*INCLUDE

The keyword *INCLUDE provides a means of reading independent input files containing model data. The file contents are placed directly at the location of the *INCLUDE line.

INCLUDE_*{OPTION}

*INCLUDE_AUTO_OFFSET

*INCLUDE_COMPENSATION_BEFORE_SPRINGBACK

*INCLUDE_COMPENSATION_BLANK_AFTER_SPRINGBACK

*INCLUDE_COMPENSATION_BLANK_BEFORE_SPRINGBACK

*INCLUDE_COMPENSATION_COMPENSATED_SHAPE

*INCLUDE_COMPENSATION_COMPENSATED_SHAPE_NEXT_STEP

*INCLUDE_COMPENSATION_CURRENT_TOOLS

*INCLUDE_COMPENSATION_CURVE

*INCLUDE_COMPENSATION_DESIRED_BLANK_SHAPE

*INCLUDE_COMPENSATION_NEW_RIGID_TOOL

*INCLUDE_COMPENSATION_ORIGINAL_DYNAIN

*INCLUDE_COMPENSATION_ORIGINAL_RIGID_TOOL

*INCLUDE_COMPENSATION_ORIGINAL_TOOL

*INCLUDE_COMPENSATION_SPRINGBACK_INPUT

*INCLUDE_COMPENSATION_SYMMETRIC_LINES

*INCLUDE_COMPENSATION_TANGENT_CONSTRAINT

*INCLUDE_COMPENSATION_TRIM_CURVE

*INCLUDE_COMPENSATION_TRIM_NODE

*INCLUDE_COMPENSATION_UPDATED_BLANK_SHAPE

*INCLUDE_COMPENSATION_UPDATED_RIGID_TOOL

*INCLUDE_COSIM

*INCLUDE_ISPG

*INCLUDE_MULTISCALE

*INCLUDE_MULTISCALE_SPOTWELD

*INCLUDE_PATH

INCLUDE_STAMPED_*OPTION1_{OPTION2}_{OPTION3}_{OPTION4}

*INCLUDE_STAMPED_PART_SOLID_TO_SOLID

*INCLUDE_TRIM

*INCLUDE_UNITCELL

*INCLUDE_WD_FINAL_PART

*INLUDE_WD_INITIAL_BLANK

*INCLUDE_WD_WELDING_CURVE

***INCLUDE**_*{OPTION}*

Purpose: Include independent input files containing model data.

Available options include:

<BLANK>

BINARY

NASTRAN

TRANSFORM

TRANSFORM_BINARY

The BINARY and TRANSFORM_BINARY options specify that the initial stress file, dynain, is written in a binary format. See the keyword *INTERFACE_SPRINGBACK.

The TRANSFORM and TRANSFORM_BINARY options allow for node, element, and set IDs to be offset and for coordinates and constitutive parameters to be transformed and scaled.

Card Summary:

Card 1a. This card is included if the keyword option is unset (<BLANK>). Include as many of this card as needed. The next keyword ("*") card terminates this input.

FILENAME

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

FILENAME

Card 2a. This card is included if the NASTRAN keyword option is used.

BEAMDF	SHELLDF	SOLIDDF					

Card 2b.1. This card is included if the TRANSFORM keyword option is used.

IDNOFF	IDEOFF	IDPOFF	IDMOFF	IDSOFF	IDFOFF	IDDOFF	

Card 2b.2. This card is included if the TRANSFORM keyword option is used.

IDROFF		PREFIX	SUFFIX				

Card 2b.3. This card is included if the TRANSFORM keyword option is used.

FCTMAS	FCTTIM	FCTLEN	FCTTEM	INCOUT1	FCTCHG		

Card 2b.4. This card is included if the TRANSFORM keyword option is used.

TRANID							

Data Card Definitions:

File name without Keyword Option Card. Include this card if the keyword option is unset (<BLANK>). Multiple file names can be specified (each on its own line(s)) which are processed sequentially. Note that each file name can be multiple lines (see Remark 2). File names are read until the next keyword ("*") card.

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

VARIABLE	DESCRIPTION
FILENAME	File name of file to be included (see Remark 2)

File name with Keyword Option Card. Include this card if the keyword option is set. Only one file name can be specified.

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

VARIABLE	DESCRIPTION
FILENAME	File name of file to be included (see Remark 2)

Nastran Card. Additional card for the NASTRAN keyword option.

Card 2a	1	2	3	4	5	6	7	8
Variable	BEAMDF	SHELLDF	SOLIDDF					
Type	I	I	I					
Default	2	21	18					

VARIABLE	DESCRIPTION
BEAMDF	LS-DYNA beam element type. Defaults to type 2.
SHELLDF	LS-DYNA shell element type. Defaults to type 21.
SOLIDDF	LS-DYNA solid element type. Defaults to type 18.

Transform Card 1. Additional card for TRANSFORM keyword option.

Card 2b.1	1	2	3	4	5	6	7	8
Variable	IDNOFF	IDEOFF	IDPOFF	IDMOFF	IDSOFF	IDFOFF	IDDOFF	
Type	I	I	I	I	I	I	I	

VARIABLE	DESCRIPTION
IDNOFF	Offset to node ID
IDEOFF	Offset to element ID
IDPOFF	Offset to part ID, nodal rigid body ID, constrained nodal set ID, rigidwall ID, and cross section ID (see *DATABASE_CROSS_SECTION)
IDMOFF	Offset to material ID and equation of state ID
IDSOFF	Offset to set ID
IDFOFF	Offset to function ID, table ID, and curve ID
IDDOFF	Offset to any ID defined through *DEFINE, except the FUNCTION, TABLE, and CURVE options (see IDFOFF)

Transform Card 2. Additional card for TRANSFORM keyword option.

Card 2b.2	1	2	3	4	5	6	7	8
Variable	IDROFF		PREFIX	SUFFIX				
Type	I		A	A				

VARIABLE	DESCRIPTION
IDROFF	Used for all offsets except for those listed above
PREFIX	Prefix added to the beginning of the titles/heads defined in the keywords (like *MAT, *PART, *SECTION, *DEFINE, for examples) of the included file. A dot, ".", is automatically added between the prefix and the existing title.
SUFFIX	Suffix added to the end of the titles/heads defined in the keywords of the included file. A dot, ".", is automatically added between the suffix and the existing title.

Transform Card 3. Additional card for TRANSFORM keyword option.

Card 2b.3	1	2	3	4	5	6	7	8
Variable	FCTMAS	FCTTIM	FCTLEN	FCTTEM	INCOUT1	FCTCHG		
Type	F	F	F	A	I	F		

VARIABLE	DESCRIPTION
FCTMAS	Mass transformation factor. For example, FCTMAS = 1000 when the original mass unit is in tons and the new unit is kg.
FCTTIM	Time transformation factor. For example, FCTTIM = .001 when the original time unit is in milliseconds and the new time unit is seconds.
FCTLEN	Length transformation factor
FCTTEM	Temperature transformation factor consisting of a four character flag: FtoC (Fahrenheit to Centigrade), CtoF, FtoK, KtoF, KtoC, and CtoK.

VARIABLE	DESCRIPTION
INCOUT1	Set to 1 for the creation of a file, DYNA.INC, which contains the transformed data. The data in this file can be used in future include files and should be checked to ensure that all the data was transformed correctly.
FCTCHG	Electric charge transformation factor. It currently only applies to piezoelectric material related cards, see *MAT_ADD_PZELECTRIC for details.

Transform Card 4. Additional card for TRANSFORM keyword option.

Card 2b.4	1	2	3	4	5	6	7	8
Variable	TRANID							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
TRANID	Transformation ID. If 0, no transformation will be applied. See *DEFINE_TRANSFORMATION.

Remarks:

1. **Scalability.** To make the input file easy to maintain, this keyword allows the input file to be split into subfiles. Each subfile can again be split into sub-subfiles and so on. This option is beneficial when the input data deck is very large. Consider the following example:

```
*TITLE
full car model
*INCLUDE
carfront.k
*INCLUDE
carback.k
*INCLUDE
occupantcompartment.k
*INCLUDE
dummy.k
*INCLUDE
bag.k
```

```
*CONTACT
⋮
*END
```

Note that the command *END terminates the include file.

The carfront.k file can again be subdivided into rightrail.k, leftrail.k, battery.k, wheel-house.k, shotgun.k, etc.. Each *.k file can include nodes, elements, boundary conditions, initial conditions, and so on.

```
*INCLUDE
rightrail.k
*INCLUDE
leftrail.k
*INCLUDE
battery.k
*INCLUDE
wheelhouse.k
*INCLUDE
shotgun.k
⋮
*END
```

2. **File Name Length Limitations.** File names are limited to 236 characters spread over up to three 80 character lines. When 2 or 3 lines are needed to specify the filename or pathname, end the preceding line with "␣+" (space followed by a plus sign) to signal that a continuation line follows. Note that the "␣+" combination is, itself, part of the 80 character line; hence the maximum number of allowed characters is 78 + 78 + 80 = 236.

3. **NASTRAN Option.** The transformed LS-DYNA deck for *INCLUDE_NASTRAN will be automatically written to file DYNA.INC.

*INCLUDE_AUTO_OFFSET_{OPTION}

Purpose: This particular *INCLUDE keyword offsets node and element IDs to avoid duplication during stamping simulations. In single process stamping simulations the rigid tools often undergo several iterations of modifications, so the node or element IDs comprising the new tools sometimes conflict with other parts of the model. In a multiple process simulation, the IDs of rigid tools from later processes may overlap the IDs of the tools or the formed sheet blank from the previous processes, which makes it difficult to automate the process simulation. This keyword automatically checks for and offsets the duplicate IDs. It works on shell, solid, beam and discrete (*ELEMENT_DISCRETE) elements. The *CONTROL_FORMING_MAXID keyword is related.

Available option includes:

<BLANK>

USER

The USER option allows for specifying offset IDs of the nodes and elements (shells, solids and beams) for the included file.

Note this keyword does not work on the dynain file with stress and strain information. Thus, in a multiple included files case, the dynain file for the blank should always be included first using simply *INCLUDE; see example below.

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

User Set Offset Card. This card is included if and only if the USER option is used.

Card 2	1	2	3	4	5	6	7	8
Variable	NOFFSET	NEOFFSET						
Type	I	I						

VARIABLE	DESCRIPTION
FILENAME	Name of file to be included

VARIABLE	DESCRIPTION
NOFFSET	An offset value for the node IDs of the included file
NEOFFSET	An offset value for the element IDs of the included file

***INCLUDE_AUTO_OFFSET (without option):**

This keyword can be used to offset element and node IDs of the tooling. *This keyword will not offset IDs of meshes with initial stress and strain information*. As such, the sheet blank (including dynain file) should always be included first using the *INCLUDE keyword, followed by *INCLUDE_AUTO_OFFSET to offset tooling mesh IDs which do not have stress and strain information.

Incoming element and node IDs of the rigid tooling mesh files, such as the punch, die, and binder, could be overlapped with each other or overlapped with those on the sheet blank. Multiple *INCLUDE_AUTO_OFFSET keywords can be used to include the punch, die, and binder separately, if desired. For example, four different components of the tooling, the upper die, lower punch, binder and gage pins, can be included and their element and node IDs properly offset after those of a gravity-loaded sheet blank:

```
*INCLUDE
gravity.dynain
*INCLUDE_AUTO_OFFSET
upperdie.k
*INCLUDE_AUTO_OFFSET
lowerpunch.k
*INCLUDE_AUTO_OFFSET
binder.k
*INCLUDE_AUTO_OFFSET
pins.k
```

All of the included meshes can have conflicting node and element IDs starting from 1. Node and element IDs will be offset and reordered in the order of the included files. Included tool files whose node and element IDs do not overlap with those on either the blank or other tools will not be offset or reordered. In many circumstances this feature allows for bypassing a metal forming setup GUI when updating just one or two tooling pieces.

***INCLUDE_AUTO_OFFSET_USER:**

This option gives you more control on the starting node and element IDs of the included files. It applies to shell, solid and beam elements. The following example shows node and element IDs of multiple included files, upper.k, lower.k, binder.k, pins.k, and formed.dynain (formed sheet blank), are offset by 1000000, 2000000, 3000000, 4000000, and 5000000, respectively. As stated above, this option does not allow for the offset of the IDs of the sheet blank with stress and strain information, so formed.dynain file should be included first using just the keyword *INCLUDE. Care should be taken that the offset IDs defined for all the rigid tools do not overlap with those in formed.dynain and also do not overlap each other.

```
*INCLUDE
./formed.dynain
*INCLUDE_AUTO_OFFSET_USER
./upper.k
1000000, 1000000
*INCLUDE_AUTO_OFFSET_USER
./lower.k
2000000, 2000000
*INCLUDE_AUTO_OFFSET_USER
./binder.k
3000000, 3000000
*INCLUDE_AUTO_OFFSET_USER
./pins.k
4000000, 4000000
```

Revision Information:

- This feature is available in SMP and MPP in LS-DYNA Revision 92417. Support for shell elements only.
- Revision 123467 extends to beams (used to model draw beads, for example) and solids.
- Revision 122115: extends to *ELEMENT_DISCRETE.
- Revision 123467: the option USER is available.

*INCLUDE_COMPENSATION

Purpose: This group of keywords allows for the inclusion of stamping tool and blank information for springback compensation. In addition, trim curves from the target geometry can be included for mapping onto the intermediate compensated tool geometry, which can be used for the next compensation iteration. Furthermore, compensation can be done for a localized tool region.

Basic options to perform springback compensation include the following:

*INCLUDE_COMPENSATION_BLANK_BEFORE_SPRINGBACK

*INCLUDE_COMPENSATION_BLANK_AFTER_SPRINGBACK

*INCLUDE_COMPENSATION_DESIRED_BLANK_SHAPE

*INCLUDE_COMPENSATION_COMPENSATED_SHAPE

*INCLUDE_COMPENSATION_CURRENT_TOOLS

*INCLUDE_COMPENSATION_ORIGINAL_TOOL

*INCLUDE_COMPENSATION_COMPENSATED_SHAPE_NEXT_STEP

If springback prediction is performed after trimming, the trimming curves need to be modified based on the boundary change of the blank due to springback. The keyword to do that is:

*INCLUDE_COMPENSATION_TRIM_CURVE

Compensation can be performed locally with

*INCLUDE_COMPENSATION_CURVE

To reduce the number of iterations in springback compensation, the following keywords must be used with *INTERFACE_COMPENSATION_3D_ACCELERATOR:

*INCLUDE_COMPENSATION_ORIGINAL_DYNAIN

*INCLUDE_COMPENSATION_SPRINGBACK_INPUT

If the part has a symmetric condition, this keyword ensures that the compensated rigid tools are also symmetric:

*INCLUDE_COMPENSATION_SYMMETRIC_LINES

If there are regions with undesired mesh quality, the surface can be smoothed locally with the following keyword together with *INTERFACE_COMPENSATION_3D_LOCAL_SMOOTH:

*INCLUDE_COMPENSATION_ORIGINAL_RIGID_TOOL

After compensation, the default file name for the new rigid tool is rigid.new. However, the user can change its name using

*INCLUDE_COMPENSATION_NEW_RIGID_TOOL

After compensation has been completed, if some minor part change is needed, the compensated tool may be modified without redoing the iteration using the following keywords and *INTERFACE_COMPENSATION_3D_PART_CHANGE:

*INCLUDE_COMPENSATION_UPDATED_BLANK_SHAPE

*INCLUDE_COMPENSATION_UPDATED_RIGID_TOOL

During a hot forming process, due to the thickness change of the blank, the gap between the blank and the rigid tools is not homogeneous which will cause inhomogeneity in cooling. The following keyword is used with *INTERFACE_THICKNESS_CHANGE_COMPENSATION to modify the tool surface and ensure a homogenous gap:

*INCLUDE_COMPENSATION_BEFORE_SPRINGBACK

*INCLUDE_COMPENSATION_CURRENT_TOOLS

To ensure tangency continuity in the neighborhood of a trim curve, the following keyword can be used. These nodes can also be generated from *INTERFACE_COMPENSATION_3D_REFINE_RIGID.

*INCLUDE_COMPENSATION_TRIM_NODE

If the user does not want to change the tangency along the punch boundary, the following keyword can be used:

*INCLUDE_COMPENSATION_TANGENT_CONSTRAINT

Full Example of a Springback Compensation:

The following example is for compensation of a localized area, defined by the file curves.k. Trim lines are mapped onto the new compensated rigid tool, with trimcurves.k. Both files which were generated by *LS-PrePost* 4.0 are in the XYZ format. A detailed explanation of each keyword is given in the manual pages related to *INTERFACE_COMPENSATION_3D.

```
*KEYWORD
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+--
*INTERFACE_COMPENSATION_3D
$   METHOD        SL        SF     ELREF      PSID    UNDRCT     ANGLE NLINEAR
         8    10.000     1.000         0         1         0       0.0       1
*INCLUDE_COMPENSATION_BLANK_BEFORE_SPRINGBACK
blank0.k
*INCLUDE_COMPENSATION_BLANK_AFTER_SPRINGBACK
spbk.k
*INCLUDE_COMPENSATION_DESIRED_BLANK_SHAPE
reference0.k
*INCLUDE_COMPENSATION_COMPENSATED_SHAPE
reference1.k
*INCLUDE_COMPENSATION_CURRENT_TOOLS
tools.k
*INCLUDE_COMPENSATION_TRIM_CURVE
trimcurves.k
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$ For compensation of a localize region only, add the following keyword:
*INCLUDE_COMPENSATION_CURVE
curves.k
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
*SET_PART_LIST
$      PSID
         1
$      PID
         3
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+--
*END
```

*INCLUDE_COMPENSATION_BEFORE_SPRINGBACK

Purpose: In a hot forming process, the gap between the blank and rigid tools is not homogeneous due the thickness change of the blank which will cause inhomogeneity in cooling. This keyword when used with *INTERFACE_THICKNESS_CHANGE_COMPENSATION modifies the tool surface and ensures a homogeneous gap. The stamping tool set should be included with *INCLUDE_COMPENSATION_CURRENT_TOOLS. See *INTERFACE_THICKNESS_CHANGE_COMPENSATION for more details.

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

VARIABLE	DESCRIPTION
FILENAME	A dynain file from a forming simulation; the varied, non-homogenous thickness of the sheet blank will be used to change the uniform tool gap.

Revision Information:

This keyword is available as of Revision 106357.

*INCLUDE_COMPENSATION_BLANK_AFTER_SPRINGBACK

Purpose: Include the mesh information in keyword format for the last state (from d3plot) of the springback simulation or the dynain file after springback. This keyword is used with *INTERFACE_COMPENSATION_3D to compensate stamping tool shapes for springback with an iterative method.

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	A80							
Default	spbk.tmp							

VARIABLE	DESCRIPTION
FILENAME	File that includes the nodes and element information with adaptive constraints if they exist for the last state of the springback simulation.

Remarks:

Sheet blanks included using the following related keyword options of *INCLUDE_COMPENSATION must have the same number of nodes and elements for a compensation simulation:

BLANK_AFTER_SPRINGBACK

BLANK_BEFORE_SPRINGBACK

*INCLUDE_COMPENSATION_BLANK_BEFORE_SPRINGBACK

Purpose: Include the mesh information in keyword format for the first state (from d3plot) of the springback simulation or the dynain file after trimming (before springback and with no mesh coarsening). This keyword is used with *INTERFACE_COMPENSATION_NEW to compensate stamping tool shapes for springback with an iterative method.

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	A80							
Default	blank0.tmp							

VARIABLE	DESCRIPTION
FILENAME	File that includes the nodes and element information with adaptive constraints if they exist for the first state of the springback simulation.

Remarks:

Sheet blanks included using the following related keyword options of *INCLUDE_COMPENSATION must have the same number of nodes and elements for a compensation simulation:

BLANK_AFTER_SPRINGBACK

BLANK_BEFORE_SPRINGBACK

*INCLUDE_COMPENSATION_COMPENSATED_SHAPE

Purpose: Include the mesh information in a dynain file for the first iteration (same as INCLUDE_COMPENSATION_DESIRED_BLANK_SHAPE). For the following compensation iterations, this file is obtained from disp.tmp which is generated as an output file during the previous compensation iteration. This keyword is used with *INTERFACE_-COMPENSATION_NEW to compensate stamping tool shapes for springback with an iterative method.

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	A80							
Default	reference1.dat							

VARIABLE	DESCRIPTION
FILENAME	File that includes the nodes and element information with adaptive constraints if they exist for the first compensation iteration.

Remarks:

Sheet blanks included using the following related keyword options of *INCLUDE_COMPENSATION must have the same number of nodes and elements for a compensation simulation:

DESIRED_BLANK_SHAPE

COMPENSATED_SHAPE

*INCLUDE_COMPENSATION_COMPENSATED_SHAPE_NEXT_STEP

Purpose: Enable compensation of tools for the next die process. This keyword is used with *INTERFACE_COMPENSATION_3D_MULTI_STEPS.

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

VARIABLE	DESCRIPTION
FILENAME	File that includes the nodes and element information with adaptive constraints (if adaptive constraints exist for the next die process)

Remarks:

Sheet blanks included using the following related keyword options of *INCLUDE_COMPENSATION must have the same number of nodes and elements for a compensation simulation:

DESIRED_BLANK_SHAPE

COMPENSATED_SHAPE`

COMPENSATED_SHAPE_NEXT_STEP

Revision Information:

This keyword is available as of Revision 61406.

*INCLUDE_COMPENSATION_CURRENT_TOOLS

Purpose: Include the tool mesh information in keyword format. This file is the tool mesh used for the current forming simulation. This keyword is used with *INTERFACE_COMPENSATION_3D to compensate stamping tool shapes for springback with an iterative method.

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	A80							
Default	rigid.tmp							

VARIABLE	DESCRIPTION
FILENAME	File that includes the nodes and element information for the last state of the springback simulation. If the file is named as rigid0.tmp, the elements of the tools get refined along the outline of the part. Draw bead nodes must be included in this file so that they will be modified together with the rigid tools.

*INCLUDE_COMPENSATION_CURVE

Purpose: Perform compensation on localized tooling areas by defining a compensation zone. This keyword allows for die face compensation of a local region in a stamping die.

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

VARIABLE	DESCRIPTION
FILENAME	Name of the keyword file containing x, y, z coordinates of two curves defining the compensation zone, using keywords: *DEFINE_CURVE_COMPENSATION_CONSTRAINT_BEGIN, and, *DEFINE_CURVE_COMPENSATION_CONSTRAINT_END.

Revision Information:

This keyword is available as of Revision 62038.

*INCLUDE_COMPENSATION_DESIRED_BLANK_SHAPE

Purpose: Include the dynain file for the blank after trimming or before springback in the first compensation iteration. This file is the target for compensation, so it never changes in all subsequent compensation iterations. This keyword is used with *INTERFACE_-COMPENSATION_3D to compensate stamping tool shapes for springback with an iterative method.

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	A80							
Default	reference0.dat							

VARIABLE	DESCRIPTION
FILENAME	File that includes the nodes and element information with adaptive constraints if they exist for the blank after trimming in the first compensation iteration.

Remarks:

Sheet blanks included using the following related keyword options of *INCLUDE_COMPENSATION must have the same number of nodes and elements for a compensation simulation:

DESIRED_BLANK_SHAPE

COMPENSATED_SHAPE

*INCLUDE_COMPENSATION_NEW_RIGID_TOOL

Purpose: Change the file name for the new rigid tool following compensation from "rigid.new."

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

VARIABLE	DESCRIPTION
FILENAME	File name for new rigid tool.

Revision Information:

This keyword is available as of Revision 73850.

*INCLUDE_COMPENSATION_ORIGINAL_DYNAIN

Purpose: This keyword is to be used in conjunction with *INTERFACE_COMPENSATION_3D_ACCELERATOR and INCLUDE_COMPENSATION_SPRINGBACK_INPUT for a springback compensation with a faster convergence rate and a simplified user interface. For detailed usage, please refer to the manual pages under *INTERFACE_COMPENSATION_3D_{*OPTION*}.

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

VARIABLE	DESCRIPTION
FILENAME	The dynain file name from an LS-DYNA simulation that contains model information, adaptive constraints, and stress and strain tensor information.

Revision Information:

This keyword is available as of Revision 61264.

*INCLUDE_COMPENSATION_ORIGINAL_RIGID_TOOL

Purpose: Locally smooth mesh in regions with undesired mesh quality.

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

VARIABLE	DESCRIPTION
FILENAME	Keyword file that contains the meshes of the rigid tools.

Remarks:

This keyword is used together with *INTERFACE_COMPENSATION_NEW_LOCAL_-SMOOTH and *SET_NODE_LIST_SMOOTH to smooth local areas of distorted meshes of a tooling surface. Details can be found in the manual pages for *INTERFACE_COMPENSATION_NEW_LOCAL_SMOOTH.

Revision Information:

This keyword is available as of Revision 73850.

*INCLUDE_COMPENSATION_ORIGINAL_TOOL

Purpose: Obtain a smoother mesh for the addendum and binder region for the current compensation by using the original tool mesh (of better quality) instead of the last compensated tool mesh (maybe distorted). This method reduces the accumulative error in mesh extrapolation outside of the trim lines. Details can be found in the manual pages for *INTERFACE_COMPENSATION_3D.

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

VARIABLE	DESCRIPTION
FILENAME	Original tool (without any compensation) mesh file that contains node and element information in keyword format.

Revision Information:

This keyword is a available as of Revision 82701.

*INCLUDE_COMPENSATION_SPRINGBACK_INPUT

Purpose: This keyword is to be used in conjunction with *INTERFACE_COMPENSATION_3D_ACCELERATOR and INCLUDE_COMPENSATION_ORIGINAL_DYNAIN for a springback compensation with a faster convergence rate and a simplified user interface. For detailed usage, please refer to the manual pages under *INTERFACE_COMPENSATION_3D_{*OPTION*}.

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

VARIABLE	DESCRIPTION
FILENAME	File name of springback simulation input deck for the baseline iteration zero simulation.

Revision Information:

This keyword is available as of Revision 61264.

*INCLUDE_COMPENSATION_SYMMETRIC_LINES

Purpose: Ensure the compensated rigid tools are symmetric if the part has a symmetrical condition. This keyword is used with *INTERFACE_COMPENSATION_3D for METHOD = 7 or 8.

Card 1	1	2	3	4	5	6	7	8
Variable	SYMID	SYMXY	X0	Y0				
Type	I	I	F	F				
Default	1	none	0.0	0.0				

VARIABLE	DESCRIPTION
SYMID	ID of the symmetric condition being defined.
SYMXY	Code defining symmetric boundary conditions: EQ.1: symmetric about y-axis. EQ.2: symmetric about x-axis.
X0, Y0	Coordinates of a point on the symmetric plane.

Example:

In a complete keyword input example below, part set ID 1 is being compensated with symmetric boundary condition about x-axis. The symmetric plane passes a point with coordinates of $x = 101.5$, and $y = 0.0$.

```
*KEYWORD
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$*INTERFACE_COMPENSATION_NEW
$ Method = 8 changes the binder; Method = 7 binder/P.O. no changes.
*INTERFACE_COMPENSATION_NEW
$   METHOD        SL        SF     ELREF      PSID    UNDRCT     ANGLE   NLINEAR
         7    10.000     1.000         2         1         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
./currenttools.k
```

```
*INCLUDE_COMPENSATION_SYMMETRIC_LINES
$    SYMID     SYMXY        X0        Y0
         1         2     101.5       0.0
$ SYMXY = 2: symmetric about X-axis
*SET_PART_LIST
$      PSID
         1
$      PID
         1
*END
```

Revision Information:

This keyword is available as of revision 63618. It was updated in Revision 83711.

*INCLUDE_COMPENSATION_TANGENT_CONSTRAINT

Purpose: Maintain tangency in the part boundary (along trim curves) between compensated and uncompensated areas of the tool. It also preserves the tangential transition off the rigid tool area at the trim curves to the addendum part of the tool. This keyword was replaced by the TANGENT variable under *INTERFACE_COMPENSATION_3D.

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

VARIABLE	DESCRIPTION
UFLAG	Tangency flag: EQ.1: Maintain tangency between compensated and uncompensated parts of the tool and maintain tangential transition off the rigid tool area at the trim curves to the addendum part of the tool.

*INCLUDE_COMPENSATION_TRIM_CURVE

Purpose: Map trim curves of current tools onto the compensated tools. This keyword is used when springback predication is performed after trimming.

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

VARIABLE	DESCRIPTION
FILENAME	Name of the keyword file containing x, y, z coordinates as defined using keyword *DEFINE_CURVE_TRIM_3D (only TCTYPE = 0 or 1 is supported). This option is used to map the trim curve to the new, compensated tooling mesh for next iterative simulation.

Remarks:

If the trimming curve is in IGES format, a new file, geocur.trm, will be generated after trimming. The file contains XYZ data of the trim curves under the keyword *DEFINE_-CURVE_TRIM_{OPTIONS}, which is used for the compensation run. Note that the variable TCTYPE in the keyword must be set to “0” (or “1”) for the compensation. Length of lines everywhere in the compensated part are calculated according to springback amounts (including the die expansion factors, therefore no die expansion needs to be included in the NC machining of the compensated tooling). The manual pages for *INTERFACE_BLANKSIZE contain a procedure to convert IGES files to XYZ format in LS-PrePost.

Revision Information:

This keyword is available as of Revision 60398.

*INCLUDE_COMPENSATION_TRIM_NODE

Purpose: Ensure tangency continuity in the neighborhood of a trim curve. These nodes can also be generated from *INTERFACE_COMPENSATION_3D_REFINE_RIGID.

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	A80							
Default	bndnd0.tmp							

VARIABLE	DESCRIPTION
FILENAME	File that contains a list of nodes (in type I10 per line) along part trim curves to ensure tangency during compensation. This file can be generated from *INTERFACE_COMPENSATION_3D_-REFINE_RIGID.

Revision Information:

This keyword is available as of Revision 114206.

*INCLUDE_COMPENSATION_UPDATED_BLANK_SHAPE

Purpose: After compensation has been done, modify the compensated tool without the need to re-do the iterations to compensate the tool. This keyword must be used with *INCLUDE_COMPENSATION_UPDATED_RIGID_TOOL and *INTERFACE_COMPENSATION_3D_PART_CHANGE. Note that these options are intended only for small part changes that do not substantially affect the amount of springback. More details can be found in the manual pages for *INTERFACE_COMPENSATION_3D_PART_CHANGE.

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	A80							
Default	updatedpart.tmp							

VARIABLE	DESCRIPTION
FILENAME	Updated blank shape mesh that has been formed (or trimmed) to the shape based on the new tool geometry.

Revision Information:

This keyword is available as of Revision 82698.

*INCLUDE_COMPENSATION_UPDATED_RIGID_TOOL

Purpose: After compensation has been done, modify the compensated tool without the need to re-do the iterations to compensate the tool. This keyword must be used with *INCLUDE_COMPENSATION_UPDATED_BLANK_SHAPE and *INTERFACE_COMPENSATION_3D_PART_CHANGE. Note that these options are intended only for small part changes that do not substantially affect the amount of springback. More details can be found in the manual pages for *INTERFACE_COMPENSATION_3D_PART_CHANGE.

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME							
Type	A80							
Default	newrigid.tmp							

VARIABLE	DESCRIPTION
FILENAME	Updated rigid tool mesh.

Revision Information:

This keyword is available as of Revision 82698.

*INCLUDE_COSIM

Purpose: Specify an include file giving the coupling interface across two models in different scales (global and local) running on an MPI-based co-simulation. The input deck for each model should contain this keyword. The included file for each model gives the interface(s) for that model.

This feature requires MPP LS-DYNA version R13.1, R14.0, or newer.

The method invoked with this keyword differs from the method used with *INCLUDE_-MULTISCALE in that it imposes strong coupling. In the weak coupling method, the large-scale model imposes kinematic constraints on the small-scale model at the coupling interface and drives its deformation. The failure of the large-scale representative beams is determined by the small-scale analysis. The strong coupling method invoked with this keyword, in contrast, is a fully concurrent simulation across two scales. The large-scale model imposes the kinematic constraints on the small-scale model and obtains the kinetic response in return. Therefore, this method does not need a simplified representation of the small-scale model in the large-scale model.

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

VARIABLE	DESCRIPTION
FILENAME	Name of a keyword input file that contains the coupling information, which includes sets of segments/solids in the global scale model and sets of nodes in the local scale one. See Remark 1 and the example below.

Remarks:

1. **Overview of multiscale coupling method.** This keyword is for specifying the coupling interface to exchange kinematic and kinetic information between global (large-scale) and local (small-scale) models running on two MPP jobs. The global model is usually a large-scale structure, such as a full car model or printed circuit board (PCB). The local model should be much smaller in dimension. For instance, it could be spot welds, rivets, or solders. The smaller

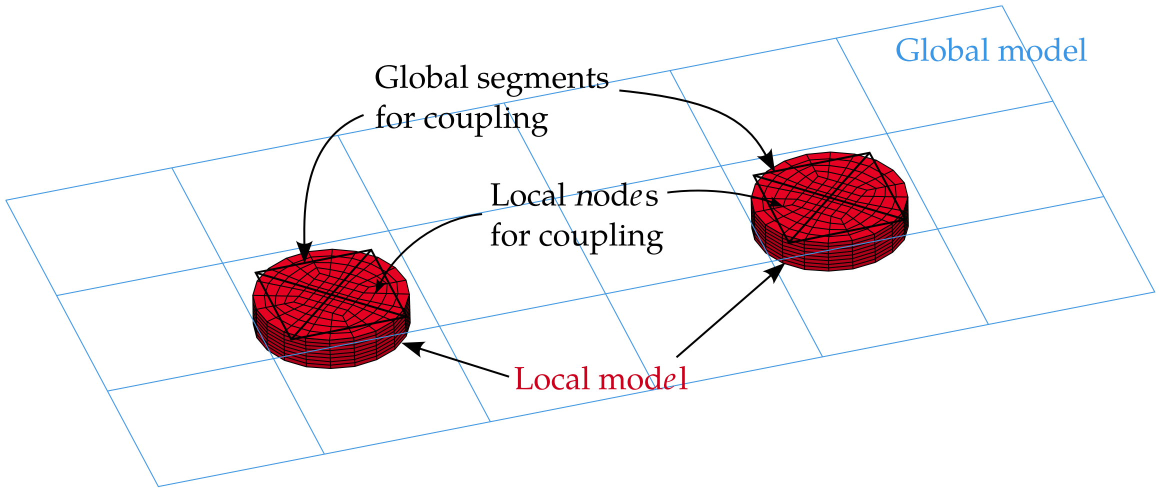

Figure 27-1. Example of tied contact coupling

dimension leads to a smaller mesh and time step size in explicit analysis. To accelerate the computation across global and local scales, you use two input files and run two MPP jobs simultaneously with different time step sizes. The synchronization of time-stepping information across the coupling interface is performed automatically at every time step of the global model. The time step size of the local model is adjusted accordingly to guarantee the numerical consistency and accuracy of state variables at the coupling interface in the spatial and temporal domains.

For the global model, a segment set (see *SET_SEGMENT) in the included file specifies the interface, except in the case of the solid-in-solid immersion coupling algorithm (see item c below). For solid-in-solid immersion, a solid set (see *SET_SOLID) defines the interface in the global model. For the local model, a node set (see *SET_NODE) determines the interface. These two sets must have the same set ID (SID) to indicate how to couple the global and local models. Note that the included files can contain multiple coupling interfaces (see the example below).

Three types of coupling algorithms are currently implemented:

a) *Tied contact*. With tied contact, the set of segments in the global model drives the motion of the set of nodes in the local model by imposing kinematic constraints on nodal translational DOFs. The local model returns the constraint forces interpolated onto the nodes of the set of global segments. To invoke this type of coupling, set the flag ITS to 1 for both the segment and node sets.

b) *Solid-in-shell immersion*. With solid-in-shell immersion, the local solid model occupies the same space as the global shell model at the coupling interface. The node set in the local model follows both the translational and rotational motion of the shell segment set in the global model, while the nodes of the

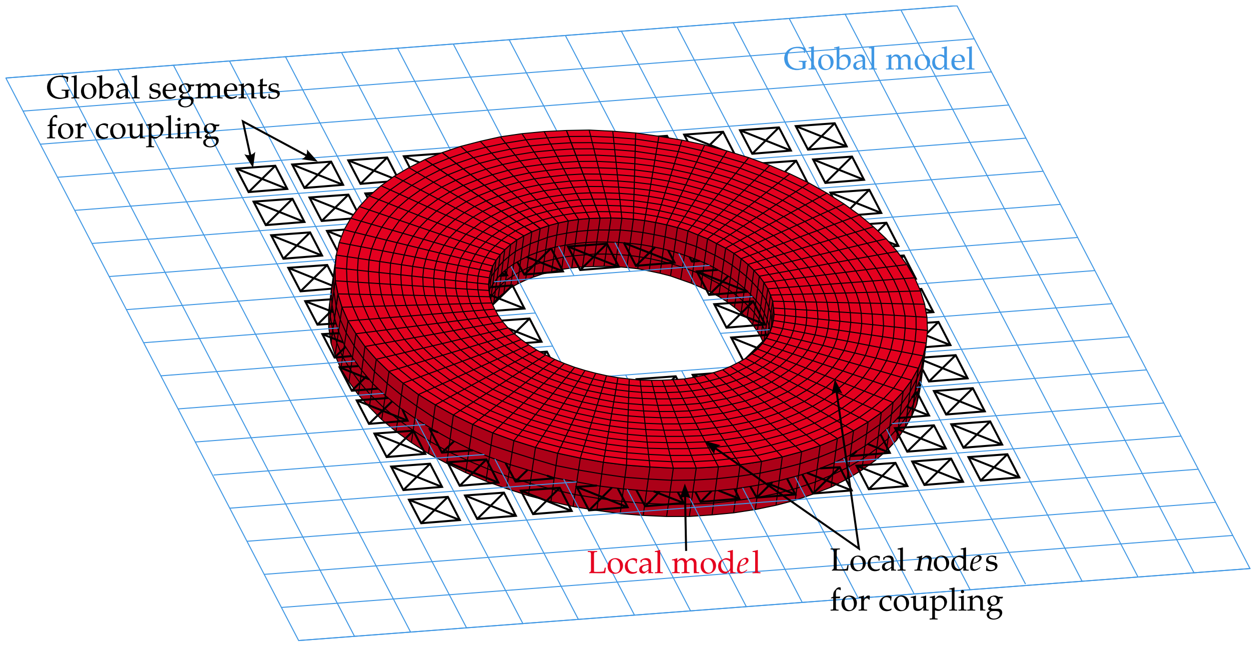

Figure 27-2. Example of solid-in-shell immersed coupling

global segments obtain the constraint forces and moments in return. To invoke this type of coupling, set the flag ITS to 2 for both the segment and node sets.

c) *Solid-in-solid immersion.* With solid-in-solid immersion, the local solid model occupies the same space as the global solid model at the coupling interface. The node set in the local model follows the translational motion of the solid set in the global model, while the nodes of the global solid elements obtain the constraint forces in return. In this case, set the flag ITS to 3 for both the solid and node sets.

2. **Running two MPP jobs.** The specifics of running two MPP jobs are installation-dependent. The following is an example of running two MPP jobs for two different scales:

   mpirun -np 36 mppdyna i=input.k ncsp=24 jobid=jid

   The "ncsp" flag informs the LS-DYNA software that the local model runs on a separate job using 24 out of the total 36 MPI processes. The main input file of the local model must have the same name as the global model with the addition of the prefix cs_. For example, the local model's main input file for the example above needs to be named cs_input.k. The main input files of both the global and local models must be in the same working folder. In practice, you can create an appfile file with the following lines:

```
-np 36 mppdyna i=input.k ncsp=24 jobid=jid
```

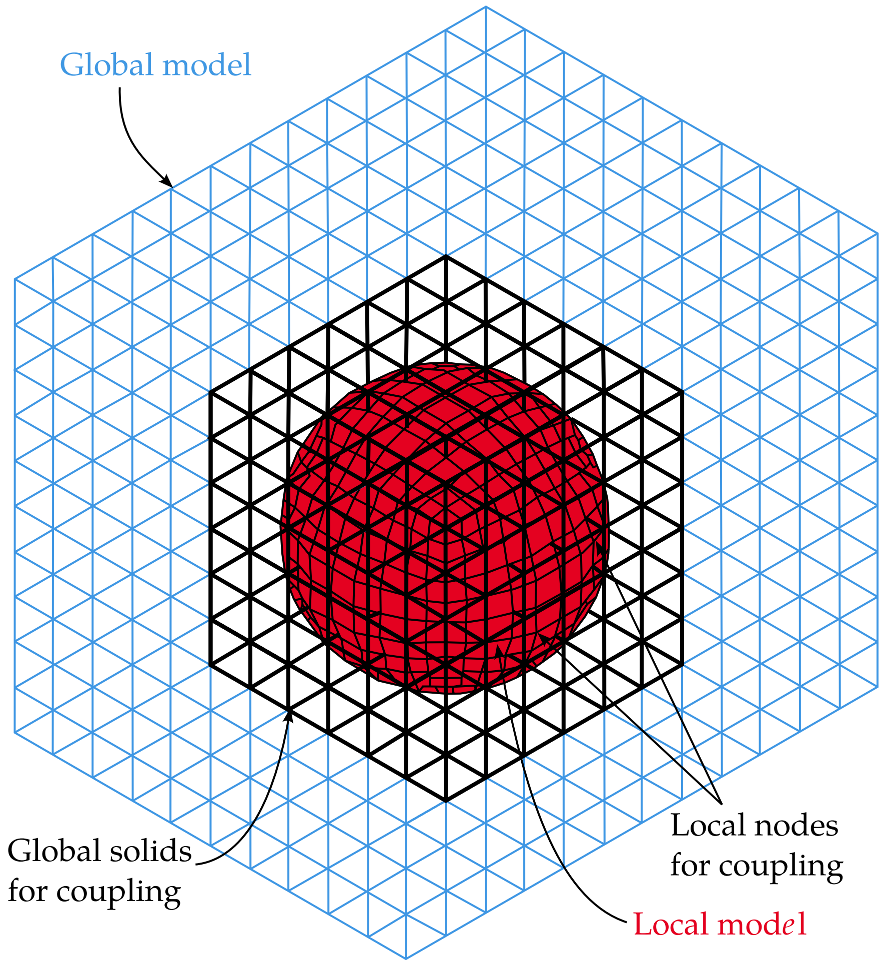

Figure 27-3. Example of solid-in-solid immersion coupling

Then, for Linux, run the problem with mpirun -f appfile. For Windows, use mpiexec -configfile appfile.

The message file contains a category called "Two-scale Cosim" at normal termination. This timing information under this category gives the CPU time spent on the data exchange between two jobs through MPI communication. For a good load balance between the two jobs, the number of MPI processes for the two jobs needs to be adjusted according to the time step size difference and other factors, such as the number of elements and contacts.

All the output and temporary files are separate for the two jobs. cs_ is prefixed to all the files of the local job run to avoid conflicts with the files from the global job.

3. **Termination time.** The global and local files must have the same termination time specified with *CONTROL_TERMINATION.

4. **d3plot output interval.** The time interval of d3plot output set in both input decks must be the same to have a time-consistent post-processing database.

Example:

input.k:

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$  * Main input file for the global model
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
*KEYWORD
$
*CONTROL_TERMINATION
$   ENDTIM    ENDCYC     DTMIN    ENDENG    ENDMAS     NOSOL
      1e-4
$
*DATABASE_D3PLOT
      1e-6
$
*INCLUDE_COSIM
global_cosim.k
$
$$ Other keyword cards including control, mesh, contact, …
…
$
*END
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

global_cosim.k:

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$  * Define coupling interface in the global model
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
*KEYWORD
$
*SET_SEGMENT
$      SID       DA1       DA2       DA3       DA4    SOLVER       ITS
        22       0.0       0.0       0.0       0.0MECH               1
$       N1        N2        N3        N4        A1        A2        A3        A4
         8         9        12        11       0.0       0.0       0.0       0.0
        36        37        40        39       0.0       0.0       0.0       0.0
…
$
*SET_SEGMENT
$      SID       DA1       DA2       DA3       DA4    SOLVER       ITS
        11       0.0       0.0       0.0       0.0MECH               1
$       N1        N2        N3        N4        A1        A2        A3        A4
        23        21        25        27       0.0       0.0       0.0       0.0
        47        45        49        51       0.0       0.0       0.0       0.0
…
$
*SET_SOLID
$#     SID    SOLVER
         3MECH               3
$#      K1        K2        K3        K4        K5        K6        K7        K8
       372       373       374       375       376       377       402       403
```

```
       404       405       406       407       432       433       434       435
…
$
*END
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$


cs_input.k:

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$  * Main input file for the local model
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
*KEYWORD
$
*CONTROL_TERMINATION
$   ENDTIM    ENDCYC     DTMIN    ENDENG    ENDMAS     NOSOL
      1e-4
$
*DATABASE_D3PLOT
      1e-6
$
*INCLUDE_COSIM
local_cosim.k
$
$$ Other keyword cards including control, mesh, contact, …
…
$
*END
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$


local_cosim.k:

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$$  * Define coupling interface in the local model
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
*KEYWORD
$
*SET_NODE_LIST
$      SID       DA1       DA2       DA3       DA4    SOLVER       ITS
        11       0.0       0.0       0.0       0.0MECH               1
$       N1        N2        N3        N4        N5        N6        N7        N8
         1         2         3         4         5         6         7         8
         9        10        11        12        13        14        15        16
…
$
*SET_NODE_LIST
$      SID       DA1       DA2       DA3       DA4    SOLVER       ITS
        22       0.0       0.0       0.0       0.0MECH               1
$       N1        N2        N3        N4        N5        N6        N7        N8
       929       930       931       932       933       934       935       936
       937       938       939       940       941       942       943       944
…
$
*SET_NODE_LIST
$      SID       DA1       DA2       DA3       DA4    SOLVER       ITS
         3       0.0       0.0       0.0       0.0MECH               3
$       N1        N2        N3        N4        N5        N6        N7        N8
      4250      4604      4508      4058      4106      4292      4171      4165
      4177      4183      4004      4381      4658      4322      4082      4164
```

```
…

$
*END
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

*INCLUDE_ISPG

Purpose: Include a file that provides the model for the ISPG parts. These parts are used for simulating incompressible fluids and coupled with structural parts. For ease of defining ISPG models, these models use keywords for solids (*ELEMENT_SOLID, *NODE, *PART). However, the ISPG models are separate from the structural models (meaning different sets of numbering from structural models for the nodes and elements used by the ISPG models). Currently, only 4-noded tetrahedral elements can be used for the ISPG models.

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

VARIABLE	DESCRIPTION
FILENAME	Name of file containing the information about the ISPG models. Only one file can be included for each keyword (“*”) card.

Remarks:

The following keywords can be defined in the ISPG model file included with this keyword:

*ISPG_ASCII_CPL

*ISPG_ASCII_SUMFORC

*ISPG_BOUNDARY_CONVECTION_SET

*ISPG_BOUNDARY_SYMMETRIC

*ISPG_CONTROL_ADAPTIVITY

*ISPG_CONTROL_D3ISPG

*ISPG_CONTROL_IMPLICIT

*ISPG_CONTROL_MPP

*ISPG_CONTROL_SOLUTION

*ISPG_DAMPING_SURF

*ISPG_DEFINE_MERGER_MPP

*ISPG_INITIAL_TEMPERATURE_SET

*ISPG_LOAD_GRAVITY

*MAT_ISPG_CARREAU

*MAT_ISPG_CROSS_CASTRO_MACOSKO

*MAT_ISPG_CROSSMODEL

*MAT_ISPG_ISO_NEWTONIAN

*MAT_THERMAL_ISPG

*SECTION_ISPG

*KEYWORD

*ELEMENT_SOLID

*NODE

*PART

*END

Note that *SECTION_FPD and *MAT_IFPD are not supported for ISPG models defined in this way.

*INCLUDE_MULTISCALE

Purpose: Provide a detailed local model for a coupled multiscale simulation. The local model is coupled to heuristic beams in the large-scale model. Failure of the heuristic beams is determined through this local model. This type of simulation allows including details necessary for failure without using an infeasible time step. See the Remarks section for how this type of simulation should be set up and performed. This keyword is an extension of *INCLUDE_MULTISCALE_SPOTWELD for a more general local model.

This capability is available *only* in the MPP version of LS-DYNA.

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

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

VARIABLE	DESCRIPTION
ID	ID for this multiscale local model. This ID is used in the keyword *DEFINE_MULTISCALE. Any unique integer will do.
FILENAME	Name of file from which to read the local model definition.

Remarks:

With the multiscale feature, heuristic beams models are replaced with zoomed-in geometrically and constitutively correct continuum models, which, in turn are coupled to the large-scale calculation without reducing the time step. Because the detailed local models are run in a separate process, they can run at a much smaller time step without slowing down the rest of the simulation. A brief outline of their use looks like this:

- Create one (or more) detailed local model(s) of their beams and include these definitions into the large-scale model using the keyword *INCLUDE_MULTISCALE. In the large-scale model, the beams must be on the microscale model side of a tied interface.
- Indicate which beams should be coupled to these models with *DEFINE_MULTISCALE.
- When MPP-DYNA is started, a special (MPI dependent) invocation is required to run in a "multiple program" mode. Effectively, two separate instances of MPP-DYNA are started together, one to run the full model and a separate instance to run the local models.
- As the macroscale process runs, each cycle it communicates to the microscale process deformation information for the area surrounding each coupled beam. The microscale process imposes this deformation on the detailed solid local models, computes a failure flag for each, and communicates this back to the macroscale process.
- The coupled beams in the macroscale process have their failure determined solely by these failure flags.

The file referred to on the *INCLUDE_MULTISCALE card should contain one generic instance of a detailed local model. For each coupled beam in the main model, a specific instance of this beam will be generated which is translated, rotated, and scaled to match the solid local model to which it is coupled. In this way, many beams can be coupled with only a single *INCLUDE_MULTISCALE. The included file should contain everything required to define a detailed local model, such as *MAT and *PART definitions, any required *DEFINE_CURVEs, etc., as well as *NODE and *ELEMENT definitions. For the translation and scaling to work properly, we make the following assumptions about the detailed local model:

- It consists entirely of solid elements.
- The z-axis is aligned with the coupled local model in the main model, with $z = 0$ and $z = 1$ at the two ends of the local model.
- The cross-sectional area of the local model in the xy-plane is equal to 1.
- That portion of the "top" and "bottom" of the local model that are coupled are identified using a single *SET_NODE_LIST card.
- One *BOUNDARY_COUPLED card referencing the *SET_NODE_LIST of the boundary nodes is required. It must specify a coupling type of 2 and a coupling program of 1.
- The detailed local model can include multiple parts.
- The detailed local model does not support *INCLUDE cards.

Failure of the local model is determined topologically. Any element of the detailed local model having all four nodes of one of its faces belonging to the *SET_NODE_LIST of tied nodes is classified as a "tied" element. The "tied" elements are partitioned into two disjoint sets: the "top," and "bottom". When there is no longer a complete path from any "top" to any "bottom" element (where a "path" passes through non-failed elements that share a common face), then this detailed model has failed. Note that this places some restrictions on the *SET_NODE_LIST and element geometry, namely that some "tied" elements exist, and the set of "tied" elements consists of exactly two disjoint subsets.

The specifics of launching a multi-program MPI program are installation-dependent. The idea behind running a coupled model is to run one set of MPI ranks as if running a normal MPP-DYNA job, such as:

```
mpirun -np 18 mppdyna i=input.k nmsp=12
```

The nmsp flag informs LS-DYNA that the local model runs in a separate job using 12 out of the total 18 MPI processors.

The main instance knows to look for the microscale model (because of the presence of the *INCLUDE_MULTISCALE card) and runs the main model. The "microscale" instance runs all the detailed local models. Due to the nature of the coupling, the main model cannot progress when the detailed local models are being processed, nor can the detailed local models run while the main model is being computed. From a processor efficiency standpoint, it therefore makes sense to run as many microscale processes as macroscale processes and run them on the same CPUs, so that each processing core has one microscale and one macroscale process running on it. But you don't have to – the processes are independent and you can have any number of either.

*INCLUDE_MULTISCALE_SPOTWELD

Purpose: Define a type of multiscale spot weld to be used for coupling and for modeling of spot weld failure.

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

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

VARIABLE	DESCRIPTION
TYPE	TYPE for this multiscale spot weld. This type is used in the keyword *DEFINE_SPOTWELD_MULTISCALE. Any unique integer will do.
FILENAME	Name of file from which to read the spot weld definition.

Remarks:

This capability is available *only* in the MPP version of LS-DYNA.

With the multiscale spot weld feature, heuristic spot weld models are replaced with zoomed-in geometrically and constitutively correct continuum models, which, in turn are coupled to the large-scale calculation without reducing the time step. In some respects, multiscale models are similar to the "hex spot weld assemblies," capability but more general in terms of their geometry. Because the spot weld models are run in a separate process, they can run at a much smaller time step without slowing down the rest of the simulation. A brief outline of their use looks like this:

- You create one (or more) detailed models of your spot welds and include these definitions into your model using the keyword *INCLUDE_MULTISCALE_-SPOTWELD. In the large-scale model, the spot weld must be on the tracked (SURFA) side of a tied interface.
- You indicate which beam (or hex assembly) spot welds should be coupled to these models with the keyword *DEFINE_SPOTWELD_MULTISCALE
- When MPP-DYNA is started, a special (MPI dependent) invocation is required in order to run in a "multiple program" mode. Effectively, two separate instances of MPP-DYNA are started together, one to run the full model (macroscale) and a separate instance to run the spot welds (microscale).
- As the macroscale process runs, each cycle it communicates deformation information for the area surrounding each coupled spot weld to the microscale process. The microscale process imposes this deformation on the detailed spot welds, computes a failure flag for each, and communicates this back to the macroscale process.
- The coupled spot welds in the macroscale process have their failure determined solely by these failure flags.

The file referred to on the *INCLUDE_MULTISCALE_SPOTWELD card should contain one generic instance of a detailed spot weld. For each coupled spot weld in the main model, a specific instance of this spot weld will be generated which is translated, rotated, and scaled to match the spot weld to which it is coupled. In this way, many spot welds can be coupled with only a single *INCLUDE_MULTISCALE_SPOTWELD. The included file should contain everything required to define the spot weld, such as *MAT and *PART definitions, any required *DEFINE_CURVEs, etc., as well as *NODE and *ELEMENT definitions. For the translation and scaling to work properly, we assume the following about the spot weld model:

- It consists entirely of solid elements.
- The z-axis is aligned with the coupled spot weld in the main model, with $z = 0$ and $z = 1$ at the two ends of the spot weld.
- The cross-sectional area of the spot weld in the xy-plane is equal to 1.
- The portion of the "top" and "bottom" of the spot weld that are coupled are identified using a single *SET_NODE_LIST card.
- One *BOUNDARY_COUPLED card referencing the *SET_NODE_LIST of the boundary nodes is required. It must specify a coupling type of 2 and a coupling program of 1.
- The spot weld model does not support *INCLUDE cards.

Failure of the microscale model is determined topologically. Any element of the spot weld having all four nodes of one of its faces belonging to the *SET_NODE_LIST of tied nodes is classified as a "tied" element. The "tied" elements are partitioned into two disjoint sets: "top" and "bottom". When there is no longer a complete path from any "top" to any "bottom" element (where a "path" passes through non-failed elements that share a common face), then the spot weld has failed. Note that this places some restrictions on the *SET_NODE_LIST and element geometry, namely that some "tied" elements exist, and the set of "tied" elements consists of exactly two disjoint subsets.

The specifics of launching a multi-program MPI program are installation dependent. But the idea behind running a coupled model is that you want to run one set of MPI ranks as if you were running a normal MPP-DYNA job, such as:

mpirun –np 4 mppdyna i=**input.k** memory=**200m** p=**pfile**

and a second set with just the command line argument "microscale" (no input file):

mpirun –np 4 mppdyna microscale memory=**100m** p=**pfile**

The main instance knows to look for the microscale instance (because of the presence of the *INCLUDE_MULTISCALE_SPOTWELD card) and will run the main model. The "microscale" instance will run all the detailed spot weld models. Due to the nature of the coupling, the main model cannot progress when the detailed spot welds are being processed, nor can the detailed spot welds run while the main model is being computed. From a processor efficiency standpoint, it therefore makes sense to run as many microscale processes as macroscale processes, and run them on the same CPUs, so that each processing core has one microscale and one macroscale process running on it. But you don't have to: the processes are independent, and you can have any number of either.

INCLUDE_PATH_*{OPTION}

Available options include:

<BLANK>

RELATIVE

Purpose: Define directories in which to look for include files. LS-DYNA's default behavior is to search for files in the local directory first. If an include file is not found and the filename has no path, LS-DYNA will search for it in all the directories defined by *INCLUDE_PATH.

When the RELATIVE option is used, all directories are relative to the location of the input file. For example, if "i=/home/test/problems/input.k" is given on the command line, and the input contains

```
*INCLUDE_PATH_RELATIVE
includes
../includes
```

then the two directories /home/test/problems/includes and /home/test/includes will be searched for include files.

Pathname Card. Directory paths are read until the next keyword ("*") card is encountered. A directory path can have up to 236 characters (see Remark 2).

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

VARIABLE	DESCRIPTION
PATHNAME	Directory path. If the RELATIVE keyword option is used, this directory is relative to the input file.

Remarks:

1. **Pathname Length Limitations.** Pathnames are limited to 236 characters spread over up to three 80 character lines. When 2 or 3 lines are needed to specify the pathname, end the preceding line with "␣+" (space followed by a plus sign) to signal that a continuation line follows. Note that the "␣+" combination is, itself,

part of the 80 character line; hence the maximum number of allowed characters is $78 + 78 + 80 = 236$.

INCLUDE_STAMPED_*OPTION1_{OPTION2}_{OPTION3}_{OPTION4}

Purpose: Map the plastic strain and thickness distribution of a shell part or part set from a stamping simulation (source part or part) to a shell part or part set in a crash model (target part or part set). This keyword only applies to shell elements. As of R15, it can also map stamping information from a standard finite element model to IGA shell (*IGA_SHELL) parts.

Available options for *OPTION1* include:

PART

SET

For PART a stamped part is mapped while for SET a stamped part set is mapped.

OPTION2 only applies when *OPTION1* is PART:

<BLANK>

SET

If SET is used, PID will be a part set ID. All the parts included in this set will be considered in this mapping. Note that *INCLUDE_STAMPED_PART_SET and *INCLUDE_-STAMPED_SET are equivalent keyword names, meaning they call the same subroutines in LS-DYNA.

The available options for *OPTION3* are:

<BLANK>

MATRIX

If MATRIX is used, LS-DYNA directly reads the transformation matrix with a translation vector instead of using orientation nodes to create the transformation.

The available options for *OPTION4* are:

<BLANK>

INVERSE

INVERSE can only be included if using the MATRIX keyword option. If used, LS-DYNA inverts the matrix. This keyword option is useful for the case where you have the transformation from the crash part to the stamped part.

When using STAMPED_SET or STAMPED_PART_SET, the target is a part set. Between the stamped part and the crash part, note the following points:

1. The outer boundaries of the parts do not need to match because only the regions of the crash part that overlap the stamped part are initialized.

2. Arbitrary mesh patterns are assumed.

3. Element formulations can change.

4. Unless using the MATRIX keyword option, three nodes on each part are used to reorient the stamped part for the mapping of the strain and thickness distributions. After reorientation, the three nodes on each part should approximately coincide. Otherwise, directly input the transformation.

5. The number of in-plane integration points can change.

6. The number of through-thickness integration points can change. Full interpolation is used.

7. The node and element IDs between the stamped part and the crash part do not need to be unique.

Card Summary:

Card 1. This card is required.

FILENAME

Card 2. This card is required.

PID	THICK	PSTRN	STRAIN	STRESS	INCOUT		RMAX

Card 3a. Include this card if *not* using the MATRIX keyword option.

N1S	N2S	N3S	N1C	N2C	N3C	TENSOR	THKSCL

Card 3b.1. Include this card if using the MATRIX keyword option.

R11	R12	R13	XP	TENSOR	THKSCL		

Card 3b.2. Include this card if using the MATRIX keyword option.

R21	R22	R23	YP				

Card 3b.3. Include this card if using the MATRIX keyword option.

R31	R32	R33	ZP				

Card 4. This card is optional.

ISYM	IAFTER	PERCELE	IORTHO		ISRCOUT	MTYPE	

Card 5. This card is optional.

X01	Y01	Z01					

Card 6. This card is optional.

X02	Y02	Z02	X03	Y03	Z03		

Data Card Definitions:

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

VARIABLE	DESCRIPTION
FILENAME	Name of file to be included; see Remark 2. This is the dynain file containing the metal stamping result.

Card 2	1	2	3	4	5	6	7	8
Variable	PID	THICK	PSTRN	STRAIN	STRESS	INCOUT		RMAX
Type	I	I	I	I	I	I		F
Default	none	0	0	0	0	0		20.0

VARIABLE	DESCRIPTION
PID	Part or part set ID of crash part/parts for remapping

VARIABLE	DESCRIPTION
THICK	Thickness remap: EQ.0: Map thickness EQ.1: Do not map thickness EQ.2: Average value inside a circle defined by RMAX
PSTRN	Plastic strain remap: EQ.0: Map plastic strain EQ.1: Do not map plastic strain EQ.2: Average value inside a circle defined by RMAX
STRAIN	Strain remap: EQ.0: Map strains EQ.1: Do not map strains
STRESS	Stress tensor remap: EQ.0: Map stress tensor and history variables EQ.1: Do not map stress tensor, only history variables EQ.2: Do not map stress tensor or history variables EQ.-1: Map stress tensor in an internal large format (binary files) EQ.-3: Do not map stress tensor in an internal large format, only history variables (binary files)
INCOUT	Save mapped data: EQ.1: Save the mapped data for the part/part set (PID) to a file called dyna.inc. This option is useful for when the mapped data may be required in a future simulation. EQ.2: Save the mapped data for the specified part or part set (PID) to a file called dynain_xx (xx is the part or part set ID). EQ.3: Save the mapped data for the specified part or part set (PID) to a file called nastran_xx (in nastran format). xx is the part or part set ID. Note that INCOUT does *not* account for changes in the number of in-plane or through-thickness integration points.

VARIABLE	DESCRIPTION
RMAX	Search radius. LS-DYNA remaps history variables from the mesh of the stamped part to the mesh of the crash part with a spatial tolerance of RMAX. If an element in the crash part lies within RMAX of the stamped part, data will be mapped to that element. If set less than 0.001, RMAX automatically assumes the default value of 20.

Nodal Orientation Card. Include this card if the MATRIX option is *not* used.

Card 3a	1	2	3	4	5	6	7	8
Variable	N1S	N2S	N3S	N1C	N2C	N3C	TENSOR	THKSCL
Type	I	I	I	I	I	I	I	F
Default	0	0	0	0	0	0	0	1.0
Remarks	1	1	1	1	1	1	3	

VARIABLE	DESCRIPTION
N1S	First of 3 nodes needed to reorient the stamped part or part set
N2S	Second of 3 nodes needed to reorient the stamped part or part set
N3S	Third of 3 nodes needed to reorient the stamped part or part set
N1C	First of 3 nodes needed to reorient the crash model part or part set
N2C	Second of 3 nodes needed to reorient the crash model part or part set
N3C	Third of 3 nodes needed to reorient the crash model part or part set
TENSOR	Tensor remap: EQ.0: Map tensor data from history variables. EQ.1: Do not map tensor data from history variables
THKSCL	Thickness scale factor

Transformation Matrix Card 1. Include this card for the MATRIX keyword option.

Card 3b.1	1	2	3	4	5	6	7	8
Variable	R11	R12	R13	XP	TENSOR	THKSCL		
Type	F	F	F	F	I	F		
Default	0	0	0	0	0	1.0		

Transformation Matrix Card 2. Include this card for the MATRIX keyword option.

Card 3b.2	1	2	3	4	5	6	7	8
Variable	R21	R22	R23	YP				
Type	F	F	F	F				
Default	0	0	0	0				

Transformation Matrix Card 3. Include this card for the MATRIX keyword option.

Card 3b.3	1	2	3	4	5	6	7	8
Variable	R31	R32	R33	ZP				
Type	F	F	F	F				
Default	0	0	0	0				

VARIABLE	DESCRIPTION
R*ij*	Components of the transformation matrix (see Remark 1)
XP, YP, ZP	Translational distance (see Remark 1)
TENSOR	Tensor remap: EQ.0: Map tensor data from history variables. EQ.1: Do not map tensor data from history variables

VARIABLE	DESCRIPTION
THKSCL	Thickness scale factor

This card is optional.

Card 4	1	2	3	4	5	6	7	8
Variable	ISYM	IAFTER	PERCELE	IORTHO		ISRCOUT	MTYPE	
Type	I	I	F	I		I	I	

VARIABLE	DESCRIPTION
ISYM	Symmetric switch EQ.0: No symmetric mapping EQ.1: *yz* plane symmetric mapping EQ.2: *zx* plane symmetric mapping EQ.3: *zx* and *yz* planes symmetric mapping EQ.4: User defined symmetric plane mapping
IAFTER	Mirroring sequence switch EQ.0: Generate a symmetric part before transformation EQ.1: Generate a symmetric part after transformation
PERCELE	Percentage of elements that should be mapped for the simulation to proceed (default = 0); otherwise an error termination occurs. See Remark 5.
IORTHO	Location of the material direction cosine in the array of history variables of an orthotropic material. See Remark 4.
ISRCOUT	Optional output of stamped part after transformation(s) EQ.0: No output is written. NE.0: Keyword output file "srcmsh_<ISRCOUT>" is created.
MTYPE	Expected material model. If given, the target part must have the material model given by MTYPE. Otherwise, an error termination occurs. Use the material number for MTYPE. This feature is

VARIABLE	DESCRIPTION
	helpful for detecting an unintended replacement of the material model when mapping history variables.

This card is optional.

Card 5	1	2	3	4	5	6	7	8
Variable	X01	Y01	Z01					
Type	F	F	F					

This card is optional.

Card 6	1	2	3	4	5	6	7	8
Variable	X02	Y02	Z02	X03	Y03	Z03		
Type	F	F	F	F	F	F		

VARIABLE	DESCRIPTION
X01, Y01, Z01	First point in the symmetric plane (required if ISYM ≠ 0)
X02, Y02, Z02	Second point in the symmetric plane
X03, Y03, Z03	Third point in the symmetric plane

Remarks:

1. **Reorienting the result of a stamping simulation.** The target mesh must be read in before including the stamped part with this keyword.

   If the MATRIX keyword is not used, N1S, N2S, N3S, N1C, N2C, and N3C are used for transforming the stamped part to the crash part, such that it is in the same position as the crash part. If the stamped part is in the same position as the crash part, then N1S, N2S, N3S, N1C, N2C, and N3C can all be set to 0. Note that if these 6 nodes are input as 0, LS-DYNA will not transform the stamped part. With the MATRIX keyword option, a transformation matrix with a translation vector reorients the part instead of nodes.

When symmetric mapping is used (ISYM is nonzero), the three points should not be in one line. If ISYM = 1, 2, or 3, only the first point, namely (X01, Y01, Z01), is needed. If ISYM = 4, all three points are needed.

2. **File name length limitations.** Filenames are limited to 236 characters spread over up to three 80-character lines. When 2 or 3 lines are needed to specify the filename or pathname, end the preceding line with "␣+" (space followed by a plus sign) to signal that a continuation line follows. Note that the "␣+" combination is, itself, part of the 80-character line; hence the maximum number of allowed characters is 78 + 78 + 80 = 236.

3. **Mapping material data for springback.** Certain material models (notably Material 190) have tensor data stored within the history variables. Within material subroutines this data is typically stored in element local coordinate systems. In order to properly map this information between models it is necessary to have the tensor data present on the *INITIAL_STRESS_SHELL card and have it stored in global coordinates. During mapping the data is then converted into the local coordinate system of the crash mesh. This data can be dumped into the dynain file that is created at termination time if the parameter FTENSR is set to 0 on the *INTERFACE_SPRINGBACK_LSDYNA card. Currently, the only material model that supports the mapping of element history tensor data is Material 190.

4. **IORTHO.** If IORTHO is set, correct mapping between non-matching meshes is invoked for the directions of orthotropic materials. A list of appropriate values for several materials is given here:

IORTHO.EQ.1:	materials 23, 122, 157, 234
IORTHO.EQ.3:	materials 22, 33, 36, 133, 189, 233, 243
IORTHO.EQ.4:	material 59
IORTHO.EQ.6:	materials 58, 104, 158
IORTHO.EQ.8:	materials 54, 55
IORTHO.EQ.9:	material 39
IORTHO.EQ.10:	material 82
IORTHO.EQ.13:	materials 2, 86, 103

5. **Mapping mismatch.** Sometimes, during mapping, the two meshes (stamping mesh and crash mesh) do not match exactly, and therefore, not all elements of the new mesh obtain results from the old mesh. The message file contains the total number of crash elements that are or are not mapped. By default (PERCELE = 0), the calculation continues even when no elements are mapped.

PERCELE > 0 allows you to define the minimum percentage of required mapped elements for the calculation to proceed. If a percentage less than PERCELE is mapped, the calculation stops with an error termination.

INCLUDE_STAMPED_PART_SOLID_TO_SOLID_*{OPTION}

The available options are:

<BLANK>

GENERAL

Purpose: Map the final stress and strain tensors, history variables, and plastic strain of a solid part in one model (source part) to the corresponding solid part in a second model (target part). The total thickness of the target part is adjusted as necessary to match the final thickness of the source part. This feature was designed for the situation where the source part is a thick blank from a forming analysis and the target part is in a vehicle crash model. The mesh refinements of the two parts may differ. Differences in the spatial location of the target part and source part are allowed (see fields on Card 3). This keyword has several limitations which are discussed in the next paragraph and in Remark 2.

The default mapping uses the top surface to find the matching elements in each layer. In this case, it requires the solid model to be extruded from a shell part and the part to have the same number of layers in each location. Advantageously, this method can find the corresponding layer for mapping. Thus, this mapping feature applies only to 8-noded hexahedral and 6-noded pentahedral elements because it requires consistent, through-thickness orientation of solid element normals. When the solid part is not extruded, the default algorithm does not work. The keyword option GENERAL invokes an algorithm that finds the closest element in the source and maps the information. Thus, the part does not have to have the same number of layers in each location and can include tetrahedral elements. However, this approach might not find the right layer in the mapping.

INCLUDE_STAMPED_PART_SOLID_TO_SOLID_*{OPTION} *must* be the final keyword in the second input deck.

Card Summary:

Card 1. This card is required.

FILENAME

Card 2. This card is required.

PID	THICK	PSTRN	STRAIN	STRESS			

Card 3. This card is required.

N1SORC	N2SORC	N3SORC	N1TRGT	N2TRGT	N3TRGT		

Data Card Definitions:

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

VARIABLE	DESCRIPTION
FILENAME	File name of the dynain file created by *INTERFACE_SPRINGBACK_LSDYNA in the source model.

Card 2	1	2	3	4	5	6	7	8
Variable	PID	THICK	PSTRN	STRAIN	STRESS			
Type	I	I	I	I	I			
Default	none	0	0	0	0			

VARIABLE	DESCRIPTION
PID	Part ID of the target part onto which the source model's results in FILENAME will be mapped. LT.0: Part ID of the target part is |PID| and the normals of the target part are flipped before mapping. A negative PID would be used if the target part's normals were oriented exactly opposite those of the source part.
THICK	Flag to map part thickness. The thickness direction is determined from the element normals, hence the need for consistency in the element normals. The thicknesses of the target part elements are adjusted so the target part thickness matches the source part thickness at any given location. Currently, this variable is hardwired so that thickness mapping is always on.
PSTRN	Flag to map effective plastic strain. Currently setting this flag with any integer will map the effective plastic strain, and there is no other option.

VARIABLE	DESCRIPTION
	EQ.0: Map effective plastic strain. NE.0: Do not map effective plastic strain.
STRAIN	Flag to map the strain tensor. Currently setting this flag with any integer will map the tensorial strains, and there is no other option available. Note "STRFLG" in *DATABASE_EXTENT_BINARY must be set to "1" for output to d3plot as well as dynain files. EQ.0: Map strain tensor. NE.0: Do not map strain tensor.
STRESS	Flag to map stress tensor. Currently setting this flag with any integer will map the stresses and history variables, and there is no other option available. Only the history variables included in the dynain file specified by FILENAME are mapped; see "NSHV" in *INTERFACE_SPRINGBACK_LSDYNA for control of history variable output to dynain. EQ.0: Map stress tensor and history variables. NE.0: Do not map stress tensor and history variables.

Card 3	1	2	3	4	5	6	7	8
Variable	N1SORC	N2SORC	N3SORC	N1TRGT	N2TRGT	N3TRGT		
Type	I	I	I	I	I	I		
Default	0	0	0	0	0	0		

VARIABLE	DESCRIPTION
N1SORC	First of 3 nodes needed to reorient the source part. No transformation if undefined.
N2SORC	Second of 3 nodes needed to reorient the source part. No transformation if undefined.
N3SORC	Third of 3 nodes needed to reorient the source part. No transformation if undefined.

VARIABLE	DESCRIPTION
N1TRGT	First of 3 nodes needed to reorient the target part. No transformation if undefined.
N2TRGT	Second of 3 nodes needed to reorient the target part. No transformation if undefined.
N3TRGT	Third of 3 nodes needed to reorient the target part. No transformation if undefined.

Remarks:

1. **Element normals.** Unless using the GENERAL keyword option, the solid element normals in the source model and the target model part must be consistent with each other and in the through-thickness direction of the part. The first three nodes of the solid determine the bottom face and the element normal is the normal to that face using the right-hand rule. Solid element normals can be displayed and modified using *EleTol* → *Normal* → *Solid* in LS-PrePost.

2. **Geometry limitations.** This keyword was designed with a particular application in mind: mapping results from a forming analysis of a thick blank. It is not designed to handle situations where the target part's geometry is dissimilar from the source part's geometry, such as where the target part has a hole but the source part does not. See *INITIAL_LAG_MAPPING for a more general solid-to-solid mapping capability.

Example:

The following keyword example maps the stress and strain tensors, history variables, and effective plastic strain from a stamped (source) solid blank mesh drawn.dynain with PID 1, onto a coarser mesh (target) s1.k with PID 2.

The mapping may be checked from the dynain file created after a single time step (note that the termination time is set to 0.0 for this purpose).

```
*KEYWORD
*PARAMETER_EXPRESSION
I elform         2
*INTERFACE_SPRINGBACK_LSDYNA
1,100
*DATABASE_EXTENT_BINARY
$#   NEIPH     NEIPS    MAXINT    STRFLG    SIGFLG    EPSFLG    RLTFLG    ENGFLG
       100       100         8         1         1         1         1         1
$#  CMPFLG    IEVERP    BEAMIP     DCOMP      SHGE     STSSZ    N3THDT   IALEMAT
         0         0         0         1         1         1         2         1
$# NINTSLD   PKP_SEN      SCLP     HYDRO     MSSCL     THERM    INTOUT    NODOUT
         0         0       1.0         0         0         0
$#    DTDT    RESPLT     NEIPB     QUADR     CUBIC
```

```
         0         0         0         0         0
*SET_PART
1
1
*CONTROL_TERMINATION
     0.000
*DATABASE_BINARY_D3PLOT
      0.50
$---------------------------------------------------------------------
*PART
PID 1 is of a source mesh (e.g. stamping dynain file)
$      PID       SID       MID
         1         1         1
PID 2 is a target mesh (no stress/strains, with normals consistently aligned)
         2         1         1
$---------------------------------------------------------------------
*MAT_PIECEWISE_LINEAR_PLASTICITY
$      MID        RO         E        PR      SIGY      ETAN      FAIL      TDEL
         1  7.83E-09  2.07E+05     0.333
$        C         P      LCSS      LCSR        VP
       0.0       0.0        11         0       0.0
$     EPS1      EPS2      EPS3      EPS4      EPS5      EPS6      EPS7      EPS8

$      ES1       ES2       ES3       ES4       ES5       ES6       ES7       ES8

*DEFINE_CURVE
        11
    0.0000000000E+00    3.8000500000E+02
    2.0000000000E-03    3.8854500000E+02
    1.5000000000E+00    8.3929800000E+02
$---------------------------------------------------------------------
*SECTION_SOLID
$    SECID    ELFORM      SHRF       NIP     PROPT   QR/IRID     ICOMP
         1   &elform
$---------------------------------------------------------------------
*INCLUDE
$ coarser solid mesh file with no stress/strain:
s1.k
*INCLUDE_STAMPED_PART_SOLID_TO_SOLID
$ dynain file name from stamping simulation (source); native file should be used.
../drawn.dynain
$      PID     THICK     PSTRN    STRAIN    STRESS
$  PID here is PID of the target mesh
        -2         0         0         0         0
$   N1SORC    N2SORC    N3SORC    N1TRGT    N2TRGT    N3TRGT

*END
```

*INCLUDE_TRIM

Purpose: The naive procedure to trim a part imported using the *INCLUDE keyword involves: (1) including the part using *INCLUDE; (2) designating the part for trimming using *CONTROL_FORMING_TRIMMING; and (3) defining trim curves using *DEFINE_TRIM_CURVE. By replacing *INCLUDE in step (1) with this keyword, namely, *INCLUDE_TRIM, LS-DYNA is directed to filter the included part while reading in the include file (as opposed to after reading in), thereby saving memory that would otherwise be allocated to store the (immediately) discarded data. This feature has been developed in conjunction with the *Ford Motor Company Research and Advanced Engineering Laboratory*.

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

VARIABLE	DESCRIPTION
FILENAME	File name of the part to be trimmed

Remarks:

Similar to the trimming method with *INCLUDE, the name of the file to be trimmed must be put into the FILENAME field. Unlike the *INCLUDE method, all information about the included part (such as stress and strain) should be in the named file (as opposed to split across several files). *INCLUDE_TRIM *cannot* be used unless the included file contains stress and strain information. Namely, the dynain file must contain *INITIAL_STRESS_SHELL and *INITIAL_STRAIN_SHELL cards; otherwise the *INCLUDE keyword should be used instead.

For solid element trimming, the keyword *INCLUDE_TRIM *must* be used. In other words, the dynain file from a previous process (for example, a forming simulation) must be included in a main trim file using *INCLUDE_TRIM (not *INCLUDE). Refer to the *DEFINE_CURVE_TRIM manual entry for more details.

For example, a drawn panel from a previous simulation can be included in a current simulation for trimming as follows,

```
*INCLUDE_TRIM
Drawnpanel.dynain
*CONTROL_FORMING_TRIMMING
      ⋮
*DEFINE_CURVE_TRIM_3D
```

```
        ⋮
*CONTROL_ADAPTIVE_CURVE
        ⋮
```

Performance Improvements:

Referring to the table below (parts courtesy of the *Ford Motor Company*), compared with *INCLUDE, the *INCLUDE_TRIM keyword reduces memory requirements by more than 50%. Levels of CPU time reductions exceed in some cases 50%.

Performance Improvements

		Roof	Hood Inr	B-Plr	Fender	BSA Otr	Door Otr	Wheel House (2 in 1)	Boxside Otr
#Element		410810	1021171	351007	189936	380988	315556	261702	1908369
CPU	old	7m26s	10m20s	3m11s	2m6s	5m45s	4m27s	2m52s	27m31s
	new	4m	9m18s	2m56s	1m22s	4m54s	3m35s	2m30s	13m59s
Memory (MW)	old	282	616	221	119	233	217	157	1150
	new	112	383	117	50	130	114	75	539

Revision Information:

This feature is available in LS-DYNA Revision 62207 or later releases, where the output of strain tensors for the shells is included. Prior Revisions do not include strain tensors for the shells.

*INCLUDE_UNITCELL

Purpose: This card creates a unit cell model with periodic boundary conditions using *CONSTRAINED_MULTIPLE_GLOBAL.

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

Card 2	1	2	3	4	5	6	7	8
Variable	INPT	OUPT	NEDOF					
Type	I	I	I					
Default	0	0	0					
Remarks	1		2					

Card 3	1	2	3	4	5	6	7	8
Variable	DX	DY	DZ	NEX	NEY	NEZ	NNPE	TOL
Type	F	F	F	I	I	I	I	F
Default	1.0	1.0	1.0	1	1	1	8	↓

Card 4	1	2	3	4	5	6	7	8
Variable	NOFF	EOFF	PNM					
Type	I	I	I					
Default	none	none	none					

Card 5	1	2	3	4	5	6	7	8
Variable	CNX	CNY	CNZ					
Type	I	I	I					
Default	none	none	none					

Node ID Cards. Included as many cards as needed. See Remark 2. Input is terminated at the next keyword ("*") card

Card 6	1	2	3	4	5	6	7	8
Variable	ECNX	ECNY	ECNZ					
Type	I	I	I					
Default	none	none	none					

VARIABLE	DESCRIPTION
FILENAME	Name of the file containing the information about the unit cell
INPT	Type of input: EQ.0: Read *NODE information from the include file and add periodic boundary conditions to the include file. EQ.1: Create a unit cell mesh with periodic boundary conditions, and output to the include file.

VARIABLE	DESCRIPTION
OUPT	Type of output: EQ.1: Create a new main keyword file where the keyword *INCLUDE_UNITCELL is replaced by *INCLUDE with the include file name.
NEDOF	Number of extra nodal degrees of freedom (DOFs) for user-defined element. In the current implementation, the limit of NEDOF is 15.
DX	Length of the unit cell in the x-direction
DY	Length of the unit cell in the y-direction
DZ	Length of the unit cell in the z-direction
NEX	Number of elements along the x-direction
NEY	Number of elements along the y-direction
NEZ	Number of elements along the z-direction
NNPE	Number of nodes per element. The current implementation supports only 4-node tetrahedron or 8-node hexahedron elements.
TOL	Tolerance for searching for each pair of nodes in the periodic positions to create the periodic boundary conditions. This tolerance may be needed because numerical errors in the mesh can cause the coordinates of the pairs of nodes to *not* be exactly in the periodic positions. The default tolerance is computed based on the size of unit cell.
NOFF	Offset of node IDs
EOFF	Offset of element IDs
PNM	Part ID
CNX	Node ID of the 1st control point for the constraint in the x direction
CNY	Node ID of the 2nd control point for the constraint in the y direction
CNZ	Node ID of the 3rd control point for the constraint in the z direction
ECNX	Node ID of extra control point for the constraint in x direction of 3 extra nodal DOFs

VARIABLE	DESCRIPTION
ECNY	Node ID of extra control point for the constraint in y direction of 3 extra nodal DOFs
ECNZ	Node ID of extra control point for the constraint in z direction of 3 extra nodal DOFs

Remarks:

1. **Include File Field.** If INPT = 0, the geometry and discretization information for the unit cell are from the include file. In this case, the parameters set on Cards 3 and 4 are ignored.

2. **Extra Degrees of Freedom.** The extra degrees of freedom (DOFs) specified by NEDOF > 0 are represented by extra nodes with regular x, y and z DOFs. When NEDOF = 7, for example, the following chart shows the mapping from the extra DOFs to the regular ones of extra nodes:

Extra Node #	Extra DOFs	Regular DOFs
1	1	x
	2	y
	3	z
2	4	x
	5	y
	6	z
3	7	x

In this case, 3 control points for x, y, and z directions, respectively, need to be defined for each extra node.

*INCLUDE_WD

The *INCLUDE_WD keywords are needed during a weld line development calculation caused by including *INTERFACE_WELDLINE_DEVELOPMENT in the input deck. This feature outputs either the initial or final predicted welding curve. This feature requires the following *INCLUDE_WD keywords listed below in the order that they must appear in the input deck following *INCLUDE_WELDLINE_DEVELOPMENT:

*INCLUDE_WD_INITIAL_BLANK

*INCLUDE_WD_FINAL_PART

*INCLUDE_WD_WELDING_CURVE

*INCLUDE_WD_FINAL_PART

Purpose: Include the keyword format file that consists of geometry information for the final formed blank. This file is read to perform a weld line development calculation. This keyword must be used with *INTERFACE_WELDLINE_DEVELOPMENT and must follow *INCLUDE_WD_INITIAL_BLANK in the keyword deck. See *INTERFACE_WELDLINE_DEVELOPMENT for more details.

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

VARIABLE	DESCRIPTION
FILENAME	File name of the final formed blank in keyword format, consisting of *NODE and *ELEMENT_SHELL information

*INCLUDE_WD_INITIAL_BLANK

Purpose: Include the keyword format file that defines the geometry information for the initial sheet blank. This file is read to perform a weld line development calculation. This keyword must be used with and must directly follow *INTERFACE_WELDLINE_DEVELOPMENT (see there for more details).

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

VARIABLE	DESCRIPTION
FILENAME	File name of the initial sheet blank in keyword format, consisting of *NODE and *ELEMENT_SHELL information.

*INCLUDE_WD_WELDING_CURVE

Purpose: Include the keyword format file that defines the welding curve (see *DEFINE_-CURVE_TRIM_3D) to perform a weld line development calculation. This keyword must be used with *INTERFACE_WELDLINE_DEVELOPMENT. Depending on the value of IOPTION on *INTERFACE_WELDLINE_DEVELOPMENT, this file defines either the final or initial welding curve. This keyword must follow *INCLUDE_WD_FINAL_PART directly. See *INTERFACE_WELDLINE_DEVELOPMENT for more details.

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

VARIABLE	DESCRIPTION
FILENAME	File name of the welding curve in keyword format (see *DEFINE_-CURVE_TRIM_3D). If IOPTION = 1 on *INTERFACE_WELDLINE_DEVELOPMENT, it should define the final welding curve. If IOPTION = -1, it should define the initial welding curve.