphere is defined in the negative *z*-direction given by the cross product of the *y*- and *x*-axes.

*DEFINE

*DEFINE_PLANE

*DEFINE_PLANE

Purpose: Define a plane with three non-collinear points. The plane can be used to define a reflection boundary condition for problems like acoustics.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	X1	Y1	Z1	X2	Y2	Z2	CID
Type	I	F	F	F	F	F	F	I
Default	0	0.0	0.0	0.0	0.0	0.0	0.0	0

Card 2	1	2	3	4	5	6	7	8
Variable	X3	Y3	Z3					
Type	F	F	F					
Default	0.0	0.0	0.0					

VARIABLE	DESCRIPTION
PID	Plane ID. A unique number has to be defined.
X1	X-coordinate of point 1.
Y1	Y-coordinate of point 1.
Z1	Z-coordinate of point 1.
X2	X-coordinate of point 2.
Y2	Y-coordinate of point 2.
Z2	Z-coordinate of point 2.
CID	Coordinate system ID applied to the coordinates used to define the current plane. The coordinates X1, Y1, Z1, X2, Y2, Z2, X3, Y3 and Z3 are defined with respect to the coordinate system CID.
X3	X-coordinate of point 3.

VARIABLE	DESCRIPTION
Y3	Y-coordinate of point 3.
Z3	Z-coordinate of point 3.

Remarks:

1. **Setup Suggestions.** The coordinates of the points must be separated by a reasonable distance and not collinear to avoid numerical inaccuracies.

*DEFINE

*DEFINE_POINT_CLOUD

*DEFINE_POINT_CLOUD_{OPTION}

Available options include:

<BLANK>

TITLE

Purpose: Define a set of points in the physical space. One or more fields reference a point cloud to define spatially varying data on a set of points; see [*DEFINE_FIELD](#).

Card Summary:

Card Title. This card is included if the TITLE keyword option is used.

TITLE							
-------	--	--	--	--	--	--	--

Card 1. This card is required.

PCID							
------	--	--	--	--	--	--	--

Card 2. This card is required. Include as many instantiations of this card as needed. This input ends at the next keyword ("*") card. See [Remarks 1](#) and [2](#).

X	Y	Z		
---	---	---	--	--

Data Card Definitions:

Title Card. Additional card for the TITLE keyword option.

Card Title	1	2	3	4	5	6	7	8
Variable				TITLE				
Type				A80				

VARIABLE

DESCRIPTION

TITLE

Name or description of the point cloud defined in this keyword.

DEFINE_POINT_CLOUD**DEFINE**

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

VARIABLE	DESCRIPTION
PCID	Point cloud ID. A unique ID number must be used.

Point Coordinates Card. Include as many instantiations of this card as needed, one for each point in the point cloud. This input ends at the next keyword ("*") card. See [Remarks 1](#) and [2](#).

Card 2	1	2	3	4	5	6	7	8
Variable	X		Y		Z			
Type	F		F		F			
Default	0.0		0.0		0.0			

VARIABLE	DESCRIPTION
X	x -coordinate of the i^{th} point in the point cloud in the global coordinate system
Y	y -coordinate of the i^{th} point in the point cloud in the global coordinate system
Z	z -coordinate of the i^{th} point in the point cloud in the global coordinate system

Remarks:

1. **Data consistency.** Each instance of Card 2 enables defining a single point in the point cloud. The input reader permits an arbitrary number of instantiations of Card 2. However, the number of points in the point cloud must be consistent with the number of field data included in [*DEFINE_FIELD](#), meaning that the

number of points \times NV must equal the total number of input field data (NV is the number of data input for each point). See [*DEFINE_FIELD](#) for details.

2. **Point ID.** Points defined with *DEFINE_POINT_CLOUD are not associated with an ID. Therefore, these points cannot be explicitly referenced by other keywords, such as *SET.

DEFINE_POROUS**DEFINE*****DEFINE_POROUS_OPTION**

Available options include:

ALE

LAGRANGIAN

Purpose: The *DEFINE_POROUS_ALE card defines the Ergun porous coefficients for ALE elements. It is to be used with *LOAD_BODY_POROUS. This card with the LAGRANGIAN option, *DEFINE_POROUS_LAGRANGIAN, defines the porous coefficients for Lagrangian elements and is to be used with [*CONSTRAINED_LAGRANGE-IN_SOLID](#) (Lagrangian structure parts with CTYPE = 11 or 12).

Card 1	1	2	3	4	5	6	7	8
Variable	EIDBEG	EIDEND	LOCAL	VECID1	VECID2	USERDEF		
Type	I	I	I	I	I	I		
Default	none	0	0	0	0	0		

Card 2	1	2	3	4	5	6	7	8
Variable	AXX	AYX	AXZ	BXX	BYX	BXZ		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	0.0	0.0	0.0		

Card 3	1	2	3	4	5	6	7	8
Variable	AYX	AYY	AYZ	BYX	BYY	BYZ		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	0.0	0.0	0.0		

DEFINE**DEFINE_POROUS**

Card 4	1	2	3	4	5	6	7	8
Variable	AZX	AZY	AZZ	BZX	BZY	BZZ		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	0.0	0.0	0.0		

VARIABLE	DESCRIPTION
EIDBEG, EI-DEND	EIDBEG, EIDEND > 0: Range of thick porous element IDs. These are solids in 3D and shells in 2D. EIDBEG, EIDEND < 0: Range of thin porous element IDs. These are shells in 3D and beams in 2D. The ALE option does <i>not</i> support thin porous elements. EIDBEG > 0, EIDEND = 0: EIDBEG is a set of thick porous elements EIDBEG > 0, EIDEND < 0: EIDBEG is a set of thin porous elements
LOCAL	Flag to activate an element coordinate system: EQ.0: The forces are applied in the global directions. EQ.1: The forces are applied in a local system attached to the element. The system is consistent with DIREC = 1 and CTYPE = 12 in *CONSTRAINED_LAGRANGE_IN_SOLID . For CTYPE = 11, LOCAL is always 1 and the <i>x</i> -axis is aligned with the element normal while the <i>y</i> -axis passes through the element center and the first node in the element connectivity (*ELEMENT_BEAM in 2D or *ELEMENT_SHELL in 3D)
VECID1, VE-CID2	*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 this latter is not orthogonal to VECID1, its direction will be corrected with a cross-product of <i>z</i> - and <i>x</i> -vectors. The vectors are stored as isoparametric locations to update their directions if the element deforms or rotates.

VARIABLE	DESCRIPTION
USERDEF	Flag to compute A_{ij} and B_{ij} with a user defined routine in the file dyn21.F called lagpor_getab_userdef. The file is part of the general package usermat.
A_{ij}	Viscous matrix for the porous flow Ergun equation (see Remark 1)
B_{ij}	Inertial matrix for the porous flow Ergun equation (see Remark 1)

Remarks:

1. **Ergun equation.** The Ergun equation computing the pressure gradient along each direction $i = x, y, z$ can be written as follows:

$$\frac{dP}{dx_i} = \sum_{j=1}^3 [\mu A_{ij} V_j + \rho B_{ij} |V_j| V_j]$$

where,

- a) V_i is the relative velocity of the flow in the porous media
- b) A_{ij} are the viscous coefficients of the Ergun-type porous flow equation in the i^{th} direction.. This matrix is similar to the viscous coefficients used in [*LOAD_BODY_POROUS](#).
- c) B_{ij} are the inertial coefficient of the Ergun-type porous flow equation in the i^{th} direction. This matrix is similar to the inertial coefficients used in [*LOAD_BODY_POROUS](#).

If this keyword defines the porous properties of Lagrangian elements in [*CONSTRAINED_LAGRANGE_IN_SOLID](#), the porous coupling forces are computed with the pressure gradient as defined above instead of the equations used for CTYPE = 11 and 12.

*DEFINE

*DEFINE_PRESSURE_TUBE

*DEFINE_PRESSURE_TUBE

Purpose: Define a gas-filled tube to simulate interior pressure waves resulting from changes in the tube cross-section area over time. The tube is specified with tubular beam elements, and the initial gas volume is determined by beam cross-section area and initial element lengths. Area changes are either given by beam contact penetration (only mortar contacts are currently supported) or by deformation of automatically generated shell/solid elements. The pressure calculation is not coupled with the beam deformation (except from contact penetration) and does not use any data from the material card. Pressure and tube area at the beam nodes are output through [*DATABASE_PRTUBE](#).

Card Summary:

Card 1. This card is required.

PID	WS	PR	MTD	TYPE	GAMMA		
-----	----	----	-----	------	-------	--	--

Card 2. This card is optional.

VISC	CFL	DAMP	BNDL	BNDR	CAVL	CAVR	SNODE
------	-----	------	------	------	------	------	-------

Card 3a. This card is only read if | TYPE | = 1.

NSHL	ELFORM	NIP	SHRF	BPID	ISAVE	IORIEN	
------	--------	-----	------	------	-------	--------	--

Card 3b. This card is only read if | TYPE | = 2.

NSLD	ELFORM	NTHK		BPID	ISAVE	IORIEN	
------	--------	------	--	------	-------	--------	--

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	PID	WS	PR	MTD	TYPE	GAMMA		
Type	I	F	F	I	I	F		
Default	0	0.0	0.0	0	0	1.0		

VARIABLE	DESCRIPTION
PID	<p>Part ID of the tube. All connected beam elements in the part will model a tube. Only tubular beam elements are allowed, that is, ELFORM = 1, 4, 5, and 11 with CST = 1 on *SECTION_BEAM. The initial tube cross-sectional area is calculated using the beam inner diameter TT1/TT2 fields in *SECTION_BEAM. The outer diameter TS1/TS2 fields in *SECTION_BEAM are used if no inner diameter is given.</p> <p>The beam elements may not contain junctions. Two different parts on which *DEFINE_PRESSURE_TUBE is defined may not share nodes. For MPP, all elements in the part will be on a single processor.</p>
WS	Speed of sound c_0 in the gas
PR	Initial gas pressure p_0 inside tube
MTD	<p>Solution method (see Remark 2):</p> <ul style="list-style-type: none">EQ.0: Standard Galerkin FEMEQ.1: Discontinuous GalerkinEQ.2: Discontinuous Galerkin on isentropic Euler equations
TYPE	<p>Tube element type (see Figure 17-59):</p> <ul style="list-style-type: none">EQ.0: The tube is entirely simulated with beam elements. The contact penetration of the beam elements gives the cross-sectional area. Contact stiffness governs the mechanical response in the radial direction of the beam elements. Only mortar contacts are supported.EQ.1: The tube is simulated by automatic generation of shell elements, which are assigned the beam part ID and the beam material model. A new part ID is given to the beam elements, which are no longer part of the mechanical solution. Contacts and other properties associated with the old beam part ID apply to the new shell part. The shell element nodes give the cross-sectional area. The shells entirely govern the mechanical response. All contact definitions are supported. Constraints defined by *BOUNDARY_SPC, *BOUNDARY_PRESCRIBED_MOTION, *CONSTRAINED_EXTRA_NODES, and *CONSTRAINED_NODAL_RIGID_BODY and nodes that are shared with a rigid body are moved to the new shell tube.

*DEFINE

*DEFINE_PRESSURE_TUBE

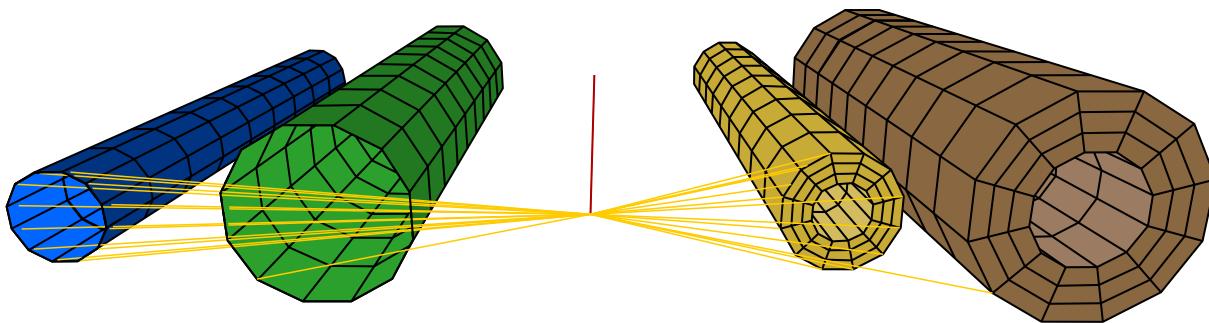


Figure 17-59. Automatically generated tubes using TYPE = 1, -1, 0, 2, and -2, from left to right. The beam tube ends are connected with *CONSTRAINED_NODAL_RIGID-BODY, which applies differently to the generated shell/solid tubes, depending on the sign of TYPE.

VARIABLE

DESCRIPTION

EQ.2: The tube is simulated by automatic generation of solid elements, similarly to TYPE = 1 above.

LT.0: Automatic generation of elements as above, but the beam nodes are given new nodal IDs. The old beam NIDs are moved to the automatically generated tube (one row of nodes along the length). Any nodal constraints thus apply to the new tube instead of the beam element tube. See [Figure 17-59](#) for an example of different values of TYPE and how they affect nodal constraints.

GAMMA

Adiabatic index, γ . It must be ≥ 1 . It is only used for MTD = 2.

Card 2	1	2	3	4	5	6	7	8
Variable	VISC	CFL	DAMP	BNDL	BNDR	CAVL	CAVR	SNODE
Type	F	F	F	F	F	F	F	I
Default	1.0	0.9	0.0	0.0	0.0	0.0	0.0	0

VARIABLE

DESCRIPTION

VISC

MTD.EQ.0: Artificial viscosity multiplier ($VISC > 0.0$); see [Remark 2](#). A smaller value gives a more resolved pulse at shorter wavelengths but may lead to instabilities. We recommend the default value for typical

VARIABLE	DESCRIPTION
	automotive crash applications (tube length ~2m, diameter ~5mm, pressure pulse width ~5ms).
MTD.GT.0:	Slope limiter smoothing factor; see Remark 2 . Smaller values give a more resolved pulse at shorter wavelengths but may lead to instabilities. Larger values lead to a smeared pulse.
CFL	Stability factor ($CFL > 0.0$); see Remark 2 . A smaller value increases stability at the expense of increased computational cost. For typical automotive crash applications, the default value is recommended.
DAMP	Linear damping ($DAMP \geq 0.0$); see Remark 2 .
BND <i>i</i>	Left/right boundary condition ($0 \leq BNDi \leq 1$); see Remark 3 . Special cases are: EQ.0.0: Closed tube end, that is, zero velocity boundary condition EQ:0.5: Non-reflecting boundary condition EQ:1.0: Open tube end, that is, constant pressure boundary condition Left/right tube end is automatically assigned to the lowest/highest beam node number on the tube, respectively.
CAV <i>i</i>	Left/right cavity; see Remark 4 . GT.0.0: A cavity replaces the elements near the end of the tube. The integer part of $CAVi$ determines the number of beam elements that belong to the cavity. The remainder of $CAVi$ determines the boundary condition on the interface between the tube and the cavity. LT:0.0: A cavity extends the tube by adding new beam elements. The length of the added cavity is given by $L = \text{int}(CAVi)/100$ where $\text{int}(x)$ truncates the decimal portion of x (leaving an integer). The remainder of $ CAVi $ determines the boundary condition on the interface between the tube and the cavity.
SNODE	Optional starting node. This node determines the left end of the tube. If not set, the tube starts at the lowest numbered beam node.

DEFINE**DEFINE_PRESSURE_TUBE**

Automatically Generated Shells Card. This card is only read if |TYPE| = 1.

Card 3a	1	2	3	4	5	6	7	8
Variable	NSHL	ELFORM	NIP	SHRF	BPID	ISAVE	IORIEN	
Type	F	F	F	F	I	I	I	
Default	12.0	16.0	3.0	1.0	optional	0	0	

Automatically Generated Solids Card. This card is only read if |TYPE| = 2.

Card 3b	1	2	3	4	5	6	7	8
Variable	NSLD	ELFORM	NTHK		BPID	ISAVE	IORIEN	
Type	F	F	F		I	I	I	
Default	12.0	1.0	3.0		optional	0	0	

VARIABLE	DESCRIPTION
NSHL/NSLD	Number of automatically generated shells/solids on the circumference of the tube
ELFORM	ELFORM for automatically generated shells/solids; see *SECTION_SHELL/SOLID .
NIP	Number of through-thickness integration points for automatically generated shells; see NIP in *SECTION_SHELL .
NTHK	Number of solid elements in the thickness of the tube for automatically generated solids
SHRF	Shear correction factor for automatically generated shells; see SHRF in *SECTION_SHELL .
BPID	Optional PID given to the beam elements when automatically generating shells/solids.
ISAVE	Save shell/solid geometry and connectivity to a keyword file: EQ.0: No saving (default)

VARIABLE	DESCRIPTION
	<p>EQ.1: Save to keyword file <code>prtube.k</code></p> <p>Subsequent runs can include the generated file, in which case TYPE and Card 3 are ignored for saved tubes. Boundary conditions and contacts are not saved. Thus, beam boundary conditions and contacts are not transferred to the saved shell/solid tubes in subsequent runs. Therefore, boundary conditions, contacts, etc., must still be supplied to the saved shell/solid geometry.</p>
IORIEN	<p>Control circumferential orientation of automatically generated shells/solids:</p> <p>EQ.0: Orient the first end segment with the global coordinate system. Subsequent segments are created in a way to minimize twisting along the length of the tube. (default)</p> <p>EQ.1: Orient each segment using the third beam node if it exists. Otherwise, use the vector from *ELEMENT_BEAM_ORIENTATION to orient each segment.</p>

Remarks:

1. **Pressure tube model.** The pressure tube is modeled with an acoustic approximation of the one-dimensional compressible Euler equations for pipes with varying thickness:

$$\begin{aligned}\frac{\partial}{\partial t}(\rho A) + \frac{\partial}{\partial x}(\rho u A) &= 0 \\ \frac{\partial}{\partial t}(\rho u A) + \frac{\partial}{\partial x}(\rho u^2 A + pA) &= p \frac{\partial A}{\partial x} \\ \frac{\partial}{\partial t}(EA) + \frac{\partial}{\partial x}(u(E+p)A) &= 0\end{aligned}$$

where $A = A(x, t)$ is the cross-sectional area and ρ, p, u , and E are density, pressure, velocity, and energy per unit volume, respectively. The above system is closed under the constitutive relations

$$E = \rho e + \frac{\rho u^2}{2}, \quad p = p(\rho, e),$$

where e is the internal energy per unit mass.

For an isentropic and isothermal flow, the pressure is proportional to the density, that is,

$$p = c_0^2 \rho,$$

and the energy equation can be dropped. This is a good approximation of the Euler equations for acoustic flows where the state variables are smooth perturbations around a background state. No shocks develop over time for such flows but may be present from initial/boundary values or source terms.

Assuming small perturbations, linearization around $\rho_0, p_0, u_0 = 0$ gives the acoustic approximation

$$\begin{aligned}\frac{\partial}{\partial t}(A\rho) + \rho_0 \frac{\partial y}{\partial x} &= 0 \\ \rho_0 \frac{\partial y}{\partial t} + c_0^2 \frac{\partial}{\partial x}(A\rho) &= p \frac{\partial A}{\partial x}\end{aligned}$$

where $y = Au$. Expressed in y and p we have

$$\begin{aligned}\frac{\partial p}{\partial t} + \frac{\partial \ln A}{\partial t} p + \frac{p_0}{A} \frac{\partial y}{\partial x} &= 0 \\ \frac{\partial y}{\partial t} + A \frac{c_0^2}{p_0} \frac{\partial p}{\partial x} &= 0\end{aligned}$$

2. **Solution methods for the pressure tube.** We have two solution methods for the linearized system given in [Remark 1](#) and one for the isothermal nonlinear system. MTD on Card 1 sets the method.

- a) *Continuous Galerkin (MTD = 0).* The linearized system can be solved using the standard Continuous Galerkin finite element method, using piecewise linear basis functions and artificial viscosity. Linear damping is then added to model energy losses from friction between the gas and the tube walls. With artificial viscosity and damping, the linear system can be written as

$$\begin{aligned}\frac{\partial p}{\partial t} + \frac{\partial \ln A}{\partial t} p + \frac{p_0}{A} \frac{\partial y}{\partial x} &= \epsilon \frac{\partial^2 p}{\partial x^2} - \text{DAMP} \times (p - p_0) \\ \frac{\partial y}{\partial t} + A \frac{c_0^2}{p_0} \frac{\partial p}{\partial x} &= \epsilon \frac{\partial^2 y}{\partial x^2}\end{aligned}$$

The artificial viscosity, ϵ , is proportional to the maximum initial beam element length, that is,

$$\epsilon = \text{VISC} \times c_0 \max_i \Delta x_i .$$

Small values of VISC may lead to convergence problems while large values smear out the solution. Time integration is independent of the mechanical solver and uses a step size less than or equal to the global time step, satisfying a CFL condition

$$\Delta t < \min_i \frac{\text{CFL} \times \Delta x_i}{\Delta x_i \left| \frac{\partial \ln A}{\partial t} \right| + 3c_0} .$$

- b) *Discontinuous Galerkin (MTD = 1).* Alternatively, the linear system can be solved with the Discontinuous Galerkin method, using piecewise linear

basis functions on each element, Lax-Friedrich flux, and a MUSCL flux limiter to limit spatial oscillations. Time integration is done with Heun's method which is a 2nd order TVD Runge Kutta method. In this case, linear damping is again added to the model.

$$\begin{aligned}\frac{\partial p}{\partial t} + \frac{\partial \ln A}{\partial t} p + \frac{p_0}{A} \frac{\partial y}{\partial x} &= -\text{DAMP} \times (p - p_0) \\ \frac{\partial y}{\partial t} + A \frac{c_0^2}{p_0} \frac{\partial p}{\partial x} &= 0\end{aligned}$$

VISC > 0 is a smoothing factor for the MUSCL limiter with VISC = 0 indicating no smoothing. The Discontinuous Galerkin method gives less diffusion (smearing) than the Continuous Galerkin method.

- c) *Discontinuous Galerkin with isentropic Euler equations (MTD = 2).* This method uses Discontinuous Galerkin to solve the damped nonlinear isentropic Euler equations

$$\begin{aligned}\frac{\partial}{\partial t} (\rho A) + \frac{\partial}{\partial x} (\rho u A) &= -\text{DAMP} \times (\rho - \rho_0) A, \\ \frac{\partial}{\partial t} (\rho u A) + \frac{\partial}{\partial x} (\rho u^2 A + p A) &= p \frac{\partial A}{\partial x},\end{aligned}$$

with

$$\begin{aligned}p &= \frac{c_0^2}{\gamma \rho_0^{\gamma-1}} \rho^\gamma, \\ \rho_0 &= \frac{\gamma p_0}{c_0^2}.\end{aligned}$$

VISC has the same meaning as for MTD = 1. This method may be beneficial for scenarios with large pressure/velocity perturbations, where the acoustic approximation is poor.

3. Boundary conditions.

The boundary conditions are

$$(p - p_0) = \pm Z u ,$$

where the sign depends on whether it is the left or right boundary. The impedance is defined as

$$Z = \frac{p_0}{c_0} \frac{\text{BND}i}{(1 - \text{BND}i)} .$$

A closed end, BNDi = 0, thus corresponds to Z = 0 and u = 0, meaning a zero-velocity condition. An open end, BNDi = 1, corresponds to Z = ∞ and p = p0, a constant-pressure condition. For the intermediate case, we note that if A is a constant, DAMP = 0, and ε = 0, then the acoustic equations become

$$\frac{\partial p}{\partial t} + p_0 \frac{\partial u}{\partial x} = 0$$

$$\frac{\partial u}{\partial t} + \frac{c_0^2}{p_0} \frac{\partial p}{\partial x} = 0$$

which is equivalent to the acoustic wave equation

$$\frac{\partial^2 p}{\partial t^2} - c_0^2 \frac{\partial^2 p}{\partial x^2} = \left(\frac{\partial}{\partial t} - c_0 \frac{\partial}{\partial x} \right) \left(\frac{\partial}{\partial t} + c_0 \frac{\partial}{\partial x} \right) p = 0 .$$

Differentiation of the boundary condition (and assuming that the wave equation holds on the boundary) gives

$$\frac{\partial p}{\partial t} = \pm Z \frac{\partial u}{\partial t} = \pm \frac{p_0}{c_0} \frac{\text{BND}i}{(1 - \text{BND}i)} \frac{\partial u}{\partial t} = \mp c_0 \frac{\text{BND}i}{(1 - \text{BND}i)} \frac{\partial p}{\partial x} .$$

Setting $\text{BND}i = 0.5$ implies

$$\frac{\partial p}{\partial t} = \mp c_0 \frac{\partial p}{\partial x} ,$$

which is the left/right traveling solution to the acoustic wave equation; that is, $\text{BND}i = 0.5$ corresponds to $Z = Z_0 = p_0/c_0$ and a non-reflecting boundary condition.

The boundary condition can also be seen as a jump in the cross-section area of the tube, meaning if we have two tubes with area A_l and A_r , then a wave (p, u) traveling from A_l to A_r can be split into a reflected part $(Rp, -Ru)$ and a transmitted part (Tp, Tu) , with reflectance/transmission coefficients

$$R = \frac{A_l - A_r}{A_l + A_r} , \quad T = \frac{2A_l}{A_l + A_r} .$$

Reflectance can also be expressed with respect to impedance at the interface:

$$R = \frac{Z_0 - Z}{Z_0 + Z} .$$

Combining the expressions for the reflectance, the boundary condition can thus be seen as an infinite extension of the original tube, with a jump from the tube area A_{in} at the boundary to an outside area

$$A_{out} = \frac{\text{BND}i}{(1 - \text{BND}i)} A_{in} .$$

$\text{BND}i = 0.5$, thus, corresponds to $A_{out} = A_{in}$, $\text{BND}i = 0$ gives $A_{out} = 0$, and $\text{BND}i = 1$ gives $A_{out} = \infty$.

4. **Cavities.** Cavities close to the tube ends can be simulated in three ways:

- a) Altering tube thickness with [*ELEMENT_BEAM_THICKNESS](#).

- b) Setting $CAVi > 0$.

For $CAVi > 0$, the original tube geometry is intact, and $n = \text{int}(CAVi)$ (the integer portion of $CAVi$) elements from the end belong to a cavity, that is, a different cross-section area (that can only be seen in binout/prtubeout). The boundary condition, $\alpha = CAVi - n$, on the interface between the tube and the cavity determines the cavity's cross-sectional area. The n elements closest to the boundary have the new area

$$A_{\text{cav}} = \frac{\alpha}{(1 - \alpha)} A_{n+1} ,$$

where A_{n+1} is the initial cross-section the area of element $n + 1$ from the boundary. For example, for a tube with initial cross-section area $A = 10$, $CAVL = 50.8$ means that the 50 elements closest to the left end will have the constant cross-section area

$$A_{\text{cav}} = \frac{0.8}{(1 - 0.8)} 10 = 40 .$$

- c) Setting $CAVi < 0$.

For $CAVi < 0$, a cavity of length $L = \text{int}(|CAVi|)/100$ is added at the end by creating new beam elements (at least 5 beam elements are used). These new beam elements will not be converted to shells/solids regardless of TYPE; thus, they are not part of the mechanical solution. The cavity cross-section area is calculated as above using the boundary condition, $\alpha = |CAVi| - \text{int}(|CAVi|)$.

When using $CAVi$, the area A_{cav} overrides any cross-sectional area updates for the cavity elements resulting from physical deformation. Also, the boundary conditions set by $BNDi$ still hold at the outer ends of the (extended) tube.

*DEFINE

*DEFINE_QUASAR_COUPLING

*DEFINE_QUASAR_COUPLING

Purpose: Define LS-DYNA node/node set that interacts with Cadlm's QUASAR ROM model. Each coupling needs to have its own keyword card and will not accept multiple entries.

WARNING: We added Card 2 in September of 2020. It breaks backward compatibility. Old input decks that include this keyword will not work with executables created September of 2020 and later.

Card 1	1	2	3	4	5	6	7	8
Variable	NODE	TYPE	ROMID	PID	PTYPE	IOPT	CID	EX_ND
Type	I	I	I	I	I	I	I	I

Card 2	1	2	3	4	5	6	7	8
Variable	FRCFRQ							
Type	I							

Card 3	1	2	3	4	5	6	7	8
Variable				FILENAME1				
Type				A80				

Card 4	1	2	3	4	5	6	7	8
Variable				FILENAME2				
Type				A80				

User Defined Constants Card. Optional card for user defined constants. This input ends at the next keyword ("*") card.

Card 5	1	2	3	4	5	6	7	8
Variable	VAR1	VAR2	VAR3	VAR4	VAR5	VAR6	VAR7	VAR8
Type	F	F	F	F	F	F	F	F

VARIABLE	DESCRIPTION
NODE	Coupled node/node set
TYPE	Region type: EQ.0: Node ID EQ.1: Node set ID
ROMID	Cadlm's ROM ID
PID	LS-DYNA part/part set ID
PTYPE	Type for PID: EQ.0: Part ID (Default) EQ.1: Part set ID
IOPT	Option for exchanging data between LS-DYNA/Cadlm Quasar: EQ.0: Default. LS-DYNA outputs nodal translational and rotational coordinates and receives nodal translational and rotational forces. EQ.1: LS-DYNA outputs nodal translational and rotational displacements and receives nodal translational and rotational forces. EQ.2: LS-DYNA outputs nodal translational coordinates and receives nodal translational forces. EQ.3: LS-DYNA outputs nodal translational displacements and receives nodal translational forces.
CID	Reference coordinate system needed to transform the data from the LS-DYNA global system to the Quasar local system

DEFINE**DEFINE_QUASAR_COUPLING**

VARIABLE	DESCRIPTION
EX_ND	Node set to exclude from Quasar output. LS-DYNA still expects the complete set of data. (Quasar can predict the forces from a reduced data set.)
FRCFRQ	Number of cycles between QUASAR force updates for the coupling interface
FILENAME1	LS-DYNA output file to QUASAR
FILENAME2	QUASAR output file to LS-DYNA
VAR <i>i</i>	User defined constant

DEFINE_REGION**DEFINE*****DEFINE_REGION**

Purpose: Define a volume of space, optionally in a local coordinate system.

Card Summary:

Card 1. This card is required.

ID	TITLE						
----	-------	--	--	--	--	--	--

Card 2. This card is required.

TYPE	CID	MOVE					
------	-----	------	--	--	--	--	--

Card 3a. This card is included if and only if TYPE = 0.

XMN	XMX	YMN	YMX	ZMN	ZMX		
-----	-----	-----	-----	-----	-----	--	--

Card 3b. This card is included if and only if TYPE = 1.

XC1	YC1	ZC1	RMIN1	RMAX1			
-----	-----	-----	-------	-------	--	--	--

Card 3c.1. This card is included if and only if TYPE = 2.

XC2	YC2	ZC2	AX2	AY2	AZ2	RMIN2	RMAX2
-----	-----	-----	-----	-----	-----	-------	-------

Card 3c.2. This card is included if and only if TYPE = 2.

L2							
----	--	--	--	--	--	--	--

Card 3d.1. This card is included if and only if TYPE = 3.

XC3	YC3	ZC3	AX3	AY3	AZ3	BX3	BY3
-----	-----	-----	-----	-----	-----	-----	-----

Card 3d.2. This card is included if and only if TYPE = 3.

BZ3	RA3	RB3	RC3				
-----	-----	-----	-----	--	--	--	--

DEFINE**DEFINE_REGION****Data Card Definitions:**

Card 1	1	2	3	4	5	6	7	8
Variable	ID				TITLE			
Type	I				A70			

VARIABLE	DESCRIPTION							
ID	Region ID							
TITLE	Title for this region							

Card 2	1	2	3	4	5	6	7	8
Variable	TYPE	CID	MOVE					
Type	I	I	0					

VARIABLE	DESCRIPTION
TYPE	Region type: EQ.0: Box EQ.1: Sphere or spherical shell EQ.2: Cylinder or cylindrical shell, infinite or finite in length EQ.3: Ellipsoid
CID	Optional local coordinate system ID. If given, all the following input parameters will be interpreted in this coordinate system.
MOVE	Flag to specify whether the region moves: EQ.0: Region is stationary. EQ.1: Region moves to follow the local origin and rotates with the local coordinate system (see CID).

DEFINE_REGION**DEFINE**

Rectangular Prism. Use when TYPE = 0.

Card 3a	1	2	3	4	5	6	7	8
Variable	XMN	XMX	YMN	YMX	ZMN	ZMX		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	0.0	0.0	0.0		

VARIABLE	DESCRIPTION
XMN	Lower <i>x</i> limit of box
YMN	Lower <i>y</i> limit of box
ZMN	Lower <i>z</i> limit of box
XMX	Upper <i>x</i> limit of box
YMX	Upper <i>y</i> limit of box
ZMX	Upper <i>z</i> limit of box

Sphere. Use when TYPE = 1.

Card 3b	1	2	3	4	5	6	7	8
Variable	XC1	YC1	ZC1	RMIN1	RMAX1			
Type	R	R	R	R	R			
Default	0.0	0.0	0.0	0.0	0.0			

VARIABLE	DESCRIPTION
XC1, YC1, ZC1	Coordinates of the center of the sphere
RMIN1, RMAX1	The inner and outer radii of the spherical shell. Set RMIN1 = 0 for a solid sphere

DEFINE**DEFINE_REGION**

Cylinder. Use when TYPE = 2.

Card 3c.1	1	2	3	4	5	6	7	8
Variable	XC2	YC2	ZC2	AX2	AY2	AZ2	RMIN2	RMAX2
Type	F	F	F	F	F	F	F	F
Default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Cylinder. Use when TYPE = 2.

Card 3c.2	1	2	3	4	5	6	7	8
Variable	L2							
Type	F							
Default	∞							

VARIABLE	DESCRIPTION
XC2, YC2, ZC2	A point on the cylindrical axis
AX2, AY2, AZ2	A vector which defines the direction of the axis of the cylinder
RMIN2, RMAX2	The inner and outer radii of the cylindrical shell. Set RMIN2 = 0 for a solid cylinder.
L2	Length of the cylinder. If L2 = 0.0, an infinite cylinder is defined. Otherwise the cylinder has one end at the point (XC2, YC2, ZC2) and the other at a distance L2 along the axis in the direction of the vector (AX2, AY2, AZ2).

DEFINE_REGION**DEFINE**

Ellipsoid. Use when TYPE = 3.

Card 3d.1	1	2	3	4	5	6	7	8
Variable	XC3	YC3	ZC3	AX3	AY3	AZ3	BX3	BY3
Type	F	F	F	F	F	F	F	F
Default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Ellipsoid. Use when TYPE = 3.

Card 3d.2	1	2	3	4	5	6	7	8
Variable	BZ3	RA3	RB3	RC3				
Type	F	F	F	F				
Default	0.0	0.0	0.0	0.0				

VARIABLE	DESCRIPTION
XC3, YC3, ZC3	Coordinates of the center of the ellipsoid
AX3, AY3, AZ3	A vector in the direction of the first axis of the ellipsoid (axis a)
BX3, BY3, BZ3	A vector, $\tilde{\mathbf{b}}$, in the plain of the first and second axes of the ellipsoid. The third axis of the ellipsoid (axis c) will be in the direction of $\mathbf{a} \times \tilde{\mathbf{b}}$ and finally the second axis $\mathbf{b} = \mathbf{c} \times \mathbf{a}$
RA3, RB3, RC3	The semi-axis lengths of the ellipsoid

*DEFINE

*DEFINE_SALEAB_PARAMETERS

*DEFINE_SALEAB_PARAMETERS

Purpose: Serve as one of the component keywords of [*AIRBAG_SALE](#). [*AIRBAG_SALE](#) is a wrapper keyword that the keyword reader internally translates into a series of component keywords. Please refer to the manual page on [*AIRBAG_SALE](#) for a detailed description. This card holds miscellaneous parameters only used for airbag simulations. They are defined in [*AIRBAG_SALE](#) but do not fit into any other existing S-ALE keywords designed for general applications.

The initialization phase generates this keyword along with some others and outputs them into `saleabcvrt.inc`. The converted keywords provide exactly the same parameters and results as the original [*AIRBAG_SALE](#) card. `saleabcvrt.inc` is provided for experienced ALE users to fine-tune certain flags in subsequent runs. However, this keyword should be kept unchanged.

Card 1	1	2	3	4	5	6	7	8
Variable	NABSALE	IUNIT	PATM					
Type	I	I	F					
Default	none	0	none					

Card 2. Define NABSALE cards; one for each original [*AIRBAG_SALE](#) instantiation.

Card 2	1	2	3	4	5	6	7	8
Variable	FSIEXT	FSIINT	FSILCK	CVID	PAIR	TSWT		
Type	I	I	I	I	F	F		
Default	0	0	0	none	none	1.0e16		

VARIABLE	DESCRIPTION
NABSALE	Number of original *AIRBAG_SALE cards
IUNIT	IUNIT in original *AIRBAG_SALE cards
PATM	PATM in original *AIRBAG_SALE cards

VARIABLE	DESCRIPTION
FSIEXT	FSI ID of *ALE_STRUCTURED_FSI_ABEXT card coupling to exterior parts. See Remark 1 .
FSIINT	FSI ID of *ALE_STRUCTURED_FSI_ABINT card coupling to interior parts. See Remark 1 .
FSILCK	FSI ID of *ALE_STRUCTURED_FSI card for preventing contact locking. See Remark 1 .
CVID	ID of *AIRBAG_HYBRID to perform switch if prescribed in the original *AIRBAG_SALE card.
PAIR	PAIR in the original *AIRBAG_SALE card.
TSWT	TSWT in the original *AIRBAG_SALE card.

Remarks:

- FSIEXT, FSIINT, FSILCK.** The conversion process of [*AIRBAG_SALE](#) generates and outputs FSI IDs FSIEXT, FSIINT, and FSILCK. Do not change any of these IDs because these IDs, combined with the [*ALE_STRUCTURED_FSI](#) cards, determine the interior and exterior portions of each airbag in subsequent runs in the absence of the original [*AIRBAG_SALE](#) card.

*DEFINE

*DEFINE_SD_ORIENTATION

*DEFINE_SD_ORIENTATION

Purpose: Define orientation vectors for discrete springs and dampers. These orientation vectors are optional for this element class. Four alternative options are possible. With the first two options, IOP = 0 or 1, the vector is defined by coordinates and is fixed permanently in space. The third and fourth option orients the vector based on the motion of two nodes, so that the direction can change as the line defined by the nodes rotates.

Card	1	2	3	4	5	6	7	8
Variable	VID	IOP	XT	YT	ZT	NID1	NID2	
Type	I	I	F	F	F	I	I	
Default	0	0	0.0	0.0	0.0	0	0	

VARIABLE	DESCRIPTION
VID	Orientation vector ID. A unique ID number must be used.
IOP	Option (see Remark 1): EQ.0: deflections/rotations are measured and forces/moments applied along the following orientation vector. EQ.1: deflections/rotations are measured and forces/moments applied along the axis between the two spring/damper nodes projected onto the plane normal to the following orientation vector. EQ.2: deflections/rotations are measured and forces/moments applied along a vector defined by the following two nodes. EQ.3: deflections/rotations are measured and forces/moments applied along the axis between the two spring/damper nodes projected onto the plane normal to the a vector defined by the following two nodes.
XT	<i>x</i> -value of orientation vector. Define if IOP = 0 or 1.
YT	<i>y</i> -value of orientation vector. Define if IOP = 0 or 1.
ZT	<i>z</i> -value of orientation vector. Define if IOP = 0 or 1.
NID1	Node 1 ID. Define if IOP = 2 or 3.

VARIABLE	DESCRIPTION
NID2	Node 2 ID. Define if IOP = 2 or 3.

Remarks:

1. **Orientation Vectors.** The orientation vectors defined by options 0 and 1 are fixed in space for the duration of the simulation. Options 2 and 3 allow the orientation vector to change with the motion of the nodes. Generally, the nodes should be members of rigid bodies, but this is not mandatory. When using nodes of deformable parts to define the orientation vector, care must be taken to ensure that these nodes will not move past each other. If this happens, the direction of the orientation vector will immediately change with the result that initiate severe instabilities can develop.

*DEFINE

*DEFINE_SET_ADAPTIVE

*DEFINE_SET_ADAPTIVE

Purpose: To control the adaptive refinement level by element or part set.

Card 1	1	2	3	4	5	6	7	8
Variable	SETID	STYPE	ADPLVL	ADPSIZE				
Type	I	I	I	F				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
SETID	Element set ID or part set ID
STYPE	Set type for SETID: EQ.1: element set EQ.2: part set
ADPLVL	Adaptive refinement level for all elements in SETID set.
ADPSIZE	Minimum element size to be adapted based on element edge length for all elements in SETID set.

Remarks:

1. **Element Restrictions.** This option is for 3D-shell *h*-adaptivity only at the present time.
2. **Keyword Priority.** Three keywords control refinement level: *CONTROL_ADAPTER, *DEFINE_BOX_ADAPTER, and *DEFINE_SET_ADAPTER. *CONTROL_ADAPTER must be included in the keyword deck for any adaptivity to be done. If all three keywords are defined in a keyword deck, *DEFINE_BOX_ADAPTER defines the refinement within the box; it unlike the other two keywords does not control the minimum element size to be adapted. *DEFINE_SET_ADAPTER has the next highest priority for defining refinement level while *CONTROL_ADAPTER has the least priority. This keyword is useful for defining different refinement levels for different parts.
3. **Multiple Refinement Definitions.** If there are multiple definitions of refinement level or element size for any elements, the latter one will be used.

DEFINE_SPG_TO_SURFACE_COUPLING**DEFINE*****DEFINE_SPG_TO_SURFACE_COUPLING**

Purpose: Define a coupling interface between smoothed particle Galerkin (SPG) particles and a surface.

Card 1	1	2	3	4	5	6	7	8
Variable	SPGP	SURF	SPTYPE	SFTYPE				
Type	I	I	I	I				
Default	none	none	1	0				

Card 2	1	2	3	4	5	6	7	8
Variable	SBC	FS	FD	DC	SFP	THK	FRQ	
Type	I	F	F	F	F	F	I	
Default	0	0.	FS	0.	0.1	0.5	50	

VARIABLE	DESCRIPTION
SPGP	Part ID/Part set ID for the SPG particles
SURF	Segment set ID specifying the surface
SPTYPE	Type for SPGP: EQ.0: Part set ID EQ.1: Part ID
SFTYPE	Type for SURF: EQ.0: Segment set ID
SBC	Type of boundary condition: EQ.0: Free-slip boundary EQ.1: Non-slip boundary (tied after contact)

DEFINE**DEFINE_SPG_TO_SURFACE_COUPLING**

VARIABLE	DESCRIPTION
FS	Static coefficient of friction. If SBC = 1, then FS is not used.
FD	Dynamic coefficient of friction. It is set to the value of FS by default.
DC	Exponential decay of the friction coefficient
SFP	Scaling factor of the penetration stiffness coefficient along the normal direction of the contact interface. Default value is 0.1.
THK	Thickness scaling factor for contact search. During initialization, LS-DYNA searches for the particle that is closest to the surface. The distance of this particle from the surface at the beginning of the simulation is scaled by THK. This scaled distance is used as the contact thickness for the surface's segments. The default value of THK is 0.5.
FRQ	Bucket sorting frequency. Default is 50.

***DEFINE_SPH_ACTIVE_REGION**

Purpose: The purpose of this keyword is to increase the efficiency of the SPH method's neighborhood search algorithm by specifying an *active region*. All SPH elements located outside of the active region are deactivated. This card supports active regions consisting of the volume bounded by two closed surfaces (boxes, centered cylinders, and centered spheres are currently supported). Once the SPH particle is deactivated, it will stay inactive.

Card Summary:

Card 1. This card is required.

ID	TYPE	STYPE	CYCLE	NID	ICID	IBUFF	

Card 2a.1. Include this card if STYPE = 0.

XIMIN	YIMIN	ZIMIN	XIMAX	YIMAX	ZIMAX		

Card 2a.2. Include this card if STYPE = 0.

XOMIN	XOMIN	ZOMIN	XOMAX	YOMAX	ZOMAX		

Card 2b.1. Include this card if STYPE = 1.

X0	Y0	Z0	XH	YH	ZH		

Card 2b.2. Include this card if STYPE = 1.

RMIN	ZMIN	RMAX	ZMAX				

Card 2c.1. Include this card if STYPE = 2.

X0	Y0	Z0					

Card 2c.2. Include this card if STYPE = 2.

RMIN	RMAX						

DEFINE**DEFINE_SPH_ACTIVE_REGION****Data Card Definitions:**

Card 1	1	2	3	4	5	6	7	8
Variable	ID	TYPE	STYPE	CYCLE	NID	ICID	IBUFF	
Type	I	I	I	I	I	I	I	
Default	none	0	0	1	0	0	0	

VARIABLE	DESCRIPTION
ID	Part Set ID/Part ID
TYPE	EQ.0: Part set EQ.1: Part
STYPE	Type of the region. EQ.0: Rectangular box EQ.1: Cylinder EQ.2: Sphere
CYCLE	Number of cycles between each check
NID	Referential nodal ID. The SPH box will move with this node.
ICID	Local coordinate system ID
IBUFF	Buffer zone flag (only used when STYPE = 0): EQ.0: Particles on the edge of the outer box do not get any special treatment. EQ.1: Particles on the edge of the outer box are frozen in space and act as neighbors for active particles inside the box. This option is mainly used for fluid simulations to prevent the fluid from spilling out of the activation box.

DEFINE_SPH_ACTIVE_REGION**DEFINE**

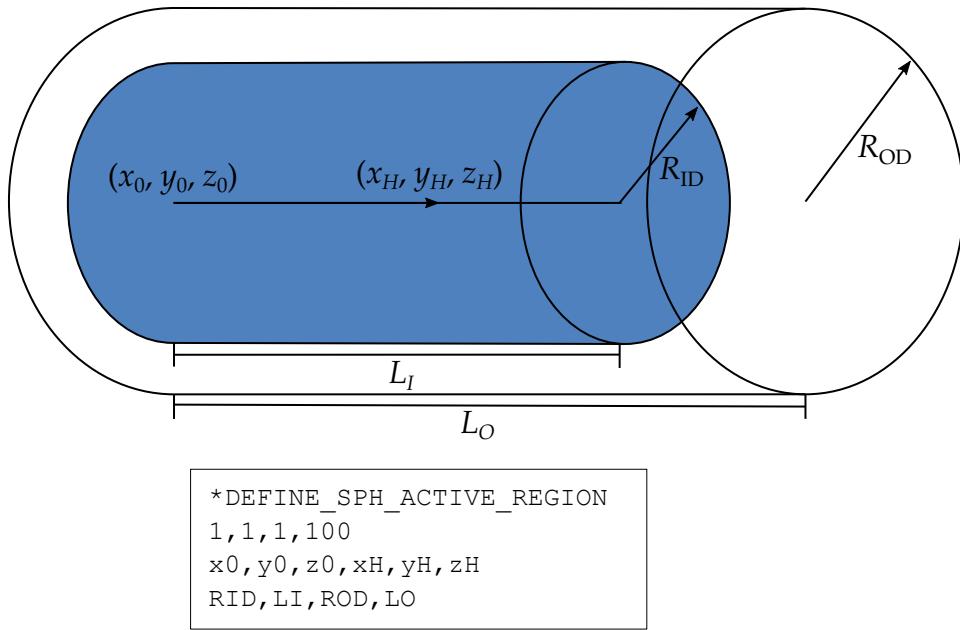
Interior Rectangular Box Card. This card is included if STYPE = 0.

Card 2a.1	1	2	3	4	5	6	7	8
Variable	XIMIN	YIMIN	ZIMIN	XIMAX	YIMAX	ZIMAX		
Type	F	F	F	F	F	F		
Default	none	none	none	None	none	none		

Outer Rectangular Box Card. This card is included if STYPE = 0.

Card 2a.2	1	2	3	4	5	6	7	8
Variable	XOMIN	XOMIN	ZOMIN	XOMAX	YOMAX	ZOMAX		
Type	F	F	F	F	F	F		
Default	none	none	none	none	none	none		

VARIABLE	DESCRIPTION
XIMIN, YIMIN, ZIMIN	Minimum x, y, z coordinate of the inner box
XIMAX, YIMAX, ZIMAX	Maximum x, y, z coordinates of the inner box
XOMIN, YOMIN, ZOMIN	Minimum x, y, z coordinate of the outer box
XOMAX, YOMAX, ZOMAX	Maximum x, y, z coordinates of the outer box

DEFINE**DEFINE_SPH_ACTIVE_REGION****Figure 17-60.** Example DEFINE_SPH_ACTIVE_REGION

Cylinder Axis Card. This card is included if STYPE = 1.

Card 2b.1	1	2	3	4	5	6	7	8
Variable	X0	Y0	Z0	XH	YH	ZH		
Type	F	F	F	F	F	F		
Default	none	none	none	none	none	none		

Cylinder Radii Card. This card is included if STYPE = 1.

Card 2b.2	1	2	3	4	5	6	7	8
Variable	RMIN	ZMIN	RMAX	ZMAX				
Type	F	F	F	F				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
X0, Y0, Z0	Coordinates of the center of the cylinder base. The nested cylinders share the same starting base plane. This point also serves as the tail for the vector specifying the direction of the cylinders' axis. See Figure 17-60 .
XH, YH, ZH	Coordinates for the head of the cylinders' axial direction vector.
RMIN, ZMIN	Radius and length of the interior cylinder.
RMAX, ZMAX	Radius and length of the outer cylinder.

Center of Sphere Card. This card is included if STYPE = 2.

Card 2c.1	1	2	3	4	5	6	7	8
Variable	X0	Y0	Z0					
Type	F	F	F					
Default	none	none	none					

Sphere Radii Card. This card is included if STYPE = 2.

Card 2c.2	1	2	3	4	5	6	7	8
Variable	RMIN	RMAX						
Type	F	F						
Default	none	none						

VARIABLE	DESCRIPTION
X0, Y0, Z0	The spheres' center.
RMIN	Radius of the interior sphere
RMAX	Radius of the outer sphere

*DEFINE

*DEFINE_SPH_AMBIENT_DRAG

*DEFINE_SPH_AMBIENT_DRAG

Purpose: Add drag forces to SPH particles that use the implicit incompressible formulation (FORM = 13 in *CONTROL_SPH). For a wading simulation, this keyword sets air properties.

Card 1	1	2	3	4	5	6	7	8
Variable	ICPL	VX	VY	VZ	RHOA	MUA	SFTENS	
Type	I	F	F	F	F	F	F	
Default	0	0.0	0.0	0.0	none	none	none	

VARIABLE	DESCRIPTION
ICPL	Coupling with ICFD: EQ.0: No coupling EQ.1: Import ambient velocity field from a steady-state CFD analysis in a profile file format. Provide the file name in Card 2. See Example 1 below for a sample profile file.
VX, VY, VZ	Velocity components of the ambient material
RHOA	Density of the ambient material
MUA	Viscosity of the ambient material
SFTENS	Surface tension coefficient for the interface between the SPH fluid and ambient material

ICPL Card. Include this card if the ICPL = 1.

Card 2	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	A60							

VARIABLE	DESCRIPTION
FILENAME	Name of the file containing steady-state CFD data for the ambient air velocity field.

Example 1:

The following example illustrates the profile file format, which can be exported directly by Fluent. Note that the first line is empty.

```
[Name]
cfд_result

[Data]
    x,           y,           z,           vx,          vy,          vz
  0.1,         0.0,         0.0,         1.0,         0.0,         0.0
  0.1,         0.0,         0.1,         1.0,         0.0,         0.0
  0.1,         0.1,         0.1,         1.0,         0.0,         0.0
  0.1,         0.1,         0.0,         1.0,         0.0,         0.0
  0.2,         0.0,         0.0,         0.9,         0.0,         0.0
  0.2,         0.0,         0.1,         0.9,         0.0,         0.0
  0.2,         0.1,         0.1,         0.9,         0.0,         0.0
  0.2,         0.1,         0.0,         0.85,        0.0,         0.0
```

*DEFINE

*DEFINE_SPH_DE_COUPLING

*DEFINE_SPH_DE_COUPLING_{OPTION}

Purpose: Define a penalty based contact. This option is to be used for the node to node contacts to couple the SPH solver and the discrete element sphere (DES) solver.

The available options include:

<BLANK>

ID

ID Card. Additional card for ID keyword option.

Card ID	1	2	3	4	5	6	7	8
Variable	DID				HEADING			
Type	I				A80			
Default	none				none			

SPH Part Cards. Include as many of this card as desired. Input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	SPHID	DESID	SPHTYP	DESTYP	PFACT	DFACT	SPHBOX	
Type	I	I	I	I	F	F	I	
Default	none	none	none	none	1.0	0.	none	

VARIABLE

DESCRIPTION

DID Definition ID. This must be a unique number.

HEADING Definition descriptor. It is suggested that unique descriptions be used.

SPHID SPH part or part set ID.

DESID DES part or part set ID.

VARIABLE	DESCRIPTION
SPHTYP	SPH part type: EQ.0: Part set ID, EQ.1: Part ID.
DESTYP	DES part type: EQ.0: Part set ID, EQ.1: Part ID.
PFACT	Penalty scale factor
DFACT	Penalty scale factor for contact damping coefficient
SPHBOX	BOX ID for SPH parts, See Remarks.

Remarks:

SPHBOX is used to define the box IDs for the SPH parts. Only the particles inside the boxes are defined for the node to node contacts.

DEFINE**DEFINE_SPH_INJECTION*****DEFINE_SPH_INJECTION**

Purpose: Inject SPH elements from user defined grid points.

Card 1	1	2	3	4	5	6	7	8
Variable	PID	NSID	CID	VX	VY	VZ	AREA	VMAG
Type	I	I	I	F	F	F	F	I
Default	none	none	none	0.0	0.0	0.0	0.0	0

Card 2	1	2	3	4	5	6	7	8
Variable	TBEG	TEND	NID	RADIUS	XLEN	YLEN	PSIZE	
Type	F	F	I	F	F	F	F	
Default	0.0	10^{20}	0	0.0	0.0	0.0	0.0	

VARIABLE	DESCRIPTION
PID	Part ID of newly generated SPH elements.
NSID	Node set ID. Nodes are used for initial injection position for the SPH elements. Ignored if RADIUS, XLEN or YLEN are defined below.
CID	Local coordinate system ID; see *DEFINE_COORDINATE_SYSTEM for example. The local <i>x</i> and <i>y</i> -directions define the injection plane, while the local <i>z</i> -direction defines the normal to the injection plane. A CID is always required. This coordinate system is used for the definition of injection velocity (VX,VY,VZ) as well.
VX, VY, VZ	Velocity of the injected particles: $\mathbf{v} = (VX, VY, VZ)$, defined in the local coordinate system from CID.
AREA	The area of initial injection surface. The density of injection flow comes from the material models; see *MAT definition. This parameter can be left blank if the nodes provided in NSID are uniformly

VARIABLE	DESCRIPTION
	distributed, or if RADIUS, XLEN or YLEN are defined below, in which case AREA will be automatically computed.
VMAG	Injected particle velocity multiplier: GT.0: The velocity of the injected particles is multiplied by VMAG. LT.0: $ VMAG $ is a curve ID defining the magnitude of the velocity vector with respect to time, for variable injection speed.
TBEG	Birth time
TEND	Death time
NID	Optional Node ID. If defined, the injection plane follows the motion of this node.
RADIUS	If this option is nonzero, the node set provided in NSID is ignored. Instead, this parameter defines the radius of a circular injection surface centered at the origin of the coordinate system defined in CID, using the local Z-axis of CID as normal to the injection plane. Particles are uniformly generated with a spacing of PSIZE.
XLEN	If this option is nonzero and RADIUS is zero, the node set provided in NSID is ignored. Instead, a rectangular injection surface is automatically created, centered at the origin of the coordinate system defined in CID, using the local Z-axis of CID as normal to the injection plane. This rectangular injection has dimensions of XLEN and YLEN in the local coordinate system defined by CID. Particles are uniformly generated with a spacing of PSIZE.
YLEN	See XLEN.
PSIZE	If RADIUS or XLEN / YLEN is defined, the particles are automatically placed on the defined injection geometry, with a spacing of PSIZE.

*DEFINE

*DEFINE_SPH_INJECTION_SIMPLIFIED

*DEFINE_SPH_INJECTION_SIMPLIFIED_{OPTION1}

For *OPTION1*, available options are:

<BLANK>

FLOWRATE

Purpose: Inject SPH elements like *DEFINE_SPH_INJECTION but only for a circular injection area. This keyword has simplified syntax compared to *DEFINE_SPH_INJECTION for this geometry.

Card Summary:

Card 1a. Include this card when not using the keyword option (<BLANK>).

PID	VEL	RADIUS	PSIZE	OFFST	CID	NID	
-----	-----	--------	-------	-------	-----	-----	--

Card 1b. Include this card when using the FLOWRATE keyword option.

PID	FLWRT	RADIUS	PSIZE	OFFST	CID	NID	
-----	-------	--------	-------	-------	-----	-----	--

Card 2. This card is required.

XC	YC	ZC	DNX	DNY	DNZ	TBEG	TEND
----	----	----	-----	-----	-----	------	------

Data Card Definitions:

Include this card when not using the keyword option (<BLANK>).

Card 1a	1	2	3	4	5	6	7	8
Variable	PID	VEL	RADIUS	PSIZE	OFFST	CID	NID	
Type	I	F	F	F	F	I	I	
Default	none	none	none	0.0	0.0	0	0	

VARIABLE	DESCRIPTION
PID	Part ID of newly generated SPH elements
VEL	Injected particle velocity in the injection direction: GT.0.0: The velocity of the injected particles is constant, equal to

VARIABLE	DESCRIPTION
	VMAG.
	LT.0.0: VMAG is a curve ID defining the injection velocity with respect to time for variable injection speed.
RADIUS	Radius of the circular injection surface. Particles are uniformly generated across the surface at a spacing of PSIZE.
PSIZE	Spacing between particles on the injection surface
OFFST	Option to offset the centroid of the injection surface, (XC, YC, ZC), along the normal vector, (DNX, DNY, DNZ), by a distance of OFFST. Only valid when CID is zero.
CID	Local coordinate system ID; see *DEFINE_COORDINATE_SYSTEM for example. The origin of this local coordinate system defines the centroid of the injection area, and the local z-direction defines the normal to the injection area, which is the direction in which particles will be injected.
NID	Optional Node ID. If defined, the injection plane follows the motion of this node.

Include this card when using the FLOWRATE keyword option.

Card 1b	1	2	3	4	5	6	7	8
Variable	PID	FLWRT	RADIUS	PSIZE	OFFST	CID	NID	
Type	I	F	F	F	F	I	I	
Default	none	none	none	0.0	0.0	0	0	

VARIABLE	DESCRIPTION
PID	Part ID of newly generated SPH elements
FLWRT	Volumetric flow rate of injection: GT.0.0: The volumetric flow rate is constant, equal to FLWRT. LT.0.0: FLWRT is a curve ID defining the volumetric flow rate with respect to time for variable injection flow rate.

DEFINE**DEFINE_SPH_INJECTION_SIMPLIFIED**

VARIABLE	DESCRIPTION
RADIUS	Radius of the circular injection surface. Particles are uniformly generated across the surface at a spacing of PSIZE.
PSIZE	Spacing between particles on the injection surface
OFFST	Option to offset the centroid of the injection surface, (XC, YC, ZC), along the normal vector, (DNX, DNY, DNZ), by a distance of OFF-ST. Only valid when CID is zero.
CID	Local coordinate system ID; see *DEFINE_COORDINATE_SYSTEM for example. The origin of this local coordinate system defines the centroid of the injection area, and the local z-direction defines the normal to the injection area, which is the direction in which particles will be injected.
NID	Optional Node ID. If defined, the injection plane follows the motion of this node.

Card 2	1	2	3	4	5	6	7	8
Variable	XC	YC	ZC	DNX	DNY	DNZ	TBEG	TEND
Type	F	F	I	F	F	F	F	F
Default	0.0	10^{20}	0	0.0	0.0	0.0	0.0	10^{20}

VARIABLE	DESCRIPTION
XC,YZ,ZC	Centroid of the injection surface. Ignored if CID is nonzero.
DNX, DNY, DNZ	Components of the normal vector to the injection surface. Ignored if CID is nonzero.
TBEG	Birth time
TEND	Death time

DEFINE_SPH_MASSFLOW_PLANE**DEFINE*****DEFINE_SPH_MASSFLOW_PLANE**

Purpose: To measure SPH mass flow rate across a defined plane. See also the accompanying keyword *DATABASE_SPHMASSFLOW which controls the output frequency.

Card 1	1	2	3	4	5	6	7	8
Variable	PRTCLSID	SURFSID	PTYPE	STYPE				
Type	I	I	I	I				
Default	0	0	0	0				

VARIABLE**DESCRIPTION**

PRTCLSID Node set ID, node ID, part set ID or part ID specifying the SPH particles to be measured. PTYPE below indicates the ID type specified by PRTCLSID.

SURFSID Part set ID or part ID defining the surface across which the flow rate is measured. STYPE below indicates the ID type specified by SURFSID.

PTYPE PRTCLSID type:
EQ.0: Node set
EQ.1: Node
EQ.2: Part set
EQ.3: Part

STYPE SURFSID type:
EQ.0: Part set
EQ.1: Part

*DEFINE

*DEFINE_SPH_MESH_BOX

*DEFINE_SPH_MESH_BOX

Purpose: Generate SPH particles inside a given box.

Card 1	1	2	3	4	5	6	7	8
Variable	XMIN	YMIN	ZMIN	XLEN	YLEN	ZLEN		
Type	F	F	F	F	F	F		
Default	none	none	none	none	none	none		

Card 2	1	2	3	4	5	6	7	8
Variable	IPID	NX	NY	NZ		IDSEG	SFSP	
Type	I	I	I	I		I	F	
Default	none	none	none	none		0	1.0	

VARIABLE	DESCRIPTION
XMIN	Minimum <i>x</i> -coordinate
YMIN	Minimum <i>y</i> -coordinate
ZMIN	Minimum <i>z</i> -coordinate
XLEN	Box length in the <i>x</i> -direction
YLEN	Box length in the <i>y</i> -direction
ZLEN	Box length in the <i>z</i> -direction
IPID	Part ID for generated SPH elements
NX	Number of SPH particles in the <i>x</i> -direction
NY	Number of SPH particles in the <i>y</i> -direction

VARIABLE	DESCRIPTION
NZ	Number of SPH particles in the z-direction
IDSEG	Segment set ID that can be used to removed generated SPH elements. segment set is used to split the box into two regions, one that has SPH elements and one without SPH (see Remark 2). The sign of IDSEG determines which region keeps the SPH elements. Also, to avoid sudden movement, elements that are “too close” to the segment set will be removed, regardless of the sign of IDSEG. Too close means the normal distance from the center of the SPH element to the nearest segment is smaller than the SPH smoothing length scaled by SFSP. EQ.0: No generated elements are removed. GT.0: Keep the SPH element if it lies nominally in the normal direction of the segments in the segment set. LT.0: Keep the SPH element if it lies nominally in the reverse normal direction of segments in the segment set.
SFSP	Scale factor for SPH smoothing length (active only when IDSEG ≠ 0). If the normal distance between an SPH particle and the nearest segment is smaller than this distance, the SPH element is removed.

Remarks:

1. **Interparticle Distances.** Interparticle distances are calculated as XLEN/NX, YLEN/NY, and ZLEN/NZ
2. **Region for Element Removal.** The region created by the segment set, IDSEG, must either define a closed volume in the box or intersect the box. If it defines an open surface in the box, the elements that are to be kept cannot be uniquely determined.

DEFINE**DEFINE_SPH_MESH_OBJ*****DEFINE_SPH_MESH_OBJ**

Purpose: Import an OBJ file whose geometry will follow a rigid body. Also include the OBJ geometry in d3plot and (if requested) ISPHFOR output files.

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

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

VARIABLE	DESCRIPTION
FILENAME	File name of OBJ file to be included
PVID	Parent part ID. The OBJ geometry will follow the motion of the rigid part with ID PVID.
PID	Optional part ID to identify this OBJ file in d3plot and ISPHFOR outputs. An ID will be automatically generated if this parameter is left blank.

Remarks:

When this keyword is included in the input deck, DCOMP = 5 is automatically set for *DATABASE_EXTENT_BINARY.

DEFINE_SPH_MESH_SURFACE**DEFINE*****DEFINE_SPH_MESH_SURFACE**

Purpose: Generate and place SPH elements on the surface of triangular shell elements.

Card 1	1	2	3	4	5	6	7	8
Variable	SID	TYPE	SPHPID	SPHXID	NSID	SPACE	IOUT	
Type	I	I	I	I	I	F	I	
Default	none	none	Rem 1	Rem 1	0	none	0	

VARIABLE	DESCRIPTION
SID	Part or part set ID for the region of the mesh upon which the SPH elements will be placed
TYPE	SID type: EQ.0: part set ID EQ.1: part ID
SPHPID	Part ID for generated SPH elements
SPHXID	Section ID for generated SPH elements
NSID	If defined, this card creates a node set with ID NSID (see *SET_NODE) for the nodes generated by this card.
SPACE	Maximum space between SPH elements
IOUT	Keyword file output: EQ.0: no output (default) EQ.1: output generated nodes, SPH elements and node set to a keyword file with SPH_surface_ prefix.

Remarks:

1. **SPHPID and SPHXID.** Part ID and/or section ID will be generated by this card if they are not provided in the input.

DEFINE**DEFINE_SPH_TO_SPH_COUPLING*****DEFINE_SPH_TO_SPH_COUPLING**

Purpose: Define a penalty-based, node-to-node contact for particles of SPH parts. Note that this contact type is an alternative to inter-part particle interaction by “particle approximation;” see the field CONT in *CONTROL_SPH and the INTERACTION option of *SECTION_SPH.

Card Sets: Each set consists of a Card 1 and may include an additional Card 2. Unless the card following Card 1 contains an “&” in its first column, the optional card is not read. Provide as many sets as necessary. This input terminates at the next keyword (“*”) card.

Card 1	1	2	3	4	5	6	7	8
Variable	SIDA	SIDB	SATYP	SBTYP	IBOXA	IBOXB	PFACT	SRAD
Type	I	I	I	I	I	I	F	F
Default	none	none	none	none	none	none	1.0	1.0

Optional. The keyword reader identifies this card by an “&” in the first column.

Card 2	1	2	3	4	5	6	7	8
Variable	DFACT	ISOFT						
Type	F	I						
Default	0.0	0						

VARIABLE	DESCRIPTION
SIDA	Part or part set ID for one set of particles in the contact
SIDB	Part or part set ID for the other set of particles in the contact
SATYP	SIDA part type: EQ.0: Part set ID EQ.1: Part ID

VARIABLE	DESCRIPTION
SBTYP	SIDB part type: EQ.0: Part set ID EQ.1: Part ID
IBOXA	Box ID for the A parts; see Remark 1 .
IBOXB	Box ID for the B parts; see Remark 1 .
PFACT	Penalty scale factor; see Remark 2 .
SRAD	Scale factor for nodes to nodes contact criteria; see Remark 3 .
DFACT	Penalty scale factor for contact damping coefficient; see Remark 4 .
ISOFT	Soft constraint option: EQ.0: Penalty formulation (default) EQ.1: Soft constraint formulation The soft constraint may be necessary if the material constants of the parts in contact have a wide variation in the elastic bulk moduli. In the soft constraint option, the interface stiffness is based on the nodal mass and the global time step size.

Remarks:

- Contact Particles.** IBOXA and IBOXB are used to define the box IDs for the A parts and the B parts, respectively. Only the particles inside the boxes are defined for the node to node contacts.
- Impact Velocity.** For High Velocity Impact problems, a smaller value ranging from 0.01 to 10^{-4} for the PFACT field is recommended. A value ranging from 0.1 to 1 is recommended for low velocity contact between two SPH parts.
- Contact Detection.** Contact between two SPH particles from different parts is detected when the distance of two SPH particles I and J is less than $SRAD \times (h_I + h_J)/2.0$. If $SRAD < 0$, the contact distance is based on the nodal volumes instead of the smoothing lengths: $SRAD \times (\sqrt[3]{V_I} + \sqrt[3]{V_J})/2.0$.
- DFACT.** The default value, DFACT = 0.0, is recommended. For DFACT > 0.0, interaction between SPH parts includes a viscous effect, providing some stickiness similar to the particle approximation invoked when CONT = 0 in *CONSTANT.

DEFINE**DEFINE_SPH_TO_SPH_COUPLING**

TROL_SPH. At present, no recommendation can be given for a value of DFACT other than the value should be less than 1.0.

***DEFINE_SPH_VICINITY_SENSOR**

Purpose: Measure the mass of specified SPH particles in the vicinity of a surface. The list of all particles flagged as within the vicinity of the surface is exported along with time at which particle moved within the vicinity. See also the accompanying keyword *DATABASE_SPHVICINITY which controls the output frequency.

Card 1	1	2	3	4	5	6	7	8
Variable	PRTCLSID	SURFSID	PTYPE	STYPE	DIST			
Type	I	I	I	I	F			
Default	0	0	0	0	0.0			

VARIABLE	DESCRIPTION
PRTCLSID	Node set ID, node ID, part set ID or part ID specifying the checked SPH particles. PTYPE below indicates the ID type specified by PRTCLSID.
SURFSID	Part set ID or part ID defining the surface. STYPE below indicates the ID type specified by SURFSID.
PTYPE	PRTCLSID type: EQ.0: Node set EQ.1: Node EQ.2: Part set EQ.3: Part
STYPE	SURFID type: EQ.0: Part set EQ.1: Part
DIST	Distance criteria. Any particle closer than this distance to the surface is considered in the vicinity.

*DEFINE

*DEFINE_SPOTWELD_FAILURE

*DEFINE_SPOTWELD_FAILURE_{OPTION}

The available options are

<BLANK>

ADD

PID

Purpose: Define spot weld failure data for the failure criterion developed by Lee and Balur (2011). This card provides the failure data for OPT = 10 on *MAT_SPOTWELD. It is available for spot welds consisting of beam elements, solid elements, or solid assemblies.

*DEFINE_SPOTWELD_FAILURE requires that the weld nodes be tied to shell elements using tied constraint based contact options:

1. For beam element welds, only *CONTACT_SPOTWELD is valid.
2. For solid element welds or solid assembly welds, valid options are the following:

*CONTACT_TIED_SURFACE_TO_SURFACE

*CONTACT_SPOTWELD

*CONTACT_TIED_SHELL_EDGE_TO_SURFACE

Other tied contact types cannot be used.

The ADD keyword option adds materials to a previously defined spot weld failure data set. The PID option changes the Card 3 input to use part set ID's rather than material ID's.

Card Summary:

Card 1. This card is required.

ID	TFLAG	DC1	DC2	DC3	DC4	EXN	EXS
----	-------	-----	-----	-----	-----	-----	-----

Card 2. This card is included unless the ADD option is used.

NAVG	D_SN	D_SS	R_SULT	TSCALE			
------	------	------	--------	--------	--	--	--

Card 3a. This card is included if the PID option is used. Include one card for each pair of weld parts associated with the data set. The next keyword ("*") card terminates the keyword.

PID1	PID2	SN	SS					
------	------	----	----	--	--	--	--	--

Card 3b. This card is included if the PID option is *not* used. Include one card for each material associated with the data set. The next keyword ("*") card terminates the keyword.

MID	SN	SS						
-----	----	----	--	--	--	--	--	--

Data Card 1. This card contains the data set's ID and the first 7 parameters. When the ADD option is active *leave the 7 parameters blank*.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	TFLAG	DC1	DC2	DC3	DC4	EXN	EXS
Type	I	I	F	F	F	F	F	F
Default	none	0	1.183	0.002963	0.0458	0.1	2.0	2.0

VARIABLE	DESCRIPTION
ID	Identification number of data set, input as FVAL on *MAT_SPOTWELD
TFLAG	Thickness flag for nominal stress calculation EQ.0: Use minimum sheet thickness EQ.1: Use average sheet thickness EQ.2: Use maximum sheet thickness EQ.3: Use sum of sheet thicknesses
DC1	Dynamic coefficient, c_1
DC2	Dynamic coefficient, c_2
DC3	Dynamic coefficient, c_3
DC4	Dynamic coefficient, c_4

DEFINE**DEFINE_SPOTWELD_FAILURE**

VARIABLE	DESCRIPTION
EXN	Exponent on the normal term, n_n
EXS	Exponent on the shear term, n_s

Data Card 2. This card contains 3 spot weld failure data parameters. *Do not include this card when the ADD option is active*

Card 2	1	2	3	4	5	6	7	8
Variable	NAVG	D_SN	D_SS	R_SULT	TSCALE			
Type	I	F	F	F	F			
Default	0	0.0	0.0	Rem 2	1.0			

VARIABLE	DESCRIPTION
NAVG	Number of points in the time average of the load rates. See Remark 4 .
D_SN	Reference value of the static normal strength, $S_{n,\text{stat}}$. See Remark 2 .
D_SS	Reference value of the static shear strength, $S_{s,\text{stat}}$. See Remark 2 .
R_SULT	Reference ultimate strength. See Remark 2 .
TSCALE	Scale factor for thickness used in nominal stress calculations

Material-Specific Strength Data Cards with PID option. Include one card for each pair of weld parts associated with the data set. The next keyword ("*") card terminates the keyword.

Card 3a	1	2	3	4	5	6	7	8
Variable	PID1	PID2	SN	SS				
Type	I	I	F	F				
Default	none	none	Rem 2	Rem 2				

VARIABLE	DESCRIPTION
PID1	Part ID of welded shell part
PID2	Part ID of part welded to PID1
SN	Static normal strength of material MID. $S_{n,\text{stat}}$. See Remark 2 .
SS	Static shear strength of material MID, $S_{s,\text{stat}}$. See Remark 2 .

Material-Specific Strength Data Cards without PID option. Include one card for each material associated with the data set. The next keyword ("*") card terminates the keyword.

Card 3b	1	2	3	4	5	6	7	8
Variable	MID	SN	SS					
Type	I	F	F					
Default	none	Rem 2	Rem 2					

VARIABLE	DESCRIPTION
MID	Material ID number of welded shell material
SN	Static normal strength of material MID. $S_{n,\text{stat}}$. See Remark 2 .
SS	Static shear strength of material MID, $S_{s,\text{stat}}$. See Remark 2 .

Remarks:

1. **Failure Model.** This stress based failure model, which was developed by Yung-Li Lee and Santosh Balur (2011) of FCA, compares nominal stress to dynamical strengths for the weld. The weld fails when the stresses are outside of the failure surface defined as

$$\left(\frac{s_n}{S_{n,\text{dyn}}}\right)^{n_n} + \left(\frac{s_s}{S_{s,\text{dyn}}}\right)^{n_s} = 1 ,$$

where s_n and s_s are nominal stresses in the normal and tangential directions, $S_{n,\text{dyn}}$ and $S_{s,\text{dyn}}$ are dynamical strengths in the normal and tangential directions, and n_n and n_s are the exponential factors EXN and EXS, respectively. The nominal stresses are defined as

$$s_n = \frac{P_n}{Dct}, \quad s_s = \frac{P_s}{Dct},$$

where P_n and P_s are the loads carried by the weld in the normal and tangential directions, D is the weld diameter, c is scale factor TSCALE, and t is the thickness of the welded sheets calculated according to the value of TFLAG. The dynamical strength terms in the denominator are load-rate dependent and are derived from static strength:

$$S_{n,dyn} = S_{n,stat} \left[c_1 + c_2 \left(\frac{\dot{P}_n}{c_4} \right) + c_3 \log \left(\frac{\dot{P}_n}{c_4} \right) \right]$$

$$S_{s,dyn} = S_{s,stat} \left[c_1 + c_2 \left(\frac{\dot{P}_s}{c_4} \right) + c_3 \log \left(\frac{\dot{P}_s}{c_4} \right) \right]$$

where the constants c_1 to c_4 are the input in the fields DC1 to DC4, \dot{P}_n and \dot{P}_s are the load rates, and $S_{n,stat}$ and $S_{s,stat}$ are the static strengths of the welded sheet materials (see [Remark 2](#)).

2. **Static Strengths.** When two different materials are welded, the material having the smaller normal strength (SN) determines the static strengths used for the weld. Materials that do not have SN and SS values default to D_SN and D_SS from Card 2.

If R_SULT is defined on Card 2 and the PID keyword option is not used, then D_SN and D_SS are interpreted to be reference values of the normal and shear static strength, and the SN field on Card 3 is interpreted as a material specific ultimate strength. These values are then used to calculate material specific strength values by

$$S_{n,stat} = S_{n,ref} \left(\frac{S_u}{S_{u,ref}} \right), \quad S_{s,stat} = S_{s,ref} \left(\frac{S_u}{S_{u,ref}} \right)$$

where $S_{n,ref}$, $S_{s,ref}$, and $S_{u,ref}$, are D_SN, D_SS, and R_SULT on Card 2, and S_u is SN on Card 3. The SS values are ignored.

If the PID keyword option is used, then R_SULT is ignored, and SN and SS are static strength values.

3. **Default Values.** The default values for DC1 to DC4, and EXN and EXS are based on the work Chao, Wang, Miller and Zhu (2010). and Wang, Chao, Zhu, and Miller (2010). These parameters are unitless except for DC4 which has units of force per unit time. The default value of 0.1 has units of MN/sec.
4. **Load Rate.** The load rate, \dot{P} , can be time averaged to reduce the effect of high frequency oscillations on the dynamic weld strength. NAVG is the number of terms in the time average.

DEFINE_SPOTWELD_FAILURE_RESULTANTS**DEFINE*****DEFINE_SPOTWELD_FAILURE_RESULTANTS**

Purpose: Define failure criteria between part pairs for predicting spot weld failure. This table is implemented for *solid* element spot welds, which are used with the tied, constraint based, contact option: *CONTACT_TIED_SURFACE_TO_SURFACE. *Note that other tied contact types cannot be used.* The input in this section continues until then next “*” card is encountered. Default values are used for any part ID pair that is not defined. Only one table can be defined. See *MAT_SPOTWELD where this option is used whenever *OPT = 7*.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	DSN	DSS	DLCIDSN	DLCIDSS			
Type	I	F	F	I	I			
Default	0	0.0	0.0	0	0			

Failure Cards. Provide as many as necessary. The next keyword (“*”) card terminates the table definition.

Card 2...	1	2	3	4	5	6	7	8
Variable	PID_I	PID_J	SNIJ	SSIJ	LCIDSNIJ	LCIDSSIJ		
Type	I	I	F	F	I	I		
Default	none	none	0.0	0.0	0	0		

VARIABLE	DESCRIPTION
ID	Identification number. Only one table is allowed.
DSN	Default value of the normal static stress at failure.
DSS	Default value of the transverse static stress at failure.
DLCIDSN	Load curve ID defining a scale factor for the normal stress as a function of strain rate. This factor multiplies DSN to obtain the failure value at a given strain rate.

DEFINE**DEFINE_SPOTWELD_FAILURE_RESULTANTS**

VARIABLE	DESCRIPTION
DLCIDSS	Load curve ID defining a scale factor for static shear stress as a function of strain rate. This factor multiplies DSS to obtain the failure value at a given strain rate.
PID_I	Part ID I
PID_J	Part ID J
SNIJ	The maximum axial stress at failure between parts I and J. The axial stress is computed from the solid element stress resultants, which are based on the nodal point forces of the solid element.
DSSIJ	The maximum shear stress at failure between parts I and J. The shear stress is computed from the solid element stress resultants, which are based on the nodal point forces of the solid element.
LCIDSNIJ	Load curve ID defining a scale factor for the normal stress as a function of strain rate. This factor multiplies SNIJ to obtain the failure value at a given strain rate.
LCIDSSIJ	Load curve ID defining a scale factor for static shear stress as a function of strain rate. This factor multiplies SSIJ to obtain the failure value at a given strain rate.

Remarks:

The stress based failure model, which was developed by *Toyota Motor Corporation*, is a function of the peak axial and transverse shear stresses. The entire weld fails if the stresses are outside of the failure surface defined by:

$$\left(\frac{\sigma_{rr}}{\sigma_{rr}^F}\right)^2 + \left(\frac{\tau}{\tau^F}\right)^2 - 1 = 0$$

where σ_{rr}^F and τ^F are specified in the above table by part ID pairs. LS-DYNA automatically identifies the part ID of the attached shell element for each node of the spot weld solid and checks for failure. If failure is detected the solid element is deleted from the calculation.

If the effects of strain rate are considered, then the failure criteria becomes:

$$\left[\frac{\sigma_{rr}}{f_{dsn}(\dot{\epsilon}^p)\sigma_{rr}^F}\right]^2 + \left[\frac{\tau}{f_{dss}(\dot{\epsilon}^p)\tau^F}\right]^2 - 1 = 0$$

***DEFINE_SPOTWELD_MULTISCALE**

Purpose: Associate beam sets with multi-scale spot weld types for modeling spot weld failure via the multi-scale spot weld method.

Spot Weld/Beam Set Association Cards. Provide as many cards as necessary. This input ends at the next keyword ("*") card.

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

VARIABLE	DESCRIPTION
TYPE	MULTISCALE spot weld type to use. See *INCLUDE_MULTISCALE_SPOTWELD.
BSET	Beam set which uses this multi-scale spot weld type for failure modeling.

Remarks:

See *INCLUDE_MULTISCALE_SPOTWELD for a detailed explanation of this capability.

DEFINE**DEFINE_SPOTWELD_RUPTURE_PARAMETER*****DEFINE_SPOTWELD_RUPTURE_PARAMETER**

Purpose: Define a parameter by part ID for shell elements attached to spot weld *beam* elements using the constrained contact option: *CONTACT_SPOTWELD. *This table will not work with other contact types.* Only one table is permitted in the problem definition. Data, which is defined in this table, is used by the stress based spot weld failure model developed by *Toyota Motor Corporation*. See *MAT_SPOTWELD where this option is activated by setting the parameter OPT to a value of 9. This spot weld failure model is a development of *Toyota Motor Corporation*.

Card 1	1	2	3	4	5	6	7	8
Variable	PID							
Type	I							

Card 2	1	2	3	4	5	6	7	8
Variable	C11	C12	C13	N11	N12	N13		SIG_PF
Type	F	F	F	F	F	F		F

Card 3	1	2	3	4	5	6	7	8
Variable	C21	C22	C23	N2				SIG_NF
Type	F	F	F	F				F

Card 4	1	2	3	4	5	6	7	8
Variable	LCDPA	LCDPM	LCDPS	LCDNA	LCDNM	LCDNS		NSMT
Type	I	I	I	I	I	I		I
Default	0	0	0	0	0	0		0

VARIABLE	DESCRIPTION
PID	Part ID for the attached shell.
C11-N2	Parameters for model, see Remarks below.
SIG_PF	Nugget pull-out stress, σ_P
SIG_NF	Nugget fracture stress, σ_F
LCDPA	Curve ID defining dynamic scale factor of spot weld axial load rate for nugget pull-out mode.
LCDPM	Curve ID defining dynamic scale factor of spot weld moment load rate for nugget pull-out mode.
LCDPS	Curve ID defining dynamic scale factor of spot weld shear load rate for nugget pull-out mode.
LCDNA	Curve ID defining dynamic scale factor of spot weld axial load rate for nugget fracture mode.
LCDNM	Curve ID defining dynamic scale factor of spot weld moment load rate for nugget fracture mode.
LCDNS	Curve ID defining dynamic scale factor of spot weld shear load rate for nugget fracture mode.
NSMT	The number of time steps used for averaging the resultant rates for the dynamic scale factors.

Remarks:

This failure model incorporates two failure functions, one for nugget pull-out and the other for nugget fracture. The nugget pull-out failure function is

$$F_p = \frac{C11 \times \frac{A}{D^{N11}} + C12 \times \frac{M}{D^{N12}} + C13 \times \frac{S}{D^{N13}}}{\sigma_P \left[1 + \left(\frac{\dot{\epsilon}^p}{C} \right)^{1/p} \right]},$$

where A , M , and S are the axial force, moment, and shear resultants respectively, D is the spot weld diameter, and the Cowper-Symonds coefficients are from the attached shell material model. If the Cowper-Symonds coefficients aren't specified, the term within the square brackets, [], is 1.0. The fracture failure function is

DEFINE**DEFINE_SPOTWELD_RUPTURE_PARAMETER**

$$F_n = \frac{\sqrt{(C21 \times A + C22 \times M)^2 + 3(C23 \times S)^2}}{D^{N2} \sigma_F \left[1 + \left(\frac{\dot{\varepsilon}^p}{C} \right)^{1/p} \right]}.$$

When the load curves for the rate effects are specified, the failure criteria are

$$F_p = \frac{C11 \times f_{dpa}(\dot{A}) \times \frac{A}{D^{N11}} + C12 \times f_{dpa}(\dot{M}) \times \frac{M}{D^{N12}} + C13 \times f_{dpa}(\dot{S}) \times \frac{S}{D^{N13}}}{\sigma_P}$$
$$F_n = \frac{\sqrt{[C21 \times f_{dna}(\dot{A}) \times A + C22 \times f_{dnm}(\dot{M}) \times M]^2 + 3[C23 \times f_{dns}(\dot{S}) \times S]^2}}{D^{N2} \sigma_F}$$

where f is the appropriate load curve scale factor. The scale factor for each term is set to 1.0 for when no load curve is specified. No extrapolation is performed if the rates fall outside of the range specified in the load curve to avoid negative scale factors. A negative load curve ID designates that the curve abscissa is the \log_{10} of the resultant rate. This option is recommended when the curve data covers several orders of magnitude in the resultant rate. Note that the load curve dynamic scaling replaces the Cowper-Symonds model for rate effects.

Failure occurs when either of the failure functions is greater than 1.0.

***DEFINE_SPOTWELD_RUPTURE_STRESS**

Purpose: Define a static stress rupture table by part ID for shell elements connected to spot weld *beam* elements using the constrained contact option: *CONTACT_SPOTWELD. *This table will not work with other contact types.* Only one table is permitted in the problem definition. Data, which is defined in this table, is used by the stress based spot weld failure model developed by *Toyota Motor Corporation*. See *MAT_SPOTWELD where this option is activated by setting the parameter *OPT* to a value of 6.

Part Cards. Define rupture stresses part by part. The next keyword ("*") card terminates this input.

Card	1	2	3	4	5	6	7	8
Variable	PID	SRSIG	SIGTAU	ALPHA				
Type	I	F	F	F				

VARIABLE	DESCRIPTION
PID	Part ID for the attached shell.
SRSIG	Axial (normal) rupture stress, σ_{rr}^F .
SRTAU	Transverse (shear) rupture stress, τ^F .
ALPHA	Scaling factor for the axial stress as defined by Toyota. The default value is 1.0.

Remarks:

The stress based failure model, which was developed by *Toyota Motor Corporation*, is a function of the peak axial and transverse shear stresses. The entire weld fails if the stresses are outside of the failure surface defined by:

$$\left(\frac{\sigma_{rr}}{\sigma_{rr}^F}\right)^2 + \left(\frac{\tau}{\tau^F}\right)^2 - 1 = 0$$

where σ_{rr}^F and τ^F are specified in the above table by part ID. LS-DYNA automatically identifies the part ID of the attached shell element for each node of the spot weld beam and independently checks each end for failure. If failure is detected in the end attached to the shell with the greatest plastic strain, the beam element is deleted from the calculation.

If the effects of strain rate are considered, then the failure criteria becomes:

DEFINE**DEFINE_SPOTWELD_RUPTURE_STRESS**

$$\left[\frac{\sigma_{rr}}{\sigma_{rr}^F(\dot{\varepsilon}^p)} \right]^2 + \left[\frac{\tau}{\tau^F(\dot{\varepsilon}^p)} \right]^2 - 1 = 0$$

where $\sigma_{rr}^F(\dot{\varepsilon}^p)$ and $\tau^F(\dot{\varepsilon}^p)$ are found by using the Cowper and Symonds model which scales the static failure stresses:

$$\sigma_{rr}^F(\dot{\varepsilon}^p) = \sigma_{rr}^F \left[1 + \left(\frac{\dot{\varepsilon}^p}{C} \right)^{1/p} \right]$$
$$\tau^F(\dot{\varepsilon}^p) = \tau^F \left[1 + \left(\frac{\dot{\varepsilon}^p}{C} \right)^{1/p} \right]$$

where $\dot{\varepsilon}^p$ is the average plastic strain rate which is integrated over the domain of the attached shell element, and the constants p and C are uniquely defined at each end of the beam element by the constitutive data of the attached shell. The constitutive model is described in the material section under keyword: *MAT_PIECEWISE_LINEAR_PLASTICITY.

The peak stresses are calculated from the resultants using simple beam theory.

$$\sigma_{rr} = \frac{N_{rr}}{A} + \frac{\sqrt{M_{rs}^2 + M_{rt}^2}}{\alpha Z} \quad \tau = \frac{M_{rr}}{2Z} + \frac{\sqrt{N_{rs}^2 + N_{rt}^2}}{A}$$

where the area and section modulus are given by:

$$A = \pi \frac{d^2}{4}$$
$$Z = \pi \frac{d^3}{32}$$

and d is the diameter of the spot weld beam.

***DEFINE_STAGED_CONSTRUCTION_PART_{OPTION}**

Available options include:

<BLANK>

SET

Purpose: Staged construction. This keyword offers a simple way to define parts that are removed (e.g., during excavation), added (e.g., new construction) and used temporarily (e.g., props) during the analysis. Available for solid, shell, thick shell, and beam element parts.

Part Cards. Provide as many as necessary. This input terminates at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	PID/PSID	STGA	STGR					
Type	I	I	I					
Default	none	none	none					

VARIABLE	DESCRIPTION
PID	Part ID (or Part Set ID for the SET keyword option)
STGA	Construction stage at which part is added. See Remark 1 .
STGR	Construction stage at which part is removed. See Remark 1 .

Remarks:

1. **Adding and Removing Stages.** Used with *DEFINE_CONSTRUCTION-STAGES (defines the meaning of stages STGA and STGR) and *CONTROL-STAGED_CONSTRUCTION. If STGA = 0, the part is present at the start of the analysis. If STGR = 0, the part is still present at the end of the analysis. Examples:
 - a) Soil that is excavated would have STGA = 0 but STGR > 0
 - b) New construction would have STGA > 0 and STGR = 0

- c) Temporary works would have STGA > 0, STGR > STGA.
- 2. **Construction Models.** This is a convenience feature that reduces the amount of input data needed for many typical construction models. Internally, LS-DYNA automatically creates *LOAD_REMOVE_PART, *LOAD_GRAVITY_PART and *LOAD_STIFFEN_PART referencing the same PID and using the default gravity and pre-construction stiffness factors from *CONTROL_STAGED_CONSTRUCTION. If the same PID is also referenced in the input file by any of the three types of *LOAD cards mentioned above, the load curves entered on the *LOAD cards will override the STGA, STGR inputs on *DEFINE_STAGED_CONSTRUCTION_PART.
- 3. **Use of Ramp Time.** Ramp time ATR on *DEFINE_CONSTRUCTION_STAGES is intended to reduce dynamic effects that would occur if loading or stresses were changed suddenly. It applies to gravity loading during addition and removal of parts, i.e. *LOAD_GRAVITY_PART automatically gets ramped up to its full value during the first ATR of the stage when a part is added (STGA), and ramped down to zero during the first ATR of the stage when a part is removed (STGR). For the stiffness, strength and stresses, ATR applies during removal of a part but not during addition. Starting from R11, when a part is added, the stiffness jumps immediately to its full value at the start of STGA. This behavior is different from versions R10 and previous, where ATR was also applied to stiffness during part addition. The disadvantage of the R10 behavior was that significant deformations could occur before the part reached its full stiffness.

DEFINE_STOCHASTIC_ELEMENT**DEFINE*****DEFINE_STOCHASTIC_ELEMENT_OPTION**

Options:

SOLID_VARIATION for solid elements

SHELL_VARIATION for shell elements

Purpose: Define the stochastic variation in the yield stress, damage/failure models, density, and elastic moduli for solid material models with the STOCHASTIC keyword option on an element basis. This keyword works currently with materials 10, 15, 24, 81, and 98. This option overrides values assigned by *DEFINE_STOCHASTIC_VARIATION.

Card 1	1	2	3	4	5	6	7	8
Variable	IDE	VARSY	VARF	VARRO	VARE			
Type	I	F	F	F	F			
Default	none	0.0	0.0	0.0	0.0			

VARIABLE	DESCRIPTION
IDE	Element ID
VARSY	The yield stress and its hardening function are scaled by $1.0 + \text{VARSY}$.
VARF	The failure criterion is scaled by $1.0 + \text{VARF}$.
VARRO	The density is scaled by $1.0 + \text{VARRO}$. This is intended to be used with topology optimization. This option is not available for shell elements.
VARE	The elastic moduli are scaled by $1.0 + \text{VARE}$. This is intended to be used with topology optimization.

*DEFINE

*DEFINE_STOCHASTIC_VARIATION

*DEFINE_STOCHASTIC_VARIATION

Purpose: Define the stochastic variation in the yield stress and damage/failure models for material models with the STOCHASTIC keyword option. This keyword currently works with materials 10, 15, 24, 81, 98, 280, and the shell version of material 123. It is now also possible to vary failure strain in GISSMO with this keyword.

Card Summary:

Card 1. This card is required

ID_SV	PID	PID_TYP	ICOR	VAR_S	VAR_F	IRNG	
-------	-----	---------	------	-------	-------	------	--

Card 2a. This card is included if and only if VAR_S = 0, 1, or 2.

R1	R2	R3					
----	----	----	--	--	--	--	--

Card 2b. This card is included if and only if VAR_S = 3 or 4.

LCID							
------	--	--	--	--	--	--	--

Card 3a. This card is included if and only if VAR_F = 0, 1, or 2.

R1	R2	R3					
----	----	----	--	--	--	--	--

Card 3b. This card is included if and only if VAR_S = 3 or 4.

LCID							
------	--	--	--	--	--	--	--

Data Cards:

Card 1	1	2	3	4	5	6	7	8
Variable	ID_SV	PID	PID_TYP	ICOR	VAR_S	VAR_F	IRNG	
Type	I	I	I	I	I	I	I	
Default	none	0	0	0	0	0	0	

VARIABLE

DESCRIPTION

ID_SV

Stochastic variation ID. A unique ID number must be used.

VARIABLE	DESCRIPTION
PID	*PART ID or *SET_PART ID.
PID_TYP	Flag for PID type. If PID and PID_TYP are both 0, then the properties defined here apply to all shell and solid parts using materials with the STOCHASTIC option. EQ.0: PID is a *PART ID. EQ.1: PID is a *SET_PART ID
ICOR	Correlation between the yield stress and failure strain scaling. EQ.0: Perfect correlation. EQ.1: No correlation. The yield stress and failure strain are independently scaled.
VAR_S	Variation type for scaling the yield stress (or tensile strength for material 280). EQ.0: The scale factor is 1.0 everywhere. EQ.1: The scale factor is a random number in the uniform random distribution in the interval defined by R1 and R2. EQ.2: The scale factor is a random number obeying the Gaussian distribution defined by R1, R2, and R3. EQ.3: The scale factor is defined by the probability distribution function defined by curve LCID. EQ.4: The scale factor is defined by the cumulative distribution function defined by curve LCID.
VAR_F	Variation type for scaling failure strain. EQ.0: The scale factor is 1.0 everywhere. EQ.1: The scale factor is random number in the uniform random distribution in the interval defined by R1 and R2. EQ.2: The scale factor is a random number obeying the Gaussian distribution defined by R1, R2, and R3. EQ.3: The scale factor is defined by the probability distribution function defined by curve LCID. EQ.4: The scale factor is defined by the cumulative distribution function defined by curve LCID.

DEFINE**DEFINE_STOCHASTIC_VARIATION**

VARIABLE	DESCRIPTION
IRNG	Flag for random number generation. EQ.0: Use deterministic (pseudo-) random number generator. The same input always leads to the same distribution. EQ.1: Use non-deterministic (true) random number generator. With the same input, a different distribution is achieved in each run.

Yield Stress Card for Built-in Distribution. Card 2 for VAR_S set to 0, 1, or 2.

Card 2a	1	2	3	4	5	6	7	8
Variable	R1	R2	R3					
Type	F	F	F					

Yield Stress Card for Load Curve. Card 2 for VAR_S set to 3 or 4.

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

Failure Strain Card for Built-in Distribution. Card 3 for VAR_F set to 0, 1, or 2.

Card 3a	1	2	3	4	5	6	7	8
Variable	R1	R2	R3					
Type	F	F	F					

Failure Strain Card for Load Curve. Card 3 for VAR_F set to 3 or 4.

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

VARIABLE	DESCRIPTION
R1, R2, R3	Real values to define the stochastic distribution. See below.
LCID	Curve ID defining the stochastic distribution. See below.

Remarks:

Each integration point x_g in the parts specified by PID is assigned the random scale factors R_S and R_F that are applied to the values calculated by the material model for the yield stress and failure strain.

$$\begin{aligned}\sigma_y &= R_s(x_g)\sigma_y(\bar{\epsilon}^p, \dots) \\ \bar{\epsilon}_{FAIL}^p &= R_F(x_g)\bar{\epsilon}_{FAIL}^p(\dot{\epsilon}, \bar{\epsilon}^p, \dots)\end{aligned}$$

The scale factors vary spatially over the model according to the chosen statistical distributions defined in this section and are independent of time. The scale factors may be completely correlated or uncorrelated with the default being completely correlated since the failure strain is generally reduced as the yield stress increases. The scale factors R_S and R_F may be stored as extra history variables as follows:

Material Model	History Variable # for R_S	History Variable # for R_F
10	5	6
15	7	8
24	6	7
81	6	7
98	7	8
123 (shells only)	6	7
280	13	n.a.
GISSMO	n.a.	ND + 20

The user is responsible for defining the distributions so that they are physically meaningful and are restricted to a realistic range. Since neither the yield stress nor the failure strain may be negative, for example, the minimum values of the distributions must always be greater than zero.

The probability that a particular value R will occur defines the *probability distribution function*, $P(R)$. Since a value must be chosen from the distribution, the integral from the minimum to the maximum value of R of the probability distribution function must be 1.0,

$$\int_{R_{\text{MIN}}}^{R_{\text{MAX}}} P(R) dR = 1.0 .$$

Another way to characterize a distribution is the *cumulative distribution function* $C(R)$ which defines the probability that a value will lie between R_{MIN} and R ,

$$C(R) = \int_{R_{\text{MIN}}}^R P(\hat{R}) d\hat{R}.$$

By definition $C(R_{\text{MIN}}) = 0$ and $C(R_{\text{MAX}}) = 1$. An inverse cumulative probability function D gives the number for a cumulative probability of $C(R)$,

$$D(C(R)) = R.$$

A random variable satisfying the probability distribution function $P(R)$ can be generated from a sequence of uniformly distributed numbers, \hat{R}_I , for $I = 1, N$, using the inverse cumulative distribution function D as

$$R_I = D(\hat{R}_I).$$

The scale factors for the yield stress and the failure strain may be generated using the same value of \hat{R}_I for both (ICOR = 0) or by using independent values each one (ICOR = 1). If the same values are used, there is perfect correlation, and the failure strain scale factor becomes an implicit value of yield stress scale factor.

VAR = 0. No Scaling.

The corresponding yield stress or failure strain is not scaled.

VAR = 1. Scaling from Uniform Distribution

A uniform distribution is specified by setting VAR = 1. The input variable R1 is interpreted as R_{MIN} and R2 as R_{MAX} . If R1 = R2, then the yield stress or failure strain will be scaled by R1.

When using the uniform random distribution, the probability of a particular value is given by

$$P(R) = \frac{1}{R_{\text{MAX}} - R_{\text{MIN}}}$$

and the cumulative probability function is given by

$$C(R) = \frac{R - R_{\text{MIN}}}{R_{\text{MAX}} - R_{\text{MIN}}}.$$

VAR = 2. Gaussian Distribution

The Gaussian distribution, VAR = 2, is smoothly varying with a peak at μ and 68% of the values occurring within the interval of one standard deviation σ , $[\mu - \sigma, \mu + \sigma]$. The input parameter R1 is interpreted as the mean, μ , while R2 is interpreted as the standard deviation, σ . There is a finite probability that the values of R will be outside of the range that are physically meaningful in the scaling process, and R3 which is interpreted as δ restricts the range of R to $[\mu - \delta, \mu + \delta]$. The resulting truncated Gaussian distribution is rescaled such that,

$$D(\mu + \delta) = 1.$$

VAR = 3 or 4. Distribution from a Load Curve

The user may directly specify the probability distribution function or the cumulative probability distribution function with *DEFINE_CURVE by setting VAR = 3 or VAR = 4, respectively, and then specifying the required curve ID on the next data card.

Stochastic variations may be used simultaneously with the heat affected zone (HAZ) options in LS-DYNA (see *DEFINE_HAZ_PROPERTIES). The effect of the scale factors from stochastic variation and HAZ options are multiplied together to scale the yield stress and failure strain,

$$\begin{aligned}\sigma_y &= R_S(x_g)R_S^{\text{HAZ}}\sigma_y(\bar{\varepsilon}^p, \dots) \\ \bar{\varepsilon}_{\text{FAIL}}^p &= R_F(x_g)R_F^{\text{HAZ}}\bar{\varepsilon}_{\text{FAIL}}^p(\dot{\varepsilon}, \bar{\varepsilon}^p, \dots).\end{aligned}$$

*DEFINE

*DEFINE_STOCHASTIC_VARIATION_PROPERTIES

*DEFINE_STOCHASTIC_VARIATION_PROPERTIES

Purpose: Define the stochastic variation in predefined material model history variables. The predefined history variables to which the stochastic variation can be applied are listed in the documentation of the material model. It is an extension of *DEFINE_STOCHASTIC_VARIATION to more material models and additional material properties.

This keyword currently works with materials 10, 15, 24, 81, 98, and 213 as well the shell version of material 123.

Card Summary:

Card 1. This card is required.

ID_SV	MTYPE	PID	PID_TYP	IRNG	NUMV	NUM_BEG	
-------	-------	-----	---------	------	------	---------	--

Card 2a. This card is included if and only if VARTYP = 0, 1, or 2 (built-in distributions). For each predefined history variable, include a combination of Cards 2a and 2b in the order of the history variables.

VARTYP	CORLGRP	R1	R2	R3			
--------	---------	----	----	----	--	--	--

Card 2b. This card is included if and only if VARTYP = 3 or 4 (distributions defined by curves). For each predefined history variable, include a combination of Cards 2a and 2b in the order of the history variables.

VARTYP	CORLGRP	LCID					
--------	---------	------	--	--	--	--	--

Data Cards:

Card 1	1	2	3	4	5	6	7	8
Variable	ID_SV	MTYPE	PID	PID_TYP	IRNG	NUMV	NUM_BEG	
Type	I	I	I	I	I	I	I	
Default	none	none	0	0	0	0	0	

VARIABLE

DESCRIPTION

ID_SV

Stochastic variation ID. A unique ID number must be used.

DEFINE_STOCHASTIC_VARIATION_PROPERTIES**DEFINE**

VARIABLE	DESCRIPTION
MTYPE	Material type. The available types are 10, 15, 24, 81, 98, and 213. This variation only works for this material.
PID	*PART ID or *SET_PART ID.
PID_TYP	Flag for PID type. If PID and PID_TYP are both 0, then the properties defined here apply to all shell and solid parts using the materials defined in the <i>OPTION</i> string. EQ.0: PID is a *PART ID. EQ.1: PID is a *SET_PART ID.
IRNG	Flag for random number generation. EQ.0: Use deterministic (pseudo-) random number generator. The same input always leads to the same distribution. EQ.1: Use non-deterministic (true) random number generator. With the same input, a different distribution is achieved in each run.
NUMV	Number of variations for a user material
NUM_BEG	The location of the first variation in the history variables for a user material. The remaining variations are added sequentially.

Card for Built-in Distributions. Card 2 for VARTYP set to 0, 1, or 2.

Card 2a	1	2	3	4	5	6	7	8
Variable	VARTYP	CORLGRP	R1	R2	R3			
Type	I	I	F	F	F			

Card for Distributions Defined by a Curve. Card 2 for VARTYP set to 3 or 4.

Card 2b	1	2	3	4	5	6	7	8
Variable	VARTYP	CORLGRP	LCID					
Type	I	I	I					

DEFINE**DEFINE_STOCHASTIC_VARIATION_PROPERTIES**

VARIABLE	DESCRIPTION
VARTYP	Variation type for scaling the material property: EQ.0: The scale factor is 1.0 everywhere. EQ.1: The scale factor is a random number in the uniform random distribution in the interval defined by R1 and R2. EQ.2: The scale factor is a random number obeying the Gaussian distribution defined by R1, R2, and R3. EQ.3: The scale factor is defined by the probability distribution function defined by curve LCID. EQ.4: The scale factor is defined by the cumulative distribution function defined by curve LCID.
CORLGRP	Correlation group number. If CORLGRP is 0, then the random number for the distribution is uncorrelated with all the other distributions. The same random number is used for evaluating all the distributions having the same positive integer value for CORLGRP.
R1, R2, R3	Real values to define the stochastic distribution. See Remark 6 .
LCID	Curve ID defining the stochastic distribution. See Remark 6 .

Remarks:

1. **Physically Meaningful Distributions.** The user is responsible for defining the distributions so that they are physically meaningful and are restricted to a realistic range. A certain history variable may not be negative, for example, so the minimum values of the distributions must always be greater than zero.
2. **Compatibility.** This implementation is compatible with *DEFINE_STOCHASTIC_VARIATION for materials 10, 15, 24, 81, 98, and the shell version of material 123 in that it scales the yield stress and the failure strain. This implementation also supports stochastic variations with user materials but does not support GISSMO. The results of the two implementations will not be identical because the generality of the correlation groups results in the random number generator being called in a different order.
3. **Correlation Group Number.** The correlation group number allows the user to selectively correlate different history variables for a given material model. For a material model that is scaling five history variables with the group numbers being 0, 0, 1, 2, 2, each of the first two variables will be uncorrelated with the other

four, the third variable is also uncorrelated with the rest, while the last two are perfectly correlated with each other but not with any others.

4. **Number of Variations.** If more variations are defined in the input than are permitted by the material model, the excess will be ignored. If fewer variations are defined, the unspecified variations will be set to 1.0.
5. **Distribution Functions.** The probability that a particular value R will occur defines the *probability distribution function*, $P(R)$. Since a value must be chosen from the distribution, the integral from the minimum to the maximum value of R of the probability distribution function must be 1.0,

$$\int_{R_{\text{MIN}}}^{R_{\text{MAX}}} P(R) dR = 1.0 .$$

Another way to characterize a distribution is the *cumulative distribution function*, $C(R)$, which defines the probability that a value will lie between R_{MIN} and R ,

$$C(R) = \int_{R_{\text{MIN}}}^R P(\hat{R}) d\hat{R} .$$

By definition $C(R_{\text{MIN}}) = 0$ and $C(R_{\text{MAX}}) = 1$. An inverse cumulative probability function D gives the number for a cumulative probability of $C(R)$,

$$D(C(R)) = R .$$

A random variable satisfying the probability distribution function $P(R)$ can be generated from a sequence of uniformly distributed numbers, \hat{R}_I , for $I = 1, \dots, N$, using the inverse cumulative distribution function D as

$$R_I = D(\hat{R}_I).$$

6. **VARTYP.** The applied distribution and input fields for Card 2 depends on VARTYP. Descriptions of each distribution follows.
 - a) *No Scaling* ($VARTYP = 0$). The distribution value is 1.0.
 - b) *Scaling from Uniform Distribution* ($VARTYP = 1$). A uniform distribution is specified by setting VAR = 1. The input variable R1 is interpreted as R_{MIN} and R2 as R_{MAX} . If R1 = R2, then the material property will be scaled by R1.

When using the uniform random distribution, the probability of a particular value is given by

$$P(R) = \frac{1}{R_{\text{MAX}} - R_{\text{MIN}}} ,$$

and the cumulative probability function is given by

$$C(R) = \frac{R - R_{\text{MIN}}}{R_{\text{MAX}} - R_{\text{MIN}}} .$$

- c) *Gaussian Distribution (VARTYP = 2)*. The Gaussian distribution, VAR = 2, is smoothly varying with a peak at μ and 63 percent of the values occurring within the interval of one standard deviation σ , $[\mu - \sigma, \mu + \sigma]$. The input parameter R1 is interpreted as the mean, μ , while R2 is interpreted as the standard deviation, σ . There is a finite probability that the values of R will be outside of the range that are physically meaningful in the scaling process, and R3 which is interpreted as δ restricts the range of R to $[\mu - \delta, \mu + \delta]$. The resulting truncated Gaussian distribution is rescaled such that,

$$D(\mu + \delta) = 1 .$$

- d) *Distribution from a Load Curve (VARTYP = 3 or 4)*. The user may directly specify the probability distribution function or the cumulative probability distribution function with *DEFINE_CURVE by setting VAR = 3 or VAR = 4, respectively, and then specifying the required curve ID.
7. **Heat Affected Zone.** Stochastic variations may be used simultaneously with the heat affected zone (HAZ) options in LS-DYNA (see *DEFINE_HAZ_PROPERTIES). The effect of the scale factors from stochastic variation and HAZ options are multiplied together to scale the predefined history variables.

DEFINE_TABLE**DEFINE*****DEFINE_TABLE**

Purpose: To interpolate from point data a continuously indexed family of nonintersecting curves. The family of curves, \mathcal{F} , consists of x-y curves, $f_s(x)$, indexed by a parameter, s .

$$\mathcal{F} = \{f_s(x) | \forall s \in [s_{\min}, s_{\max}]\}.$$

The interpolation is built up by sampling functions in \mathcal{F} at discrete parameter values, s_i ,

$$f_{s_i}(x) \in \mathcal{F}.$$

The points, s_i , are input to LS-DYNA on the data cards for the *DEFINE_TABLE keyword. LS-DYNA requires that they be ordered from least to greatest. The curves, $f_{s_i}(x)$, must be defined as lists of (x, y) pairs in a collection of *DEFINE_CURVE sections that directly follow the *DEFINE_TABLE section. Each *DEFINE_CURVE section is paired to its corresponding s_i value by list position (and not load curve ID, for that see *DEFINE_TABLE_2D).

Card 1	1	2	3	4	5	6	7	8
Variable	TBID	SFA	OFFA					
Type	I	F	F					
Default	none	1.	0.					

Points Cards. Place one point per card. The values must be in ascending order. Input is terminated when a “*DEFINE_CURVE” keyword card is found.

Card 2	1	2	3	4	5	6	7	8
Variable	VALUE							
Type	F							
Default	0.0							

Include one *DEFINE_CURVE input section here for each point defined above. The i^{th} *DEFINE_CURVE card contains the curve at the i^{th} *DEFINE_TABLE value.

*DEFINE

*DEFINE_TABLE

VARIABLE	DESCRIPTION
TBID	Table ID. Tables and Load curves may not share common ID's. LS-DYNA allows load curve ID's and table ID's to be used interchangeably.
SFA	Scale factor for VALUE.
OFFA	Offset for VALUE, see explanation below.
VALUE	Load curve will be defined corresponding to this value, e.g., this value could be a strain rate, see purpose above.

Motivation:

This capability was implemented with strain-rate dependent stress-strain relations in mind. To define such a function, the first step is to tabulate stress-strain curves at known strain-rate values. Then, the list of strain-rates is written in ascending order to the data cards following *DEFINE_TABLE. Following *DEFINE_TABLE, the tabulated stress-strain curves must be input to LS-DYNA as a set of *DEFINE_CURVE sections ordered so that the i^{th} curve corresponds to the i^{th} strain-rate point. This section is structured as:

```
*DEFINE_TABLE
strain-rate point 1
strain-rate point 2
:
strain-rate point n
*DEFINE_CURVE
[stress-strain curve at strain-rate 1]
*DEFINE_CURVE
[stress-strain curve at strain-rate 2]
:
*DEFINE_CURVE
[stress-strain curve at strain-rate n]
```

Details, Features and Limitations:

1. All the curves in a table must start from the same abscissa value and end at the same abscissa value. This limitation is necessary to avoid slow indirect addressing in the inner loops used in the constitutive model stress evaluation. Curves must not intersect except at the origin and end points.

2. Each curve may have unique spacing and an arbitrary number of points in its definition.
3. All the curves in a table must share the same value of LCINT.
4. In most applications, curves can only be extrapolated in one direction, that is, to the right of the last data point. An example would be curves representing effective stress vs. effective plastic strain. For cases when extrapolation is only to the right, the curves comprising a table are allowed to intersect *only* at their starting point but the curves *and their extrapolations* must not intersect elsewhere.

For other applications in which the curves are extrapolated in both directions, the curves and their extrapolations are not allowed to intersect except at the origin (0,0). An example would be curves representing force vs. change in gage length where negative values are compressive and positive values are tensile.

5. Load curve IDs defined for the table may be referenced elsewhere in the input.
6. No keyword commands may come between *DEFINE_TABLE and the *DEFINE_CURVE commands that feed the table. The set of *DEFINE_CURVE commands must not be interrupted by any other keyword. This coupling between *DEFINE_TABLE and subsequent *DEFINE_CURVE commands is an exception to the general order-independence of the keyword format.
7. VALUE is scaled in the same manner as in *DEFINE_CURVE, i.e.,
$$\text{Scaled value} = \text{SFA} \times (\text{Defined value} + \text{OFFA}).$$
8. Unless stated otherwise in the description of a keyword command that references a table, there is no extrapolation beyond the range of VALUES defined for the table. For example, if the table VALUE represents strain rate and the calculated strain rate exceeds the last/highest VALUE given by the table, the stress-strain curve corresponding to the last/highest table VALUE will be used.

*DEFINE

*DEFINE_TABLE_2D

*DEFINE_TABLE_2D

Purpose: Define a table. Unlike the *DEFINE_TABLE keyword above, a curve ID is specified for each value defined in the table. This allows the same curve ID to be referenced by multiple tables, and the curves may be defined anywhere in the input file. Other than these differences from *DEFINE_TABLE, the general rules given in the remarks of *DEFINE_TABLE still apply.

Card 1	1	2	3	4	5	6	7	8
Variable	TBID	SFA	OFFA					
Type	I	F	F					
Default	none	1.	0.					

Points Cards. Place one point per card. The values must be in ascending order. Input is terminated when the next keyword card is found.

Card 2	1	2	3	4
Variable	VALUE	CURVE ID		
Type	F	I		
Default	0.0	none		

VARIABLE	DESCRIPTION
TBID	Table ID. Tables and Load curves may not share common ID's. LS-DYNA allows load curve ID's and table ID's to be used interchangeably.
SFA	Scale factor for VALUE.
OFFA	Offset for VALUE; see remarks for *DEFINE_TABLE.
VALUE	Load curve will be defined corresponding to this value. The value could be, for example, a strain rate.
CURVEID	Load curve ID. See Remark 1 .

Remarks:

1. **Curve IDs.** Though generally of no concern to the user, curve CURVEID is automatically duplicated during initialization and the duplicate curve is automatically assigned a unique curve ID. The generated curve IDs used by the table are revealed in d3hsp. It is generally only necessary to know the generated curve IDs when interpreting warning messages about those curves.

*DEFINE

*DEFINE_TABLE_3D

*DEFINE_TABLE_3D

Purpose: Define a three-dimensional table. For each value defined below, a table ID is specified. For example, in a thermally dependent material model, the value given below could correspond to temperature for a table ID defining effective stress versus strain curves for a set of strain rate values. Each table ID can be referenced by multiple three-dimensional tables, and the tables may be defined anywhere in the input.

Card 1	1	2	3	4	5	6	7	8
Variable	TBID	SFA	OFFA					
Type	I	F	F					
Default	none	1.	0.					

Points Cards. Place one point per card. The values must be in ascending order. Input is terminated when the next keyword card is found.

Card 2	1	2	3	4
Variable	VALUE	TABLE ID		
Type	F	I		
Default	0.0	none		

VARIABLE	DESCRIPTION
TBID	Table ID. Tables and load curves may not share common ID's. LS-DYNA allows load curve ID's and table ID's to be used interchangeably.
SFA	Scale factor for VALUE.
OFFA	Offset for VALUE, see explanation below.
VALUE	Load curve will be defined corresponding to this value, e.g., this value could be a strain rate for example.
TABLEID	Table ID.

Remarks:

1. **Scaling.** VALUE is scaled in the same manner as in *DEFINE_CURVE, that is,
Scaled value = SFA × (Defined value + OFFA).
2. **Extrapolation.** Unless stated otherwise in the description of a keyword command that references a table, there is no extrapolation beyond the range of VALUES defined for the table. For example, if the table VALUE represents strain rate and the calculated strain rate exceeds the last/highest VALUE given by the table, the stress-strain curve corresponding to the last/highest table VALUE will be used

DEFINE**DEFINE_TABLE_{X}D*****DEFINE_TABLE_{X}D**

Available options for X are 4, 5, 6, 7, 8, or 9.

Purpose: Define an X-dimensional table. For each value defined below, a table ID of one lower dimension is specified. This keyword works in the same way as *DEFINE_TABLE_3D, but with a higher dimensionality (current maximum would be X = 9).

Card 1	1	2	3	4	5	6	7	8
Variable	TBID	SFA	OFFA					
Type	I	F	F					
Default	none	1.	0.					

Points Cards. Place one point per card. The values must be in ascending order. Input is terminated when the next keyword card is found.

Card 2	1	2	3	4
Variable	VALUE	TABLE ID		
Type	F	I		
Default	0.0	none		

VARIABLE**DESCRIPTION**

TBID Table ID. Tables and load curves may not share common IDs. LS-DYNA allows load curve IDs and table IDs to be used interchangeably.

SFA Scale factor for VALUE

OFFA Offset for VALUE; see explanation below.

VALUE Value which corresponds to a X – 1 dimension with ID TABLEID

TABLEID Table ID of table with one dimension smaller

Remarks:

1. **Scaling.** VALUE is scaled in the same manner as in *DEFINE_CURVE, that is,
Scaled value = SFA × (Defined value + OFFA).
2. **Extrapolation.** Unless stated otherwise in the description of a keyword command that references a table, there is no extrapolation beyond the range of values defined for the table. For example, if the table VALUE represents strain rate and the calculated strain rate exceeds the last/highest VALUE given by the table, the stress-strain curve corresponding to the last/highest table VALUE will be used.
3. **Example.** The following input shows the possible definition of stress-strain curves for *MAT_251 with 3 points each for two values of history variable #7 (80 and 200), two values for history variable #6 (20 and 400), and two strain rates (0 and 100) using a TABLE_4D:

```
*DEFINE_TABLE_4D
 10000
     80.0      1000
    200.0      2000
$-----
*DEFINE_TABLE_3D
 1000
    20.0      1100
   400.0      1200
$-----
*DEFINE_TABLE
 1100
    0.0      100.0
*DEFINE_CURVE
 1101
    0.0      360.0
    0.3      570.0
    1.0      780.0
*DEFINE_CURVE
 1102
    0.0      470.0
    0.3      680.0
    1.0      860.0
$-----
*DEFINE_TABLE
 1200
    0.0      100.0
*DEFINE_CURVE
 1201
    0.0      180.0
    0.3      285.0
    1.0      390.0
*DEFINE_CURVE
 1202
    0.0      235.0
    0.3      340.0
    1.0      430.0
$-----
```

```
$-----
*DEFINE_TABLE_3D
 2000
    20.0      2100
   400.0      2200
$-----
*DEFINE_TABLE
 2100
    0.0      100.0
*DEFINE_CURVE
 2101
    0.0      540.0
    0.3      855.0
    1.0      1170.0
*DEFINE_CURVE
 2102
    0.0      705.0
    0.3      1020.0
    1.0      1290.0
$-----
*DEFINE_TABLE
 2200
    0.0      100.0
*DEFINE_CURVE
 2201
    0.0      270.0
    0.3      427.5
    1.0      585.0
*DEFINE_CURVE
 2202
    0.0      352.5
    0.3      510.0
    1.0      645.0
$-----
```

*DEFINE

*DEFINE_TABLE_COMPACT

*DEFINE_TABLE_COMPACT

Purpose: Define a multi-dimensional table, meaning a functional dependence on several variables. Each ordinate value (for example, stress) can depend on several abscissa values (such as strain, strain rate, history variables, ...).

Card Summary:

Card 1. This card is required.

TBID	NVAR	LCINT	MATHIS	INEXETC	ISCALE		
------	------	-------	--------	---------	--------	--	--

Card 2. This card is included if MATHIS = 1.

	HIS1	HIS2	HIS3	HIS4	HIS5	HIS6	HIS7
--	------	------	------	------	------	------	------

Card 2.1. This card is included if MATHIS = 1 and NVAR > 7.

	HIS8	HIS9					
--	------	------	--	--	--	--	--

Card 3. This card is included if INEXETC = 1.

	IXE1	IXE2	IXE3	IXE4	IXE5	IXE6	IXE7
--	------	------	------	------	------	------	------

Card 3.1. This card is included if INEXETC = 1 and NVAR > 7.

	IXE8	IX89					
--	------	------	--	--	--	--	--

Card 4. This card is included if ISCALE = 1.

SF0	SFA1	SFA2	SFA3	SFA4	SFA5	SFA6	SFA7
-----	------	------	------	------	------	------	------

Card 4.1. This card is included if ISCALE = 1 and NVAR > 7.

	SFA8	SFA9					
--	------	------	--	--	--	--	--

Card 5. Include one of this card for each ordinate value if NVAR ≤ 7 or a set of this card and Card 4.1 for each ordinate value if NVAR > 7. This input ends with the next keyword ("**") card.

01	A1.1	A1.2	A1.3	A1.4	A1.5	A1.6	A1.7
----	------	------	------	------	------	------	------

Card 5.1. Include this in a set of this card with Card 4 if NVAR > 7 for each ordinate value. This input ends with the next keyword ("**") card,

	A1.8	A1.9					
--	------	------	--	--	--	--	--

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	TBID	NVAR	LCINT	MATHIS	INEXETC	ISCALE		
Type	I	I	I	I	I	I		
Default	none	none	0	0	0	0		

VARIABLE	DESCRIPTION
TBID	Table ID. Tables and load curves may not share common IDs. LS-DYNA allows load curve IDs and table IDs to be used interchangeably.
NVAR	Number of variables (dimension of the table). Current maximum is 9.
LCINT	Number of discretization points. EQ.0: Value of LCINT from *CONTROL SOLUTION will be used.
MATHIS	Material history flag. Option to identify the abscissa variables as a specified history variable number(s) (see Remarks 3 and 6). Additional Card 2 (and possibly Card 2.1) is read if this option is active. EQ.0: Off EQ.1: On
INEXETC	Extra curve settings flag. Option to assign settings about curve discretization, extrapolation and interpolation for each abscissa variable. Additional Card 3 (and possibly Card 3.1) is read if this option is active. EQ.0: Off EQ.1: On
ISCALE	Variable scaling flag. Option to scale ordinate and abscissa values. Additional Card 4 (and possibly Card 4.1) is read if this option is active. EQ.0: Off

DEFINE**DEFINE_TABLE_COMPACT**

VARIABLE	DESCRIPTION							
	EQ.1: On							

Material History Card. Additional card for MATHIS = 1 only.

Card 2	1	2	3	4	5	6	7	8
Variable		HIS1	HIS2	HIS3	HIS4	HIS5	HIS6	HIS7
Type		I	I	I	I	I	I	I

Optional Material History Card. Include up to 2 additional values if MATHIS = 1 and NVAR > 7.

Card 2.1	1	2	3	4	5	6	7	8
Variable		HIS8	HIS9					
Type		I	I					

VARIABLE	DESCRIPTION							
HIS1, HIS2, ...	History variable numbers which indicate that variable X is HISX. For instance, setting HIS2 = 6 indicates that variable 2 is history variable 6 for the material. A value of 0 indicates that the abscissa variable is not a history variable. See Remarks 3 and 6 .							

Extra Settings Card. Additional card for INEXETC = 1 only.

Card 3	1	2	3	4	5	6	7	8
Variable		IXE1	IXE2	IXE3	IXE4	IXE5	IXE6	IXE7
Type		I	I	I	I	I	I	I

Optional Extra Settings Card. Include up to 2 additional values if INEXETC = 1 and NVAR > 7.

Card 3.1	1	2	3	4	5	6	7	8
Variable		IXE8	IXE9					
Type		I	I					

VARIABLE**DESCRIPTION**

IXE1, IXE2,

...

Extra settings assigned to abscissa values. See [Remark 3](#). IXE is interpreted digit-wise, namely as,

$$\text{IXE} = [\text{MLK}] = K + 10 \times L + 100 \times M .$$

with the following interpretations:

K.EQ.0: Discretized curve using LCINT is used.

K.EQ.1: Original curve as defined in this keyword is used.

L.EQ.0: Extrapolation at lower and upper end is used.

L.EQ.1: No extrapolation is used, meaning the first and last values are kept constant.

M.EQ.0: Linear interpolation between data points is used.

M.EQ.1: Logarithmic interpolation between data points is used.

Variable Scaling Card. Additional card for ISCALE = 1 only.

Card 4	1	2	3	4	5	6	7	8
Variable	SF0	SFA1	SFA2	SFA3	SFA4	SFA5	SFA6	SFA7
Type	F	F	F	F	F	F	F	F

Optional Variable Scaling Card. Include up to 2 additional values if ISCALE = 1 and NVAR > 7.

Card 4.1	1	2	3	4	5	6	7	8
Variable		SFA8	SFA9					
Type		F	F					

*DEFINE

*DEFINE_TABLE_COMPACT

VARIABLE	DESCRIPTION
SFO	Scale factor for ordinate value. Default set to 1.0.
SFA1, SFA2,	Scale factors for abscissa values. Defaults set to 1.0.
...	

Point Cards. Place one ordinate value with NVAR abscissa values per card. If NVAR > 7, include sets of Card 5 and Card 5.1 for each ordinate value. The values must be placed in a special ascending order; see Remarks. Input ends with the next keyword ("*") card.

Card 5	1	2	3	4	5	6	7	8
Variable	01	A1.1	A1.2	A1.3	A1.4	A1.5	A1.6	A1.7
Type	F	F	F	F	F	F	F	F

Optional Point Cards. Include this card if NVAR > 7 in a set with Card 5 for the same ordinate value to define up to 2 additional abscissa values.

Card 5.1	1	2	3	4	5	6	7	8
Variable		A1.8	A1.9					
Type		F	F					

VARIABLE	DESCRIPTION
O1, O2, ...	Ordinate (function) values. See Remarks 2, 4 and 5 .
A1.X, A2.X,	Abscissa values of variable X. See Remarks 2, 4 and 5 .
...	

Remarks:

1. **Internal Conversion.** The input from this keyword is internally converted into a *DEFINE_TABLE_{X}D with dimension X = NVAR. Corresponding sub-tables and curves will automatically be created, and all rules/properties described in *DEFINE_CURVE / *DEFINE_TABLE hold.

2. **Input Syntax.** The abscissa values of the first variable (second column) should increase first, while values in the following columns are kept constant. Then, values of variable 2 should increase next while the remaining variables to the right are kept constant. Next, the same for variable 3, and so on.
3. **Support Material Models for Material History and Extra Settings.** The options MATHIS and INEXETC are currently supported for *MAT_024 with VP = 3 and *MAT_ADD_DAMAGE_DIEM (or *MAT_ADD_EROSION with IDAM < 0).
4. **Basic Example.** The following input shows the possible definition of stress-strain curves for *MAT_251 with 3 points each for two different values of history variable #7 (80 and 200), two values for history variable #6 (20 and 400), and two strain rates (0 and 100), as an alternative to *DEFINE_TABLE_4D (see remarks there for comparison):

```
*DEFINE_TABLE_COMPACT
$ tbid    numvar      lcint
  10000        4
$ value     var1     var2     var3     var4
$ sig       eps     rate his#6 his#7
  360.0     0.0     0.0    20.0    80.0
  570.0     0.3     0.0    20.0    80.0
  780.0     1.0     0.0    20.0    80.0
  470.0     0.0    100.0   20.0    80.0
  680.0     0.3    100.0   20.0    80.0
  860.0     1.0    100.0   20.0    80.0
  180.0     0.0     0.0   400.0   80.0
  285.0     0.3     0.0   400.0   80.0
  390.0     1.0     0.0   400.0   80.0
  235.0     0.0    100.0   400.0   80.0
  340.0     0.3    100.0   400.0   80.0
  430.0     1.0    100.0   400.0   80.0
  540.0     0.0     0.0   20.0   200.0
  855.0     0.3     0.0   20.0   200.0
 1170.0     1.0     0.0   20.0   200.0
  705.0     0.0    100.0   20.0   200.0
 1020.0     0.3    100.0   20.0   200.0
 1290.0     1.0    100.0   20.0   200.0
  270.0     0.0     0.0   400.0   200.0
  427.5     0.3     0.0   400.0   200.0
  585.0     1.0     0.0   400.0   200.0
  352.5     0.0    100.0   400.0   200.0
  510.0     0.3    100.0   400.0   200.0
```

5. **Empty Columns.** It is possible to leave individual columns without defined values. For instance, if history variable #6 were to be left out in the above example, the table would reduce to:

```
*DEFINE_TABLE_COMPACT
$ tbid    numvar      lcint
  10000        4
$ value     var1     var2     var3     var4
```

DEFINE**DEFINE_TABLE_COMPACT**

\$	sig	eps	rate	his#6	his#7
	360.0	0.0	0.0		80.0
	570.0	0.3	0.0		80.0
	780.0	1.0	0.0		80.0
	470.0	0.0	100.0		80.0
	680.0	0.3	100.0		80.0
	860.0	1.0	100.0		80.0
	540.0	0.0	0.0		200.0
	855.0	0.3	0.0		200.0
	1170.0	1.0	0.0		200.0
	705.0	0.0	100.0		200.0
	1020.0	0.3	100.0		200.0
	1290.0	1.0	100.0		200.0

6. **MATHIS Example.** MATHIS = 1 causes the reading of one to two additional cards depending on NVAR which indicate which history variable a given abscissa variable is. Usually there is a fixed order to the history variables used in this table, but MATHIS = 1 gives you more freedom when assigning the variables to the table. In the following example, variable 2 is history variable #1, variable 3 is history variable #6, variable 4 is history variable #7, variable 5 is history variable #8, and variable 6 is history variable #9.

```
*DEFINE_TABLE_COMPACT
$      tbid    numvar    lcint    mathis    inexetc
$      10000     6        201       1          1
$          his1    his2    his3    his4    his5    his6
$          0        1        6        7        8        9
$          ixe1    ixe2    ixe3    ixe4    ixe5    ixe6
$          0        111     10       10       10       10
$      value    var1    var2    var3    var4    var5    var6
$      sig      eps    his#1    his#6    his#7    his#8    his#9
360.0    0.0    0.0    20.0    80.0    0.0    1.0
570.0    0.3    0.0    20.0    80.0    0.0    1.0
780.0    1.0    0.0    20.0    80.0    0.0    1.0
:
```

DEFINE_TABLE_MATRIX**DEFINE*****DEFINE_TABLE_MATRIX**

Purpose: This is an alternative input format for *DEFINE_TABLE that allows for reading data from an unformatted text file containing a matrix with data separated by comma delimiters. The purpose is to use data saved directly from excel sheets without having to convert it to keyword syntax.

Card 1	1	2	3	4	5	6	7	8
Variable	TBID				FILENAME			
Type	I				A70			

Card 2	1	2	3	4	5	6	7	8
Variable	NROW	NCOL	SROW	SCOL	SVAL	OROW	OCOL	OVAL
Type	I	I	F	F	F	F	F	F
Default	none	none	1.	1.	1.	0.	0.	0.

VARIABLE	DESCRIPTION
TBID	Table ID. Tables and Load curves may not share common ID's. LS-DYNA allows load curve ID's and table ID's to be used interchangeably.
FILENAME	Name of file containing table data (stored as a matrix).
NROW	Number of rows in the matrix, same as number of rows in the file FILENAME. A negative value of NROW switches the interpretation of rows and columns in the read matrix, see Remarks 1 and 2 .
NCOL	Number of columns in the matrix, same as number of data entries per row in the file FILENAME
SROW	Scale factor for row data, see Remark 2 .
SCOL	Scale factor for column data, see Remark 2 .
SVAL	Scale factor for matrix values, see Remark 2 .

VARIABLE	DESCRIPTION
OROW	Offset for row data, see Remark 2 .
OCOL	Offset for column data, see Remark 2 .
OVAL	Offset for matrix values, see Remark 2 .

Remarks:

1. **Example.** The use of this keyword allows for inputting a table in form of a matrix from a file, exemplified here by a 4×5 matrix.

	C1	C2	C3	C4
R1	V11	V12	V13	V14
R2	V21	V22	V23	V24
R3	V31	V32	V33	V34
:	:	:	:	:

The unformatted file representing this matrix would contain the following data

```
,C1,C2,C3,C4  
R1,V11,V12,V13,V14  
R2,V21,V22,V23,V24  
R3,V31,V32,V33,V34
```

Note that the first entry in the matrix is a dummy and delimited by an initial comma in the file. The keyword card for this matrix is: (TBID = 1000 and the filename is file.txt)

```
*DEFINE_TABLE_MATRIX  
1000,file.txt  
4,5
```

This is equivalent to using:

```
*DEFINE_TABLE  
1000  
C1  
C2  
C3  
C4  
*DEFINE_CURVE  
1001  
R1,V11  
R2,V21  
R3,V31  
*DEFINE_CURVE
```

```
1002  
R1,V12  
R2,V22  
R3,V32  
*DEFINE_CURVE  
1003  
R1,V13  
R2,V23  
R3,V33  
*DEFINE_CURVE  
1004  
R1,V14  
R2,V24  
R3,V34
```

2. **Scaling, Offsets, and Transposition.** All entries in the matrix can be scaled and offset following the convention for other tables and curves:

$$\text{Scaled Value} = S[\text{ROW}/\text{COL}] \times (\text{Value} + O[\text{ROW}/\text{COL}])$$

Finally, the matrix can be transposed by setting NROW to a negative value. In the example above ([Remark 1](#)) this would mean that

```
*DEFINE_TABLE_MATRIX  
1000,file.txt  
-4,5
```

is equivalent to using:

```
*DEFINE_TABLE  
1000  
R1  
R2  
R3  
*DEFINE_CURVE  
1001  
C1,V11  
C2,V12  
C3,V13  
C4,V14  
*DEFINE_CURVE  
1002  
C1,V21  
C2,V22  
C3,V23  
C4,V24  
*DEFINE_CURVE  
1003  
C1,V31  
C2,V32
```

C3 , V33
C4 , V34

In this case, any scaling applies to the matrix entries before transposing the data, i.e., for row entries the scaled value is

$$\text{Scaled Value} = \text{SROW} \times (\text{R} + \text{OROW}),$$

and for column entries

$$\text{Scaled Value} = \text{SCOL} \times (\text{C} + \text{OCOL})$$

regardless the sign of NROW.

***DEFINE_TARGET_BOUNDARY**

Purpose: This keyword is used to define the desired boundary of a formed part. This boundary provides the criteria used during blank size development. The definitions associated with this keyword are used, exclusively, by the [*INTERFACE_BLANKSIZE_DEVELOPMENT](#) feature.

Point Cards. Include one card for each point in the curve. These points are interpolated to form a closed curve. This input is terminated with *END.

Card 1	1	2	3	4	5	6	7	8	9	10
Variable	X		Y		Z					
Type	E16.0		E16.0		E16.0					
Default	none		none		none					

VARIABLE	DESCRIPTION
X, Y, Z	Location coordinates of a target node.

Remarks:

1. **INTERFACE_BLANKSIZE DEVELOPMENT.** The keyword file specified on the second data card for the [*INTERFACE_BLANKSIZE DEVELOPMENT](#) keyword must contain a ***DEFINE_TARGET_BOUNDARY** keyword.
2. **Example.** A partial keyword input is shown below. Note that the input is in a 3E16.0 FORTRAN format. Also note that the first and last curve points coincide.

```
*KEYWORD
*DEFINE_TARGET_BOUNDARY
-1.83355e+02 -5.94068e+02 -1.58639e+02
-1.80736e+02 -5.94071e+02 -1.58196e+02
-1.78126e+02 -5.94098e+02 -1.57813e+02
-1.75546e+02 -5.94096e+02 -1.57433e+02
-1.72888e+02 -5.94117e+02 -1.57026e+02
      :          :          :
-1.83355e+02 -5.94068e+02 -1.58639e+02
*END
```

3. **IGES to Keyword Format.** Typically, these boundary nodes are obtained from the boundary curves for a final (trimmed) piece or from a draw blank edge at a certain distance outside of the draw beads. LS-PrePost 4.1 can generate the

points for this keyword from IGES data. To use IGES data select *Curve* → *Convert* → *Method (To Keyword)* → *Select *DEFINE_TARGET_BOUNDARY*; pick the curves; and then select *To Key*. To output in keyword format, choose *File* → *Save as* → *Save Keyword As*, and select “Output Version” as “V971_R7”.

Revisions:

This feature is available in LS-DYNA R6 Revision 74560 and later releases.

DEFINE_TRACER_PARTICLES_2D**DEFINE*****DEFINE_TRACER_PARTICLES_2D**

Purpose: Define tracer particles that follow the deformation of a material. This is useful for visualizing the deformation of a part that is being adapted in a metal forming operation. Nodes used as tracer particles should only be used for visualization and not associated with anything in the model that may alter the response of the model, e.g., they should not be used in any elements except those with null materials.

Card 1	1	2	3	4	5	6	7	8
Variable	NSET	PSET						
Type	I	I						
Default	none	0						

VARIABLE**DESCRIPTION**

NSET

The node set ID for the nodes used as tracer particles.

PSET

Optional part set ID. If this part set is specified, only tracer particles in these parts are updated and the others are stationary. If this part set is not specified, all tracer particles are updated.

*DEFINE

*DEFINE_TRANSFORMATION

*DEFINE_TRANSFORMATION

Purpose: Define a transformation for the *INCLUDE_TRANSFORM keyword option. The *DEFINE_TRANSFORMATION command must be defined before the *INCLUDE_TRANSFORM command can be used.

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

Transformation Cards. Include as many cards as necessary. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	OPTION	A1	A2	A3	A4	A5	A6	A7
Type	A	F	F	F	F	F	F	F

Matrix Card 1. Cards 3 and 4 are only included when OPTION = MATRIX.

Card 3	1	2	3	4	5	6	7	8
Variable	M11	M12	M13	M14	M21	M22	M23	M24
Type	F	F	F	F	F	F	F	F

Matrix Card 2. Cards 3 and 4 are only included when OPTION = MATRIX.

Card 4	1	2	3	4	5	6	7	8
Variable	M31	M32	M33	M34	M41	M42	M43	M44
Type	F	F	F	F	F	F	F	F

DEFINE_TRANSFORMATION**DEFINE**

VARIABLE	DESCRIPTION
TRANID	Transform ID
OPTION	For the available options, see the table below.
A1-A7	Parameters. See Table 17-61 below for the available options.

Table 17-61. List of allowed transformations.

OPTION	PARAMETERS	FUNCTION
MIRROR	a1, a2, a3, a4, a5, a6, a7	Reflect about a mirror plane defined to contain the point (a1, a2, a3) having its normal pointing from point (a1, a2, a3) toward (a4, a5, a6). Setting a7 = 1 reflects the coordinate system as well, that this, the mirrored coordinate system uses the left-hand-rule to determine the local z-axis.
MATRIX	Mij in Cards 3 & 4	Direct input of a 4x4 space transformation matrix. A node with coordinates (x, y, z) will be transformed according to $(x, y, z, 1)M$. In most cases $M_{14} = M_{24} = M_{34} = 0$ and $M_{44} = 1.0$. The matrix can be equivalent to other options. For example, TRANSL is equivalent to $M_{41} = a_1, M_{42} = a_2$ and $M_{43} = a_3$, and SCALE is the same as $M_{11} = a_1, M_{22} = a_2$ and $M_{33} = a_3$.
POINT	a1,a2,a3,a4	Define a point with ID, a1, with the initial coordinates a2, a3, and a4.
POS6P	a1, a2, a3, a4, a5, a6	Positioning by 6 points. Affine transformation (rotation and translation, no scaling) given by three start points a1, a2, and a3 and three target points a4, a5, and a6. The six POINTs must be defined before they are referenced. Only 1 POS6P option is permitted within a *DEFINE_TRANSFORMATION definition.
POS6N	a1, a2, a3, a4, a5, a6	Positioning by 6 nodes. Affine transformation (rotation and translation, no scaling) given by three start nodes a1, a2, and a3 and three target nodes a4, a5, and a6. The six nodes must be defined before they are referenced. Only 1 POS6N option is permitted within a *DEFINE_TRANSFORMATION definition.

DEFINE**DEFINE_TRANSFORMATION**

OPTION	PARAMETERS	FUNCTION
ROTATE	a1, a2, a3, a4, a5, a6, a7	Rotate through an angle (deg), a7, about a line with direction cosines a1, a2, and a3 passing through the point with coordinates a4, a5, and a6. If a4 through a7 are zero, then a1 and a2 are the IDs of two POINTs, and a3 defines the rotation angle. The axis of rotation is defined by a vector going from point with ID a1 to point with ID a2.
ROTATE3NA	a1, a2, a3, a4	Rotate through an angle (deg), a4. The axis of rotation is defined by a vector going from the node with ID a1 to the node with ID a2 and passing through the node with ID a3 (a3 could be the same as a1 or a2). The three nodes must be defined before they are referenced. Only 1 ROTATE3NA option is permitted within a *DEFINE_TRANSFORMATION definition.
SCALE	a1, a2, a3	Scale the global <i>x</i> , <i>y</i> , and <i>z</i> coordinates of a point by a1, a2, and a3, respectively. If zero, a default of unity is set.
TRANSL	a1, a2, a3	Translate the <i>x</i> , <i>y</i> , and <i>z</i> coordinates of a point by a1, a2, and a3, respectively.
TRANSL2ND	a1, a2, a3	Translate by distance a3. The direction is defined by a vector going from node with ID a1 to node with ID a2. The two nodes must be defined before they are referenced. If a3 is set to zero, then the distance between nodes a1 and a2 is used directly for the translation. Only 1 TRANSL2ND option is permitted within a *DEFINE_TRANSFORMATION definition.

Each option represents a transformation matrix. When more than one option is used, the transformation matrix defined by MIRROR, SCALE, ROTATE, ROTATE3NA, TRANSL, or TRANSL2ND is applied to the previously defined existing matrix to form the new global transformation matrix. Therefore the ordering of the MIRROR, SCALE, ROTATE, ROTATE3NA, TRANSL, and TRANSL2ND commands is important. We generally recommend first scaling, then rotating, and finally translating the model. When used together with *INCLUDE_TRANSFORM, we advise defining the 6 nodes referred to by POS6N and the *DEFINE_TRANSFORMATION in the same file.

The POINT option in ROTATE provides a means of defining rotations about axes defined by the previous transformations. The coordinates of the two POINTs are transformed by all the transformations up to the transformation where they are referenced. The POINTs must be defined before they are referenced, and their identification numbers are local to each *DEFINE_TRANSFORMATION. The coordinates of a POINT are transformed using all the transformations before it is referenced, not just the transformations between its definition and its reference. To put it another way, while the ordering of the transformations is important, the ordering between the POINTs and the transformations is not important.

NOTE: When *DEFINE_TRANSFORMATION is called from within the target of an *INCLUDE_TRANSFORM keyword, the result will involve stacked transformations.

In the following example, the *DEFINE_TRANSFORMATION command is used 3 times to input the same dummy model and position it as follows:

1. Transformation ID 1000 imports the dummy model (dummy.k) and rotates it 45 degrees about z-axis at the point (0.0,0.0,0.0). Transformation ID 1001 performs the same transformation using the POINT option.
2. Transformation ID 2000 imports the same dummy model (dummy.k) and translates 1000 units in the x direction.
3. Transformation ID 3000 imports the same dummy model (dummy.k) and translates 2000 units in the x direction. For each *DEFINE_TRANSFORMATION, the commands TRANSL, SCALE, and ROTATE are available. The transformations are applied in the order in which they are defined in the file. For instance, transformation ID 1000 in this example would translate, scale and then rotate the model. *INCLUDE_TRANSFORM uses a transformation ID defined by a *DEFINE_TRANSFORMATION command to import a model and perform the associated transformations. It also allows the user upon importing the model to apply offsets to the various entity IDs and perform unit conversion of the imported model.

```
*KEYWORD
*DEFINE_TRANSFORMATION
 1000
$ option &      dx&      dy&      dz&
TRANSL       0000.0      0.0      0.0
$ option &      dx&      dy&      dz&
SCALE        1.00      1.0      1.0
$ option &      dx&      dy&      dz&      px&      py&      pz&
angle&
ROTATE       0.00      0.0      1.0      0.00      0.00      0.0
45.00
*DEFINE_TRANSFORMATION
 1001
POINT          1      0.0      0.0      0.0
```

DEFINE**DEFINE_TRANSFORMATION**

```
POINT           2      0.0      0.0      1.0
ROTATE          1      2      45.0
*DEFINE_TRANSFORMATION
  2000
$ option &      dx&      dy&      dz&
TRANSL         1000.0      0.0      0.0
*DEFINE_TRANSFORMATION
$ tranid &
  3000
$ option &      dx&      dy&      dz&
TRANSL         2000.0      0.0      0.0
*INCLUDE_TRANSFORM
dummy.k
$idnoff &  ideoff&  idpoff& idmoff &  idsoff &  iddoff& idoff &
  0          0          0          0          0          0          0
$ idroff& ilctmf&
  0          0
$ fctmas& fcttim& fctlen& fcttem & incout&
  1.0000    1.0000    1.00      1.0        1
$ tranid &
  1000
*INCLUDE_TRANSFORM
dummy.k
$idnoff &  ideoff&  idpoff& idmoff &  idsoff &  iddoff& idoff &
  1000000  1000000  1000000  1000000  1000000  1000000  1000000
$ idroff& ilctmf&
  1000000  1000000
$ fctmas& fcttim& fctlen& fcttem & incout&
  1.0000    1.0000    1.00      1.0        1
$ tranid &
  2000
*INCLUDE_TRANSFORM
dummy.k
$idnoff &  ideoff&  idpoff& idmoff &  idsoff &  iddoff& idoff &
  2000000  2000000  2000000  2000000  2000000  2000000  2000000
$ idroff& ilctmf&
  2000000  2000000
$ fctmas& fcttim& fctlen& fcttem & incout&
  1.0000    1.0000    1.00      1.0        1
$ tranid &
  3000
*END
```

***DEFINE_TRIM_SEED_POINT_COORDINATES**

Purpose: The keyword facilitates blank trimming during a stamping line die simulation. It allows for the trimming process and inputs to be defined independent of the previous process simulation results and is applicable to shell, solid and laminate.

Card 1	1	2	3	4	5	6	7	8
Variable	NSEED	X1	Y1	Z1	X2	Y2	Z2	
Type	I	F	F	F	F	F	F	
Default	none	0.0	0.0	0.0	0.0	0.0	0.0	

VARIABLE	DESCRIPTION
NSEED	Number of seed points. Maximum value of “2” is allowed.
X1, Y1, Z1	Location coordinates of seed point 1.
X2, Y2, Z2	Location coordinates of seed point 2.

Remarks:

- Associated Keywords.** This keyword is used in conjunction with keywords *ELEMENT_TRIM and *DEFINE_CURVE_TRIM, where variables NSEED1 and NSEED2 should be left as blank. For detailed usage, refer to **Seed Node Definition** section in *DEFINE_CURVE_TRIM.
- Number of Seed Points.** Variable NSEED is set to the number of seed points desired. For example, in a double attached drawn panel trimming, NSEED would equal to 2.
- Example Input Deck.** A partial keyword inputs for a single drawn panel trimming is listed below.

```

*INCLUDE_TRIM
drawn.dynain
*ELEMENT_TRIM           1
*DEFINE_CURVE_TRIM_NEW
$#   TCID      TCTYPE        TFLG       TDIR      TCTOL      TOLN      NSEED
          1            2                  11        0.250
trimlines.iges
*DEFINE_TRIM_SEED_POINT_COORDINATES
$   NSEED      X1          Y1          Z1      X2          Y2          Z2
          1     -271.4      89.13    1125.679
*DEFINE_VECTOR

```

***DEFINE**

***DEFINE_TRIM_SEED_POINT_COORDINATES**

11,0.0,0.0,0.0,0.0,0.0,10.0

Typically, seed point coordinates can be selected from the stationary post in punch home position.

4. **Revision Information.** This feature is available in LS-DYNA R4 Revision 53048 and later releases.

***DEFINE_VECTOR**

Purpose: Define a vector by defining the coordinates of two points.

Card	1	2	3	4	5	6	7	8
Variable	VID	XT	YT	ZT	XH	YH	ZH	CID
Type	I	F	F	F	F	F	F	I
Default	0	0.0	0.0	0.0	0.0	0.0	0.0	global

VARIABLE	DESCRIPTION
VID	Vector ID
XT	X-coordinate of tail of vector
YT	Y-coordinate of tail of vector
ZT	Z-coordinate of tail of vector
XH	X-coordinate of head of vector
YH	Y-coordinate of head of vector
ZH	Z-coordinate of head of vector
CID	Coordinate system ID to define vector in local coordinate system. All coordinates, XT, YT, ZT, XH, YH, and ZH are in respect to CID. EQ.0: global (default).

Remarks:

1. **Numerical Inaccuracies.** The coordinates should differ by a certain margin to avoid numerical inaccuracies.

DEFINE**DEFINE_VECTOR_NODES*****DEFINE_VECTOR_NODES**

Purpose: Define a vector with two nodal points.

Card	1	2	3	4	5	6	7	8
Variable	VID	NODET	NODEH					
Type	I	I	I					
Default	0	0	0					

VARIABLE	DESCRIPTION
VID	Vector ID
NODET	Nodal point to define tail of vector
NODEH	Nodal point to define head of vector

EXAMPLES

The following examples demonstrate the input for these options:

```
*DEFINE_BOX  
  
*DEFINE_COORDINATE_NODES,  
  
*DEFINE_COORDINATE_SYSTEM,  
  
*DEFINE_COORDINATE_VECTOR  
  
*DEFINE_CURVE  
  
*DEFINE_SD_ORIENTATION  
  
*DEFINE_VECTOR commands.
```

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$  
$  
$$$$$ *DEFINE_BOX  
$  
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$  
$  
$ Define box number eight which encloses a volume defined by two corner  
$ points: (-20.0, -39.0, 0.0) and (20.0, 39.0, 51.0). As an example, this  
$ box can be used as an input for the *INITIAL_VELOCITY keyword in which  
$ all nodes within this box are given a specific initial velocity.  
$  
*DEFINE_BOX  
$  
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8  
$ boxid      xmm      xmx      ymn      ymx      zmn      zmx  
     8       -20.0      20.0     -39.0      39.0       0.0      51.0  
$  
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

***DEFINE**

EXAMPLES

EXAMPLES

*DEFINE

***DEFINE**

EXAMPLES