*INTERFACE

Interface definitions may be used to define surfaces, nodal lines, and nodal points for which the time histories of nodal coordinates and if applicable (*INTERFACE_LINKING_NODE/EDGE) and available (beams, shells), nodal rotations, are saved at some user specified frequency. This data may then be used in subsequent analyses as an interface ID in the *INTERFACE_LINKING_DISCRETE_NODE as constraining nodes, in *INTERFACE_LINKING_SEGMENT as constraining segments and in *INTERFACE_LINKING_EDGE as the constraining edge for a series of nodes. This capability is especially useful for studying the detailed response of a small member in a large structure.

For the first analysis, the member of interest need only be discretized sufficiently that the displacements and velocities on its boundaries are reasonably accurate. After the first analysis is completed, the member can be finely discretized in the region bounded by the interfaces. Finally, the second analysis is performed to obtain highly detailed information in the local region of interest. When beginning the first analysis, specify a name for the interface segment file using the Z = parameter on the LS-DYNA execution line. When starting the second analysis, the name of the interface segment file created in the first run should be specified using the L = parameter on the LS-DYNA command line. Following the above procedure, multiple levels of sub-modeling are easily accommodated. The interface file may contain a multitude of interface definitions so that a single run of a full model can provide enough interface data for many component analyses. The interface feature represents a powerful extension of LS-DYNA's analysis capabilities. The keyword cards for this purpose are:

*INTERFACE_COMPENSATION_3D

*INTERFACE_COMPONENT_FILE

INTERFACE_COMPONENT_*OPTION

INTERFACE_LINKING_DISCRETE_NODE_*OPTION

*INTERFACE_LINKING_EDGE

*INTERFACE_LINKING_FILE

*INTERFACE_LINKING_FORCES

*INTERFACE_LINKING_NODE

*INTERFACE_LINKING_SEGMENT

INTERFACE_SPRINGBACK_*OPTION1_OPTION2

Interface definitions may also be employed to define soil-structure interfaces in earthquake analysis involving non-linear soil-structure interaction where the structure may be non-linear, but the soil outside the soil-structure interface is assumed to be linear. Free-field earthquake ground motions are required only at the soil-structure interface for such analysis. The keyword cards for this purpose are:

*INTERFACE_SSI

*INTERFACE_SSI_AUX

*INTERFACE_SSI_AUX_EMBEDDED

*INTERFACE_SSI_STATIC

As of R13, two interface keywords are available for performing a two-stage SPG analysis. The first keyword is for specifying that the current analysis is the first stage while the second keyword is for specifying the analysis is the second stage. These keywords are:

*INTERFACE_SPG_1

*INTERFACE_SPG_2

*INTERFACE_ACOUSTIC

Purpose: Create an interface file of boundary motions that can be used by *BOUNDARY_ACOUSTIC_INTERFACE for weakly coupled acoustic fluid-structure interaction in SSD analyses (see *CONTROL_IMPLICIT_SSD_DIRECT).

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

VARIABLE	DESCRIPTION
IFID	Segment set ID of structural faces for this interface. This structural IFID can be used by *BOUNDARY_ACOUSTIC_INTERFACE.

Remarks:

1. **Acoustic Interface Filename.** The motions of the structural segments will be written to the LSDB database "baifac.db" for every SSD solution step invoked with *CONTROL_IMPLICIT_SSD_DIRECT. Multiple interfaces from a single analysis may be saved in one interface file if they reference different IFID.

INTERFACE_BLANKSIZE_*OPTION

Available options include:

DEVELOPMENT

INITIAL_TRIM

INITIAL_ADAPTIVE

SCALE_FACTOR

SYMMETRIC_PLANE

Purpose: This keyword causes LS-DYNA to run a blank-size development calculation instead of a finite element calculation. The input for this feature consists of (1) the result of a completed metal forming simulation, (2) the corresponding initial blank, and (3) the desired result from the simulation in the form of a boundary curve or full mesh. From these inputs, the *INTERFACE_BLANKSIZE method adjusts the initial blank so that the resulting formed piece more closely matches the target. Iterating may systematically improve the blank's starting geometry. This feature is available for both SMP and MPP but requires a double precision executable. A GUI for using this available in LS-PrePost as of version 4.2 under *APPLICATION → Metal Forming → Blank Size/Trim line*.

> **NOTE:** When this card is present, LS-DYNA *does not* proceed to the finite element simulation.

This keyword requires one or all three keyword options in the input deck, each corresponding to a different kind of forming operation:

1. **DEVELOPMENT.** This option takes as its target either a full mesh or a minimum boundary. It adjusts the blank so that the product more closely approximates the target. The computed blank-boundary is written to a file called trimcurves.ibo, which contains a *DEFINE_CURVE_TRIM_3D keyword.

2. **INITIAL_TRIM.** This option adjusts the blank so that trimming and mesh-refinement are mapped back onto the initial blank.

3. **INITIAL_ADAPTIVE.** This option reads in (1) the input mesh for a flanging simulation as well as (2) the adapted mesh calculated during flanging simulation. It maps the refinement back to the initial blank.

Card Set for *INTERFACE_BLANKSIZE_DEVELOPMENT.

Development Parameter Card.

Card 1	1	2	3	4	5	6	7	8
Variable	IOPTION		IADAPT	MAXSIZE	REF	SPACE	MAXGAP	ORIENT
Type	I		I	F	I	F	F	I
Default	-2		1	30.0	0	2.0	30.0	none

Target Card. See "Target curves (target.xyz)" in Figure 30-1.

Card 2	1	2	3	4	5	6	7	8
Variable	FILENAME1							
Type	A80							

Final blank Card. See "Final blank (final.k)" in Figure 30-1.

Card 3	1	2	3	4	5	6	7	8
Variable	FILENAME2							
Type	A80							

Initial blank Card. See "Initial blank (initial.k)" in Figure 30-1.

Card 4	1	2	3	4	5	6	7	8
Variable	FILENAME3							
Type	A80							

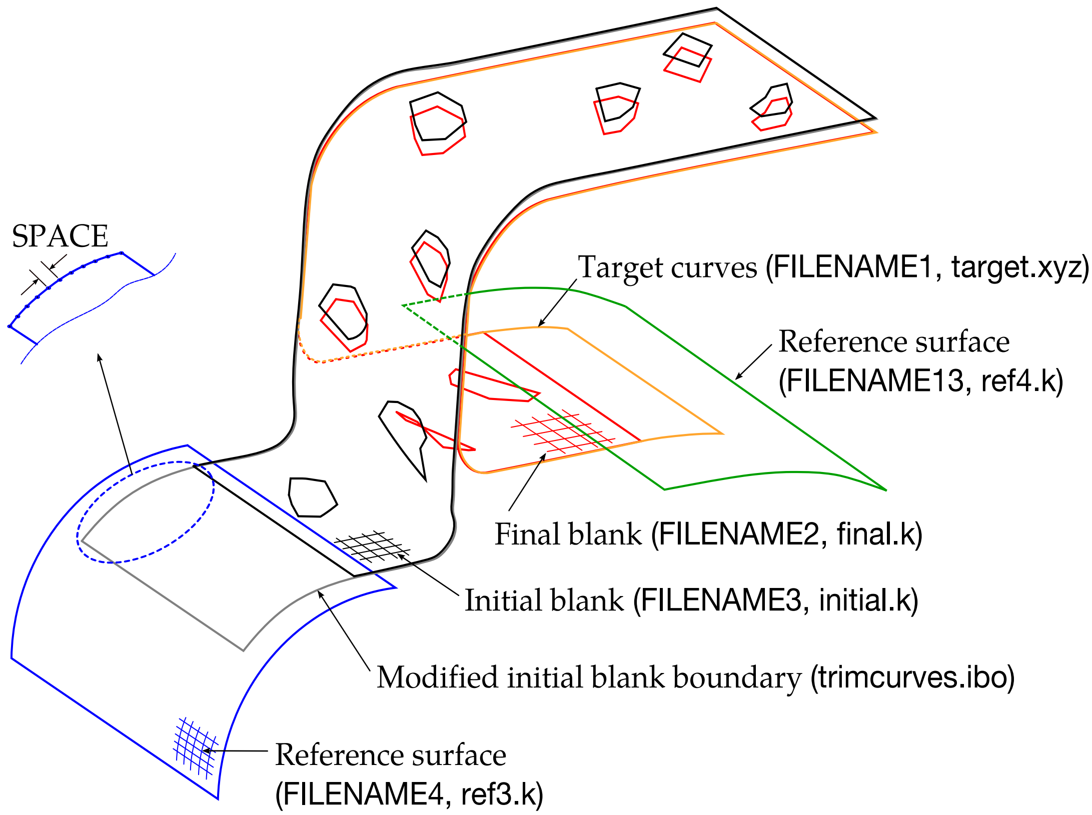


Figure 30-1. Trim curve development using a reference surface. See Example II.

Reference Surface Card. See "Reference surface (ref3.k)" in Figure 30-1and Example II.

Card 5	1	2	3	4	5	6	7	8
Variable	FILENAME4							
Type	A80							

Reference Surface Card. See "Reference surface (ref4.k)" in Figure 30-1 and Example II.

Card 6	1	2	3	4	5	6	7	8
Variable	FILENAME13							
Type	A80							

VARIABLE	DESCRIPTION
IOPTION	Target curve definition input type:

VARIABLE	DESCRIPTION
	EQ.1: (Entire) blank mesh in keyword format. EQ.2: Consecutive position coordinates of blank boundary loop curve in XYZ format, defined by *DEFINE_TARGET_BOUNDARY. The blank mesh's normal vector and the closed boundary curve are consistently oriented according to the right-hand rule, see Figure 30-2. Note the target boundary curves must have enough points for a successful prediction of the initial blank size. EQ.-2: Consecutive position coordinates of blank boundary loop curve in XYZ format, defined by *DEFINE_TARGET_BOUNDARY. The blank mesh's normal vector and the closed boundary curve are consistently oriented according to the left-hand rule, see Figure 30-2. Note the target boundary curves must have enough points for a successful prediction of the initial blank size. **NOTE:** Starting in August 2015, the curve direction is desensitized, meaning *both* IOPTION = 2 and IOPTION = -2 give the same results. In LS-PrePost 4.0,use menu option *GeoTol → ID Measure* to show the flow direction of a boundary curve. To reverse the curve direction, use *Curve → Rever(se)*.
IADAPT	Adaptive mesh control flag. If IADAPT = 1, the number of elements in the meshes initial (FILENAME3) and simulated blank (FILENAME2) can be different, so it is not necessary to use the sheet blank from the file adapt.msh (created by setting IOFLAG = 1 in *CONTROL_ADAPTIVE) for the initial blank mesh.
MAXSIZE	The maximum change in initial blank size in each iteration. It is used to limit the blank size change in each iteration during mapping to avoid convergence problems when the initial blank is curved.
REF	Flag to indicate trim curve projection to a reference surface (mesh); see Figure 30-1:

VARIABLE	DESCRIPTION
	EQ.0: No projection. EQ.1: The trim curves will be projected to the reference surface. In addition, the mesh file for the reference surface is given in FILENAME4.
SPACE	Point spacing distance on the reference surface for the projected curve; see Figure 30-1. If the gap between two neighboring points in the modified trimming curve is larger than this value, extra nodes will be added in the final blank and projected to the reference surface. A smaller value should be used for large reference surface curvature.
MAXGAP	When REF is set to 1, the nodes from the final blank will be projected to the reference surface. However, if the distance between the nodes and the surface is larger than MAXGAP, the nodes will not be projected.
ORIENT	A flag to control iterative optimization efficiency or to include a reference surface file for the final formed blank. EQ.1: Activates a new algorithm to potentially reduce the number of iterations during the iterative blank size optimization loop. This activation should be used in conjunction with the keyword option SCALE_FACTOR (0.75~0.9). EQ.2: To include a reference surface FILENAME13 (in keyword mesh format) for the simulated (formed) blank. This is useful when the formed blank is smaller than intended and extended surfaces are not flat. See Example II and Figure 30-1.
FILENAME1	Target input file name. When a blank mesh is used (IOPTION = 1), the keyword file must contain *NODE and *ELEMENT_SHELL keywords. When using the blank boundary curve for target (|IOPTION| = 2), the file must consist of *DEFINE_TARGET_BOUNDARY. See the Target Boundaries (and IGES) in the Remarks section.
FILENAME2	Simulated (formed or flanged) sheet blank mesh in keyword format. This mesh can be obtained from the final state of any downstream process simulation.

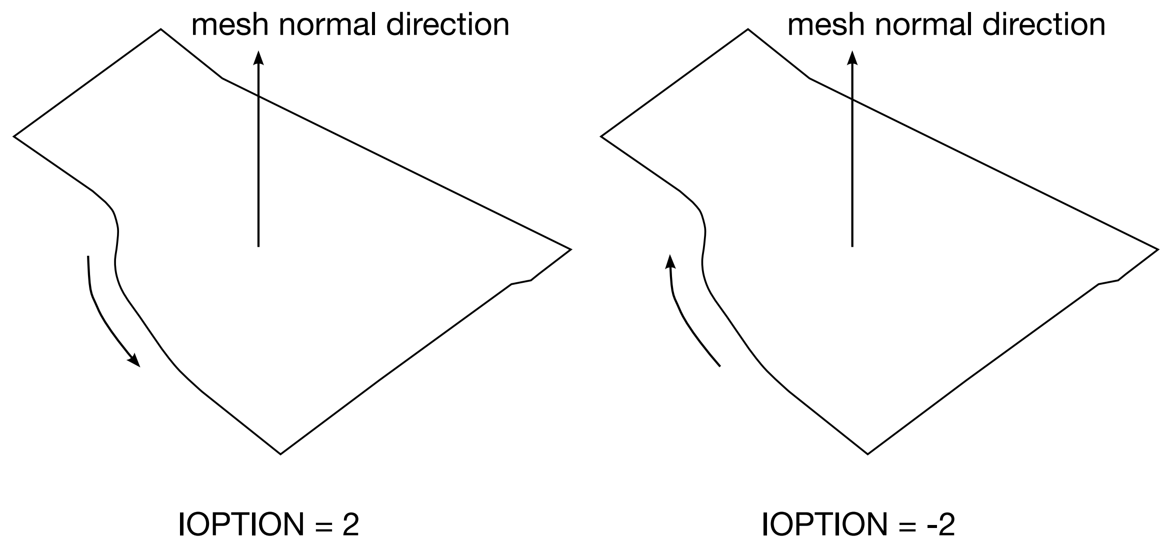


Figure 30-2. Differences between IOPTION 2 and -2.

VARIABLE	DESCRIPTION
FILENAME3	Initial sheet blank mesh in keyword format. This can be the first state mesh from any process simulation prior to FILENAME2 simulation. Set IADAPT = 1 if adaptive refinement is used in any simulation.
FILENAME4	Reference surface onto which adjustments to the blank's trim curves in its initial state are projected (ref3.k in Figure 30-1). This surface is typically a curved extension of the initial blank and must be defined as a mesh in keyword format. This file name must be defined when REF is set to 1. Also see extended reference surface for the final state (formed state), FILENAME13.
FILENAME13	Reference surface onto which adjustments to the blank's trim curves in its final state are projected (ref4.k in Figure 30-1). This surface is typically a curved extension of the formed blank and must be defined as mesh in keyword format. This file name must be defined when ORIENT is set to 2.

Card Set for *INTERFACE_BLANKSIZE_INITIAL_TRIM.

Initial Flat blank. FILENAME5 should specify the mesh of a blank that has been refined during a subsequent forming operation. FILENAME5 is usually set to the adapt.msh output. See "OP10 Initial Adapted blank mesh" in Figure 30-3.

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME5							
Type	A80							

Formed blank. FILENAME6 specifies a dynain file from a forming simulation. See "OP10 Final blank" in Figure 30-3.

Card 2	1	2	3	4	5	6	7	8
Variable	FILENAME6							
Type	A80							

Trimmed Formed blank. A dynain file from a trimming simulation that started with the state given in FILENAME6. See "OP20 Final blank" in Figure 30-3.

Card 3	1	2	3	4	5	6	7	8
Variable	FILENAME7							
Type	A80							

Trimmed Flat blank (output). This field specifies the name for the file to which the trimmed flat blank is written. See "OP20 Flat New" in Figure 30-3.

Card 4	1	2	3	4	5	6	7	8
Variable	FILENAME8							
Type	A80							

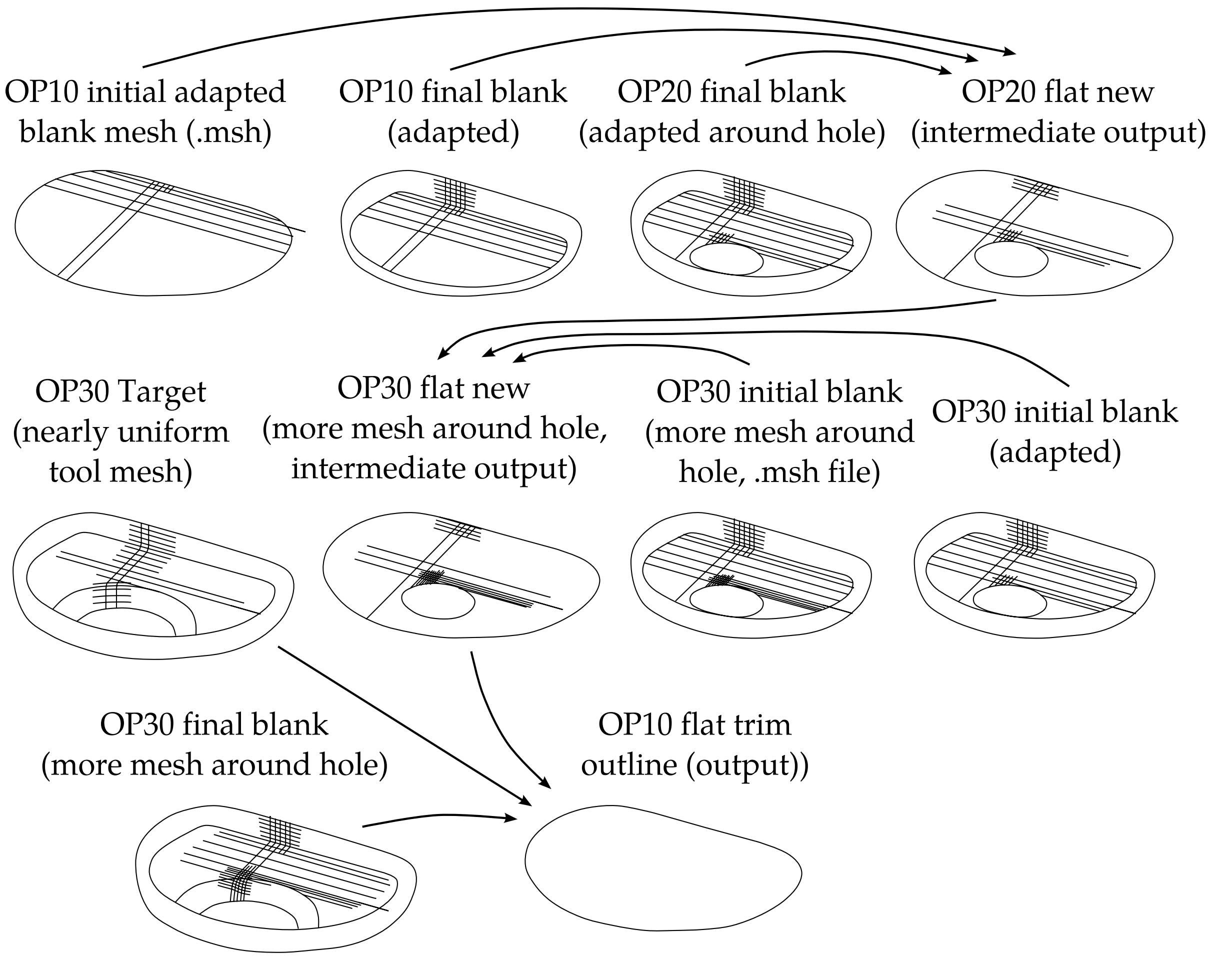


Figure 30-3. Inputs and outputs (*Courtesy of Metal Forming Analysis Corp.*). For exposition of labels OP10, OP20, and OP30 see Figure 30-4.

VARIABLE	DESCRIPTION
FILENAME5	Initial blank in its flat configuration *with* adapted mesh in keyword format. FILENAME5 usually points to an adapt.msh file. For example see OP10 in Figures 30-4, 30-22 and 30-3, in which FILENAME5 is the adapt.msh from a draw-forming calculation.
FILENAME6	Final formed blank in keyword format. This is usually the dynain file corresponding to the adapt.msh file mentioned above for FILENAME5; see Figures 30-4, 30-22 and 30-3.
FILENAME7	A trimmed blank in keyword format. This file should be derived from FILENAME6 as the dynain file from a trimming simulation. See OP20 in Figures 30-4, 30-22 and 30-3.
FILENAME8	This field specifies the name for the file in which the trimmed flat blank is to be written. See OP20 in Figures 30-4, 30-22 and 30-3.

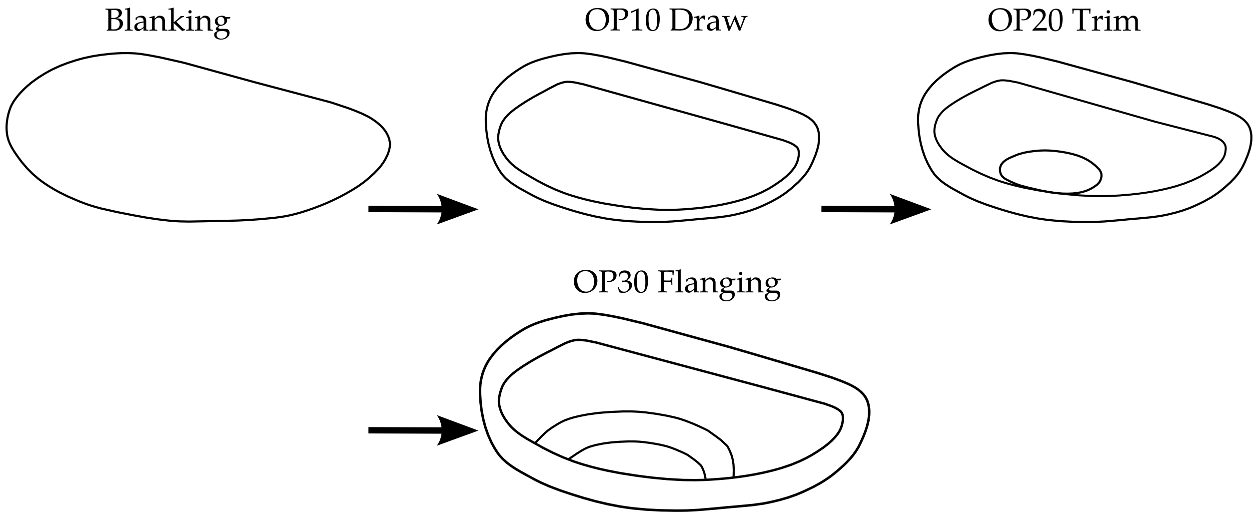


Figure 30-4. A stamping process consists of draw, trim and flanging (*Courtesy of Metal Forming Analysis Corporation*). The labels OP10, OP20, and OP30 are used *extensively* in the ensuing discussion.

Card Set for *INTERFACE_BLANKSIZE_INITIAL_ADAPTIVE.

Flat blank. FILENAME9 specifies the mesh of a blank in a flat configuration serving as the basis for a two-step metal forming process. The second-stage simulation produces an adapt.msh file, from which the refinements are to be mapped onto the flat-blank of FILENAME9. See, for example, "OP20 Flat New" in Figure 30-3 where FILENAME9 points to result, FILENAME8, of an INITIAL_TRIM calculation.

Card 1	1	2	3	4	5	6	7	8
Variable	FILENAME9							
Type	A80							

Initial blank. FILENAME10 is the result from the first stage of a two-step process. See, for example, "OP30 Initial blank" in Figure 30-3 where FILENAME10 has been formed and trimmed and refined along the way.

Card 2	1	2	3	4	5	6	7	8
Variable	FILENAME10							
Type	A80							

Adapted Initial blank. FILENAME11 contains a refined version of the mesh in FILENAME10. It is expect to name the adapt.msh file from some operation performed on the mesh of FILENAME10. See, for example, "OP30 Initial blank (with more mesh)" in Figure 30-3, where the adapt.msh file comes from a flanging simulation.

Card 3	1	2	3	4	5	6	7	8
Variable	FILENAME11							
Type	A80							

Refined Flat blank (output). This field specifies the name of the file to which the refined flat blank is written. The blank's mesh is refined to *exactly* match the forming process. See, for example, "OP30 Flat New" in Figure 30-3.

Card 4	1	2	3	4	5	6	7	8
Variable	FILENAME12							
Type	A80							

VARIABLE	DESCRIPTION
FILENAME9	FILENAME9 should point to the blank defined by FILENAME8, as in Figures 30-4, 30-22 and 30-3.
FILENAME10	Initial-stage result file name. This may be extracted from d3plot files using LS-PrePost or it may be generated by LS-DYNA as a dynain file. See, for example, "OP30 Initial Blank" in Figure 30-3 where FILENAME10 has been formed, trimmed and refined along the way. To obtain from d3plot file the necessary mesh in keyword format using LS-PrePost4.0 select *POST* → *OUTPUT* → *Dynain ASCII* and check the box for *Exclude strain and stress*.
FILENAME11	FILENAME11 contains a refined version of the mesh in FILENAME10. FILENAME11 can be obtained from adapt.msh file from the same operation performed on the mesh of FILENAME10. See, for example, "OP30 Initial Blank (with more mesh)" in Figure 30-3, where the adapt.msh file comes from a flanging calculation.
FILENAME12	This field specifies the name of the file to which the refined flat blank is written. The blank's mesh is refined to *exactly* match the forming process. See, for example, "OP30 Flat New" in Figure 30-3.

Card Set for *INTERFACE_BLANKSIZE_SCALE_FACTOR.

Scale Factor Card. Define one card for each curve. 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	IDCRV	SF	OFFX	OFFY	OFFZ			
Type	I	F	F	F	F			
Default	1	0.0	0.0	0.0	0.0			

VARIABLE	DESCRIPTION
IDCRV	Curve ID in the order of appearance as in FILENAME1 in the target card, as defined by *DEFINE_TARGET_BOUNDARY.
SF	Scale factor for the IDCRV defined above. It defines a fraction of the changes required for the predicted initial blank shape. For example, if SF is set to 0.0, the corresponding IDCRV will be excluded from the calculation (although the original initial curve still will be output); on the other hand, if SF is set to 1.0, full change will be applied to obtain the modified initial blank that reflects the forming process. A SF of 0.5 will apply 50% of the changes required to map the initial blank. This feature is especially important for inner holes that are small and hole boundary expansions are large, so the predicted initial hole can avoid "crisscross" situation. An example is provided in Scale Factor and Symmetric Plane.
OFFX, OFFY, OFFZ	Translational move of the target curve. This is useful when multiple target curves (e.g. holes) and formed curves are far away from each other. Input values of OFFX, OFFY and OFFZ helps establish one-to-one correspondence between each target curve and formed curve. OFFX.EQ.-10000.0: offset values are automatically calculated.

Card Set for *INTERFACE_BLANKSIZE_SYMMETRIC_PLANE.

Symmetric Plane Card.

Card 1	1	2	3	4	5	6	7	8
Variable	X0	Y0	Z0	V1	V2	V3		
Type	F	F	F	F	F	F		
Default	0.0	0.0	0.0	1.0	0.0	0.0		

VARIABLE	DESCRIPTION
X0, Y0, Z0	x, y, z coordinates of a point on the symmetric plane. See example in Scale Factor and Symmetric Plane.
V1, V2, V3	Vector components of the symmetric plane's normal. See example in Scale Factor and Symmetric Plane.

Remarks and Examples

Inverse Methods for Optimizing Blank Size and Trim Lines:

Finding the minimal practicable blank size and developing an optimal set of trim lines is an integral part of the die engineering process. This card, *INTERFACE_BLANKSIZE, is one of several features that have been developed to solve the inverse problem: that is, to calculate an initial blank or blank boundary that will yield a desired product based mostly on the target geometry of that final product.

1. **The One-Step Method.** The *CONTROL_FORMING_ONESTEP card is suitable for early blank-size estimates. It invokes the total-strain theory of plasticity, thereby bypassing the, as of yet, undetermined specific details of the forming process.

2. **Unflanging.** Once the product design and process plan are complete, die development begins with addendum and binder creation, followed by secondary tooling development. In this stage, *CONTROL_FORMING_UNFLANGING can be used to develop trim lines for the secondary tooling; final (or intermediate) desired flange shapes are unfolded onto the addendum or binder to obtain the corresponding flange shapes in its initial shape. It also implements failure criteria to arrive at suggested final flange curves, with strains and thickness output on the unfolded flanges.

3. **Interface Blanksize.** This card, *INTERFACE_BLANKSIZE, can be used to *accurately* determine the optimal initial blank. To do so it requires an initial configuration, the corresponding simulated configuration, and a desired target configuration. This method takes into account the entire metal-forming process.

   a) One application of this keyword is to map trim curves between dies to calculate the trim curves needed for all trim dies. An example of the application can be found in Figures 30-18 through 30-21.

	Compulational Cost	Accuracy	Information Required	Physical Process
Blanksize	Full simulation	Exact	Full Simulation	Any
Unflanging	Fast	Approximate	Process Geometry	Inverse Flanging
Onestep	Fast	Approximate	None (*Path independent*)	Any

Table 30-1. Comparison of inverse methods.

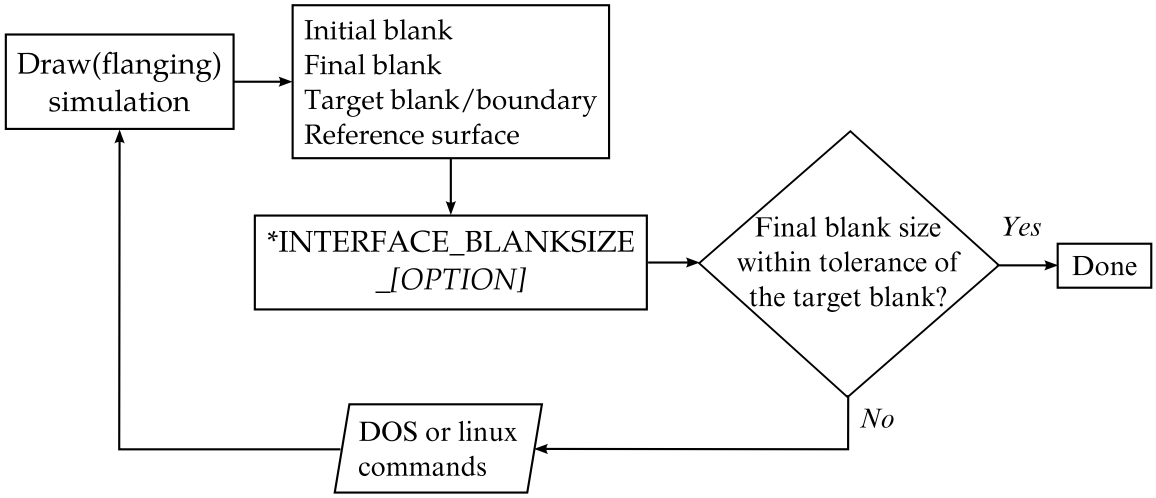


Figure 30-5. An iterative blank size development process.

b) The keyword can also be used to determine the precise minimal initial blank needed for a draw panel whose blank edge must be at specified distances from the edge of the draw beads.

Iterative Workflow:

The "interface blanksize" command produces an adjusted initial blank. It is not to be expected that the adjustment will be exact. However, this initial blank shape can be run through a second simulation to see if the final shape is close enough to the target blank. If it is not close enough, then the results from the second simulation can be used to repeat the process. Iterations can proceed until the final shape is within the range of the target shape. This iterative blank size development process is schematically presented in Figure 30-5 and exemplified in Figures 30-8 through 30-17.

Target Boundaries (and IGES):

When IOPTION = 2, or -2, a file with the keyword *DEFINE_TARGET_BOUNDARY must be present. This keyword can now be created from an IGES file using LS-PrePost 4.1 (or 4.2). To convert from IGES curves to keyword *DEFINE_TARGET_BOUNDARY in LS-PrePost 4.1, use menu option *Curve → Convert→ Method (To Keyword) → Select *DEFINE_TARGET_BOUNDARY*; pick the curves then hit *To Key*; write out the keyword file using *File → Save as → Save Keyword As*; and select *Output Version* as *V971_R7*. In LS-PrePost 4.2, select *Application → Metal Forming → Blanksize Trimline* to import the target curve directly in IGES format, the initial and final blank mesh and then write out a complete LS-DYNA input deck.

In addition, the target curves should be projected onto the final blank mesh if they do not exactly lie on the mesh surface. This can be done with LS-PrePost 4.1 using the menu option *GeoTol* → *Project* → *Project*, select *Closest Projection*, select *Project to Elements*, then define the destination mesh and source curves, and hit *Apply*.

Computed Initial Blank Boundaries (and IGES):

Computed boundary curves are written with *DEFINE_CURVE_TRIM_3D keyword into a file called trimcurves.ibo. The format of this file follows the keyword's specification. LS-PrePost4.0 can convert the computed curve to IGES. See Figure 30-10. After hitting *Apply*, the curves will show up in the graphics window, and *File* → *Save as* → *Save Geom as* can be used to write the curves out in IGES format. In the LS-PrePost 4.2 GUI, under *Results*, trimcurves.ibo can be directly imported into the graphics window for viewing and to save in either STEP or IGES format.

To convert IGES to the *DEFINE_CURVE_TRIM_3D keyword format, import the IGES file into LS-PrePost4.0 and follow the procedures shown in Figure 30-11. After finishing step 2, "*curves have been converted to keyword format*" will be reported in the command prompt. Then use *File* → *Save* → *Save keyword* to write out the keyword file.

Support for Multi-Stage Processes with the Development Option:

Original Implementation.

Prior to Revision 88708 the development option required that the final blank (FILENAME2) differ from the initial blank (FILENAME3) by no more than a deformation and mesh refinement. In practice, this means that the two meshes must come from the same process simulation. For example, in a draw, trim and flanging process, the trimmed panel mesh is used for flanging simulation. Therefore, with the original implementation, LS-DYNA required that the initial blank state (FILENAME3) be trimmed when the final blank state (FILENAME2) is flanged on trimmed panel. Failure to observe this limitation may result in error termination.

Enhanced Implementation.

A more recent improvement to the blank size development (Revision 88708) removes the requirement that initial (FILENAME3) and final (FILENAME2) blanks must be from the same process simulations. The initial and final blank states *may* differ by a trimming process. This allows trimming and other process such as flanging to occur between the initial and final blanks, without the need of invoking the INITIAL_TRIM and INITIAL_-ADAPTIVE options. For example, the initial blank can be the blank mesh from "Blanking" in Figure 30-4, and the final blank can be the blank mesh from "Flanging," which is also in Figure 30-4.

Scale Factor and Symmetric Plane:

An example of using various scale factors ranging from 0.0 to 1.0 on a model involving an initial hole shape is shown in Figure 30-25. The target curve is given in targetline.k. All nodes along the symmetric plane are constrained by the SYMMETRIC_PLANE option. The symmetric plane is defined going through point coordinates $(-76, 2.63844, 0.38)$ with plane normal vector of $(1.0, 0.0, 0.0)$. A complete input is provided below:

```
*KEYWORD
*INTERFACE_BLANKSIZE_DEVELOPMENT
$  IOPTION                 IADAPT
        -2                      1
$ target boundary curves
targetline.k
$ final formed mesh
final.k
$ initial mesh
initial.k
*INTERFACE_BLANKSIZE_SCALE_FACTOR
$    IDCRV        SF
         1       0.2
*INTERFACE_BLANKSIZE_SYMMETRIC_PLANE
$       X0        Y0        Z0        V1        V2        V3
       -76   2.63844      0.38       1.0       0.0       0.0
*END
```

If targetline.k consists of multiple curves, the following format can be used:

```
*INTERFACE_BLANKSIZE_SCALE_FACTOR
$    IDCRV        SF
         1       0.2
         2       0.8
         3       1.0
         ⋮         ⋮
```

Example I: Simple Example of the Development Option

Given the initial and final blank configuration and a target, this option calculates a new initial blank outline, corresponding to the target final blank boundary. In this example note that IADAPT = 1, meaning initial and final blank meshes may differ by an adaptively operation. The input and output files are detailed below, and output results are shown in Figure 30-6.

```
*KEYWORD
*INTERFACE_BLANKSIZE_DEVELOPMENT
$  IOPTION                 IADAPT
         2                      1
$  input file for target mesh, or target position coordinates
targetpoints.k
$  input file for formed mesh
final.k
$  input file for initial blank mesh
initial.k
*END
```

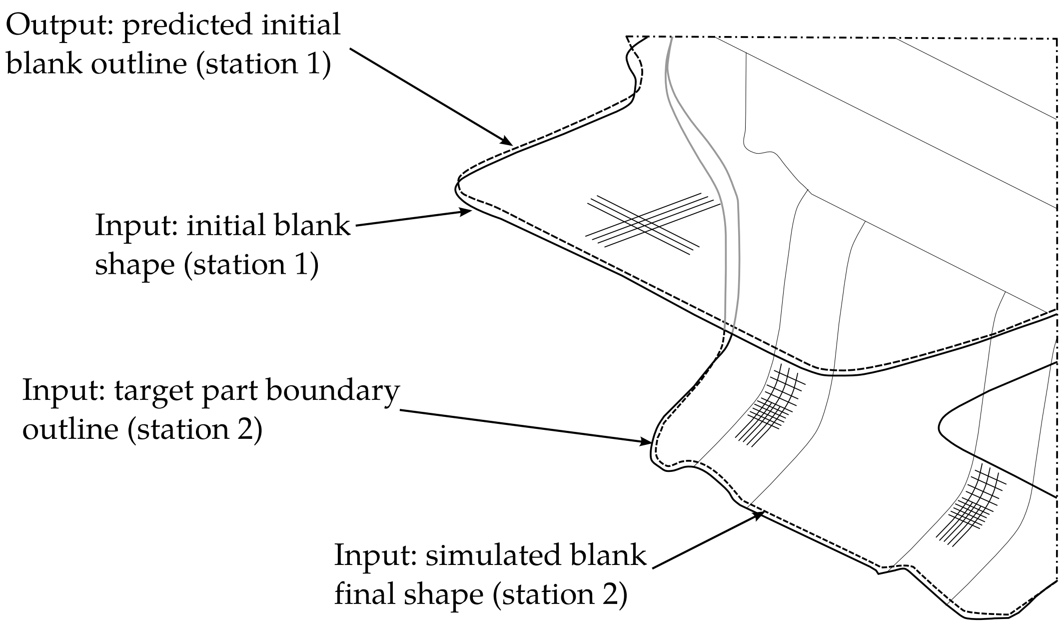


Figure 30-6. Blank size development in a progressive die with IADAPT = 1 in Example I.

The file, targetpoints.k partially shown below, was generated from IGES using *LS-PrePost 4.1*.

```
*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
```

The output is the modified initial blank outline in the file trimcurves.ibo.

Example II: The Reference Surface feature for the Development Option

For an initial blank that is not flat, the fields REF and FILENAME4 can be used to define a surface onto which changes in the boundary are needed. This is important when the adjusted boundary is not a simple tangential extension of the initial blank.

In a keyword example below, REF is set to "1" and the reference file for the extended initial shape is given as ref3.k. The maximum change between the initial and final blank size is set to be 20.0 mm per iteration. Point spacing distance (SPACE) of the calculated trim curve on the reference surface is set at 2.0 mm. Note that the inner holes and outer boundary curves are defined in target.xyz. The holes do not necessarily need to exist in the initial or final blank mesh. Also, since ORIENT is set to 2, a reference surface mesh

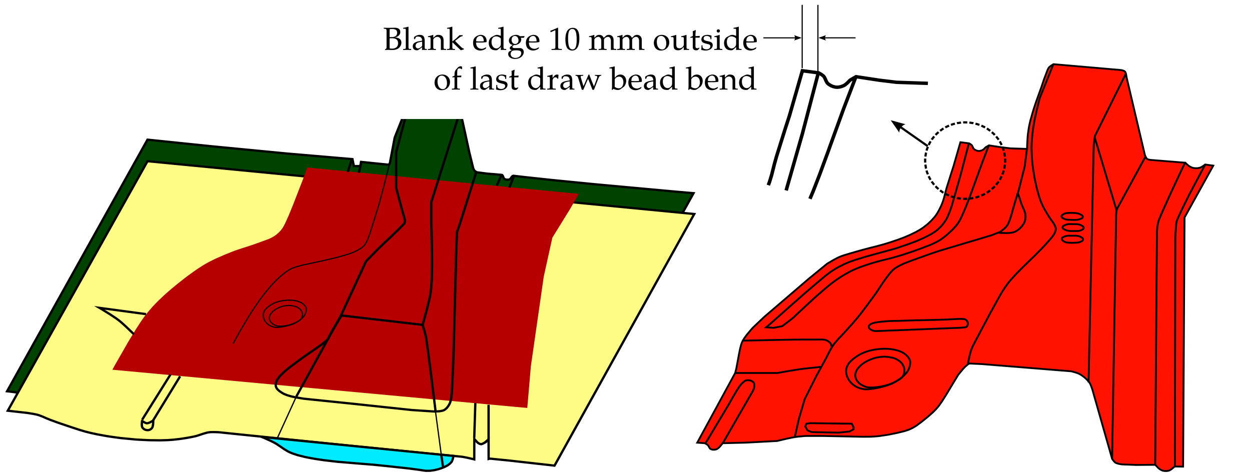


Figure 30-7. NUMISHEET 2005 cross member in Exmaple III.

file (ref4.k) is provided for the final (formed) state. The input details and output results are shown in Figure 30-1.

```
*KEYWORD
*INTERFACE_BLANKSIZE_DEVELOPMENT
$        1         2         3         4         5         6         7         8
$  IOPTION                IADAPT   MAXSIZE       REF     SPACE              ORIENT
        -2                     1    20.000         1       2.0                   2
$  input file for target mesh:
target.xyz
$  input file for formed mesh:
final.k
$  input file for initial blank mesh:
initial.k
$  reference file for extended initial shape:
ref3.k
$  reference file for extended final shape:
Ref4.k
*END
```

Example III: Development Feature Applied to a Draw Die with Physical Bead

In this example, which was created from NUMISHEET 2005, the DEVELOPMENT option has been used to design a blank such that, when formed, the edge is a specified distance outside of the last bend of a draw bead. In Figure 30-7, the tooling and blank set up is shown on the left. The right side of the figure shows the target blank, whose left and right edges everywhere are made 10 mm outside of the last bending radius of the draw beads. This setup is prototypical of one method to ensure a very stable and high-quality stamping process.

The first step towards setting up this analysis was to use *CONTROL_FORMING_ONESTEP to unfold the target blank and thereby obtain an initial guess of a flat blank, as shown to the left in Figure 30-12. The flat blank is then formed as one would usually do

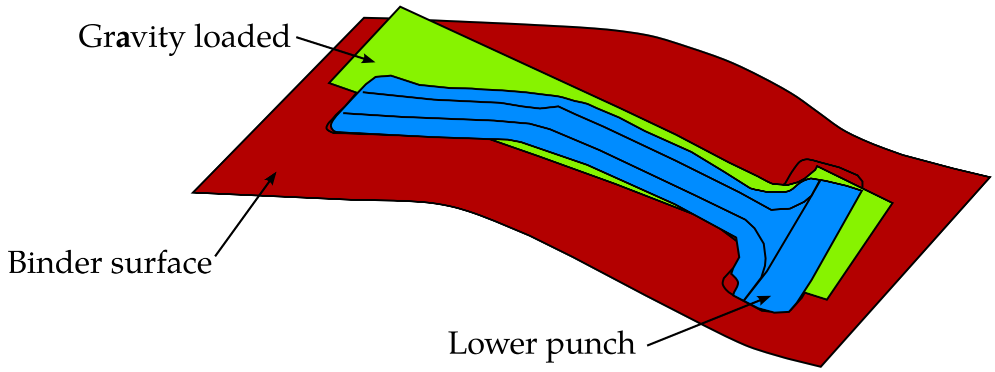


Figure 30-8. NUMISHEET 2008 B-pillar; Gravity loaded blank.

in a regular forming simulation as shown on the right side of the figure. The formed blank (Iteration 0) turns out to be larger than the target.

Next, the DEVELOPMENT is applied to generate a new and better initial blank that will lead to the target blank. The flat blank is used as input for the "initial blank mesh", the formed blank is used as input for the "simulated mesh", and the target blank mesh, or boundary points, is used to define the target.

In Figure 30-13 (left) the improved initial blank, called *the first compensated blank,* is superimposed onto the original one-step unfolded result. The one-step unfolded result is somewhat larger than the developed blank. When formed the improved blank (Iteration 1) nearly overlaps the target blank, shown in Figure 30-13 (right). If the final formed blank still deviates from the target, another iteration would ensue, until satisfactory results are obtained.

Example IV: Iterating with the Development Option

Because the NUMISHEET 2008 B-pillar model involves neither trimming nor flanging, it exmplifies the DEVELOPMENT option in its most direct use case. This model, illustrated in Figure 30-8, simulates a draw-die's action on a gravity-stressed flat blank. The B-pillar undergoes a stamping process including gravity, binder closing, and being drawn. In this example the DEVELOPMENT option is used to calculate the geometry for an initial blank that will exactly satisfy the design specification for a final formed panel. To highlight the efficiency of this feature, we start with an initial blank whose formed product deviates from the specification by a wide margin and then iterate using the DEVELOPMENT feature.

The target and the optimal initial blank are shown in Figure 30-14. The intial guess intentionally deviates from the optimal initial blank as shown in the left panel of Figure 30-15; the formed blank is compared with the target in the right panel.

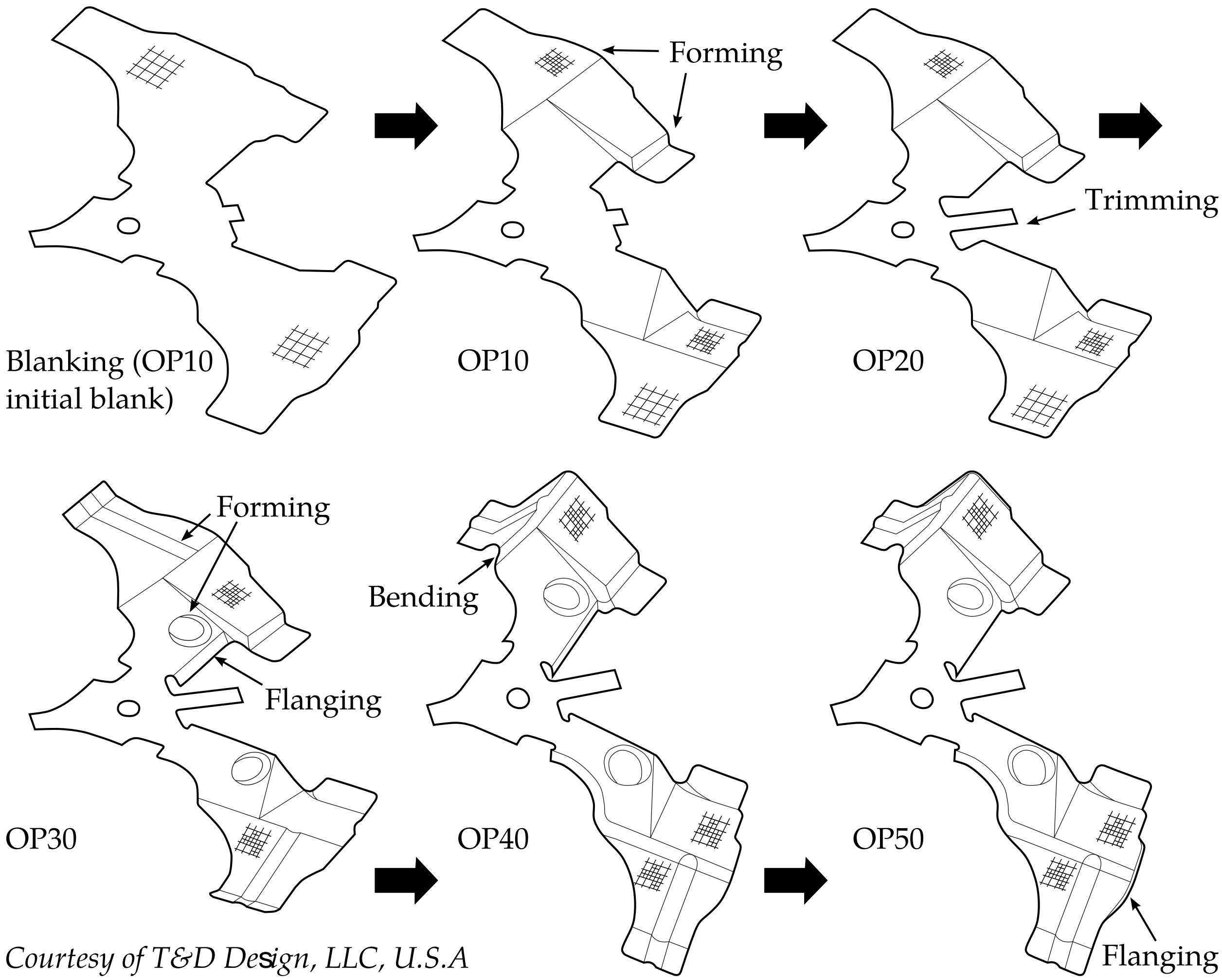


Figure 30-9. Enhanced DEVELOPMENT feature on a progressive die. Even though trimming occurs at OP20, the algorithm requires as input only OP10, OP50, and a target geometry.

In the first iteration a new initial blank is computed, and illustrated in the left panel of Figure 30-16, bearing the label *1st compensated blank*. The simulation is repeated using the first compensated blank, and in the right panel the result is compared to the target. The formed blank is narrower in the notched areas as compared with the target.

The second iteration is shown in the left panel of Figure 30-17 bearing the label, *2nd compensated blank*. Again, the simulation is repeated, but this time using the second compenated blank, and the result is compared to the target in the right panel of Figure 30-17. The resulting product is a good match to the target.

Because the initial blank intentionally deviated from its ideal shape by a large margin, this example requires two iterations to converge. Generally this processed is bootstrapped with the *CONTROL_FORMING_ONESTEP card, which calculates an initial guess by approximately unfolding the target shape.

Example V: Development Feature Applied to Flanging Process

In this example, which is schematically illustrated in Figure 30-18, the NUMISHEET 2002 fender outer is flanged along the hood line. The development feature adjusts the initial blank's boundary so that the formed piece matches the specified target flanged shape as shown in Figure 30-19. For demonstration purposes, the trimmed blank shape intentionally deviates from the optimal configuration by a large amount. This error is indicated in Figure 30-20 by the label *initial guess trim curves*. In Figure 30-20 the flanged product is shown to deviate substantially from the flanged target along the boundary. As shown in Figure 30-21 after one iteration the correct initial blank boundary is obtained.

Alternatively, *CONTROL_FORMING_UNFLANGING can be used to unfold the flanged target onto the addendum to obtain the initial blank size, or a starting guess for this process.

Example VI: Enhanced-Development Feature Applied to Progressive Die Process

In the example of Figure 30-9, *courtesy of T&D Design, LLC, U.S.A*, the blank-shape for a five stage progressive die process is calculated. Because this process involves a trimming step, the development capability prior to Revission 88708 does not support this example.

In this example, an initial blank at OP10, undergoes trimming, reforming, bending and flanging to arrive at the blank in OP50. In Figure 30-23 the computed product is compared with the specified target. A blank size development calculation produces the modified OP10 initial blank outline. The updated blank is used in a verification simulation. As seen in Figure 30-24, the blank size development feature produces a good result.

Trim lines are not optimized by the development feature, so trimming should only occur along the boundary of the target blank. The modified OP10 blank requires some refitting in the trimmed area.

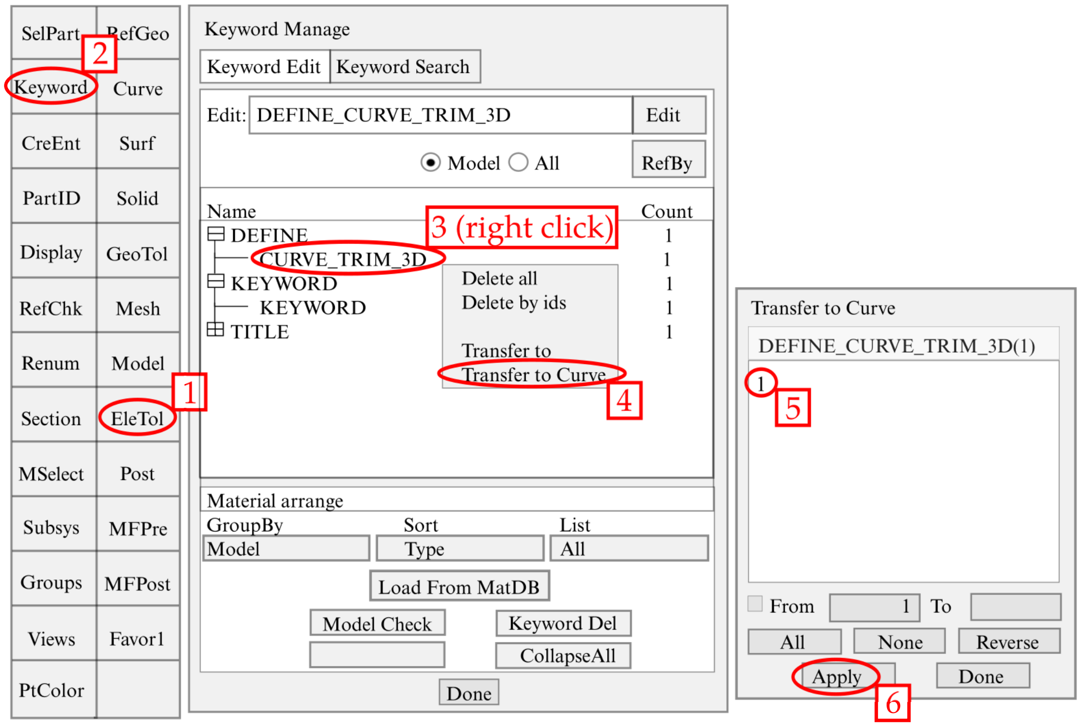


Figure 30-10. Converting trimcurves.ibo to IGES format in LSPP4.0.

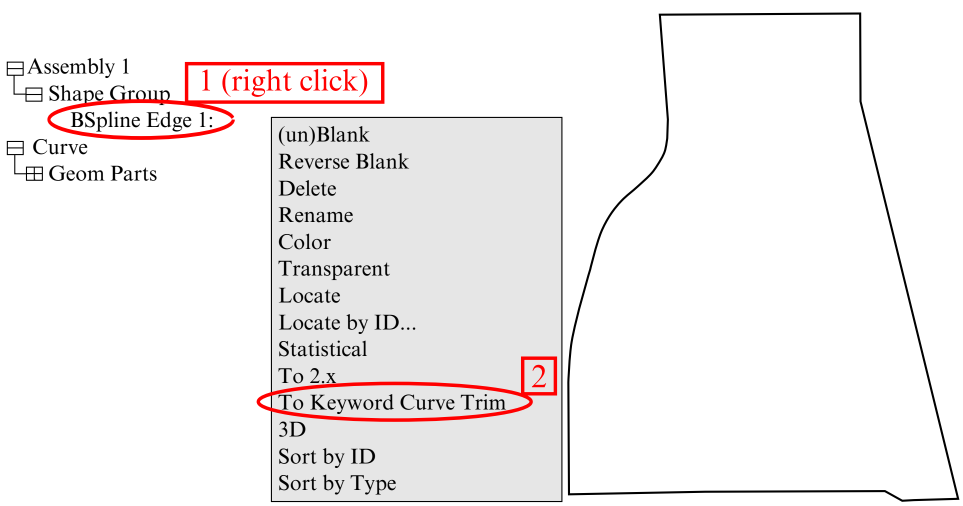


Figure 30-11. Converting IGES file to *DEFINE_CURVE_TRIM_3D.

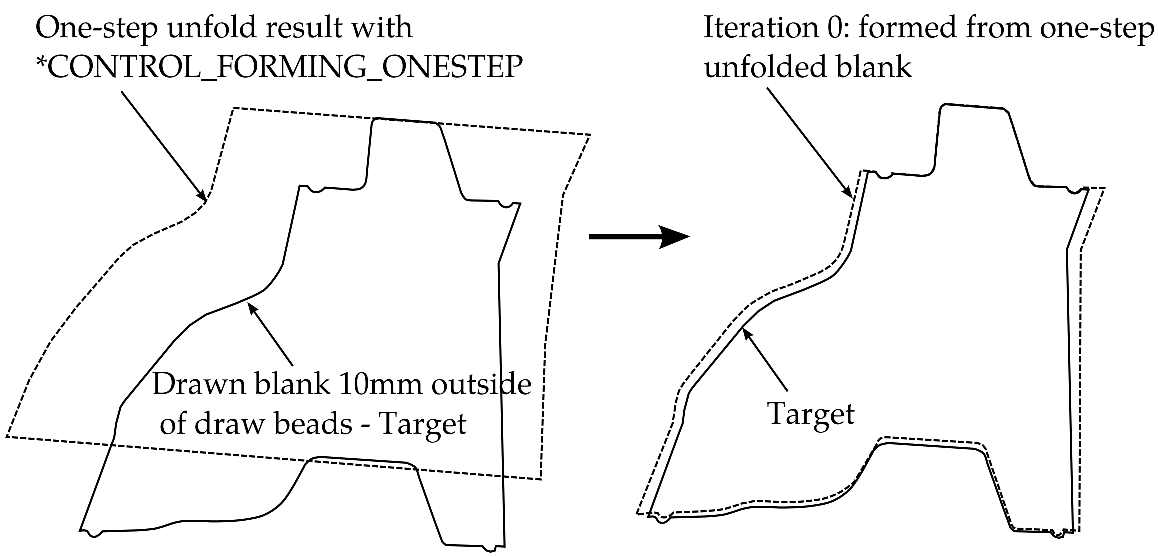


Figure 30-12. Initial blank calculation and baseline formed blank.

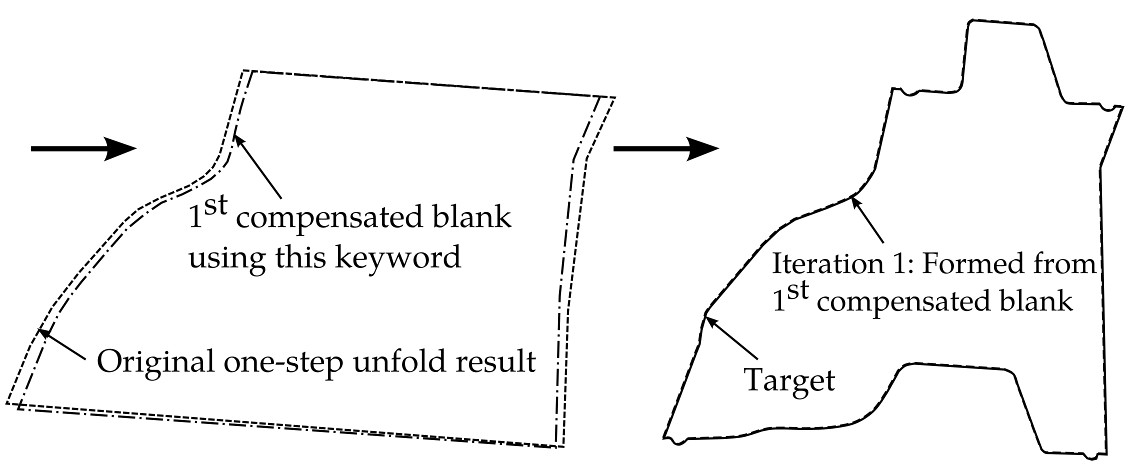


Figure 30-13. The first compensated blank and the final confirmation run.

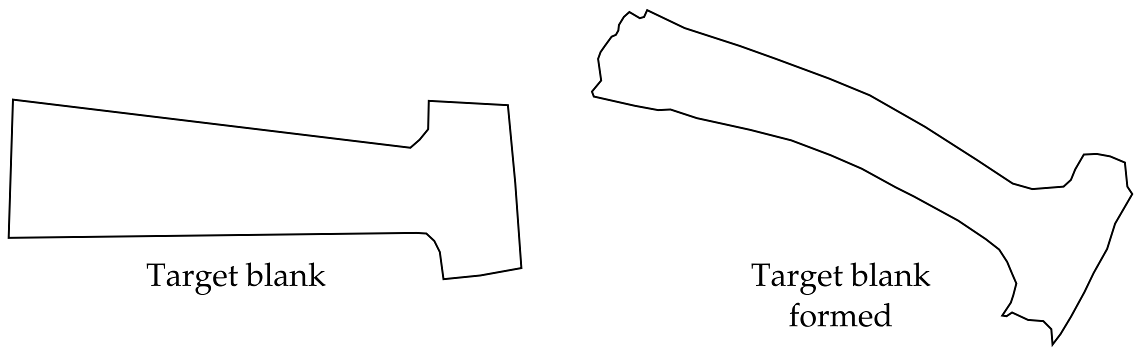


Figure 30-14. Assumed target blanks.

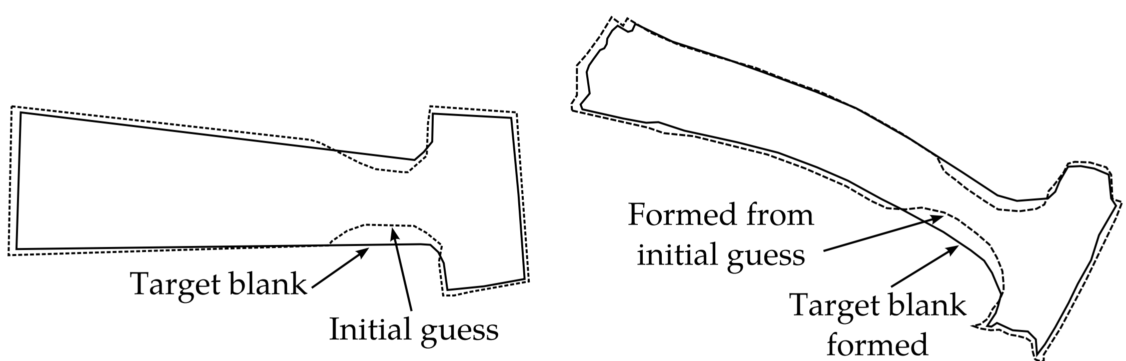


Figure 30-15. Iteration 0 results comparison with target.

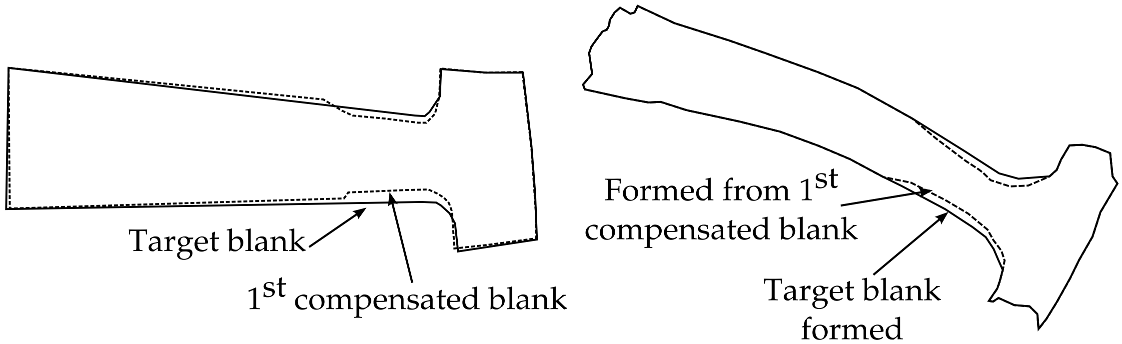


Figure 30-16. Iteration 1 results.

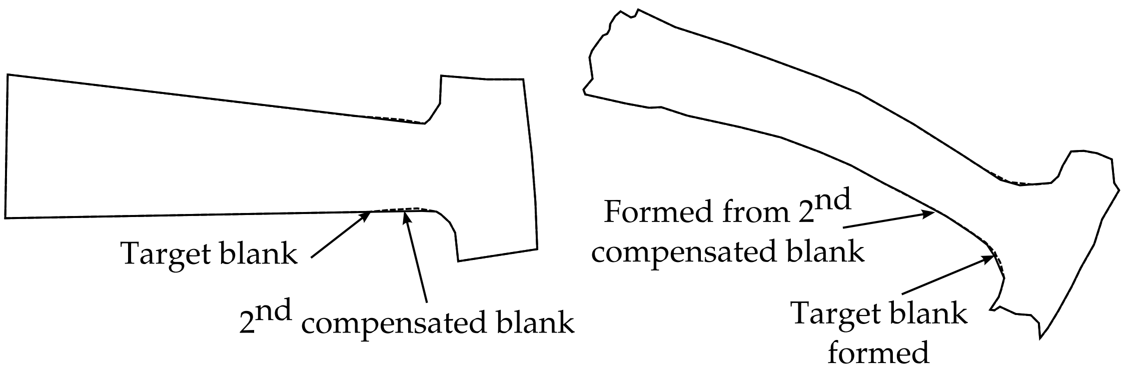


Figure 30-17. Iteration 2 results.

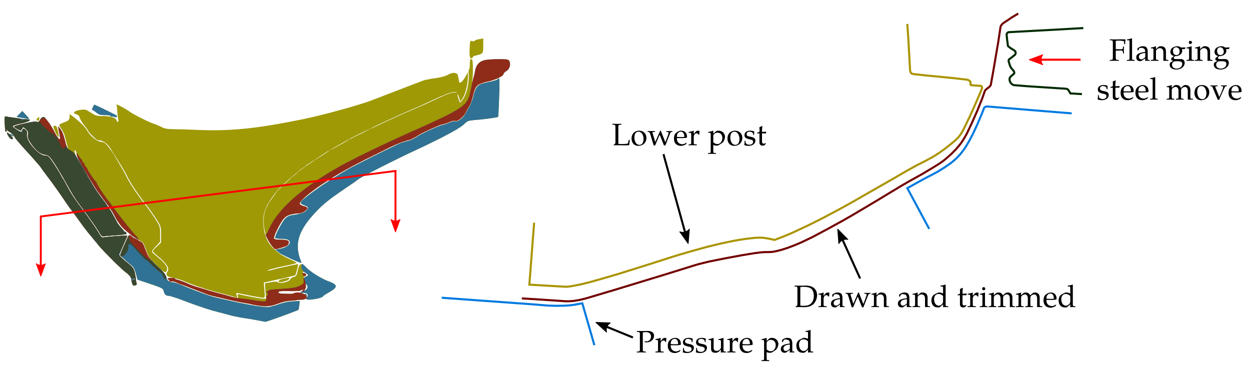


Figure 30-18. The flanging process on NUMISHEET 2002 fender outer.

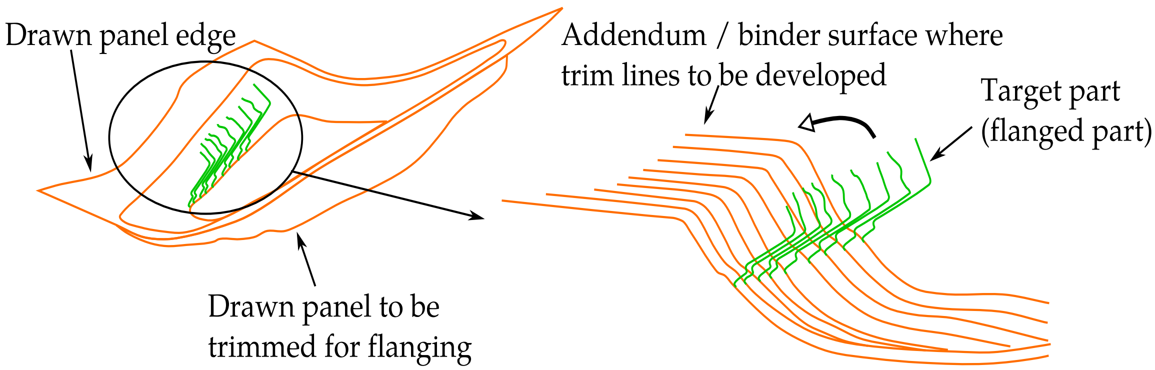


Figure 30-19. Multiple section view showing the target part and addendum surfaces.

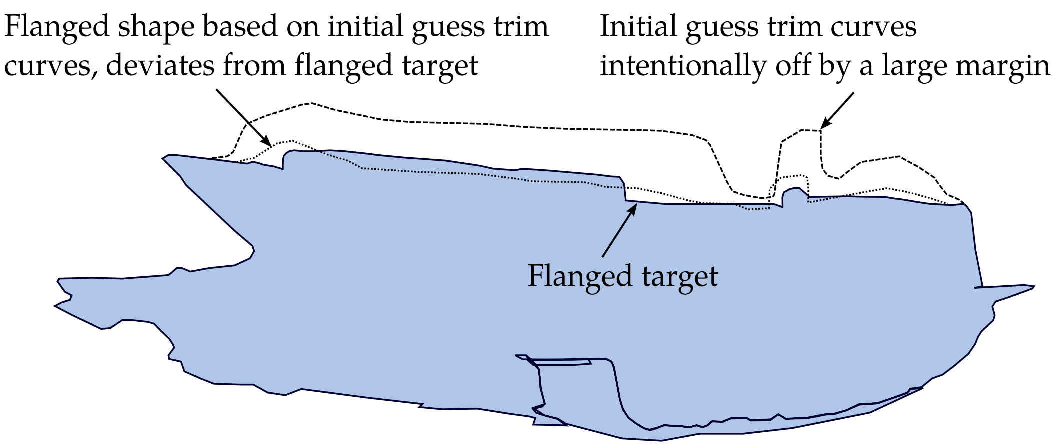


Figure 30-20. Initial trim curves intentionally made to be off by a large margin.

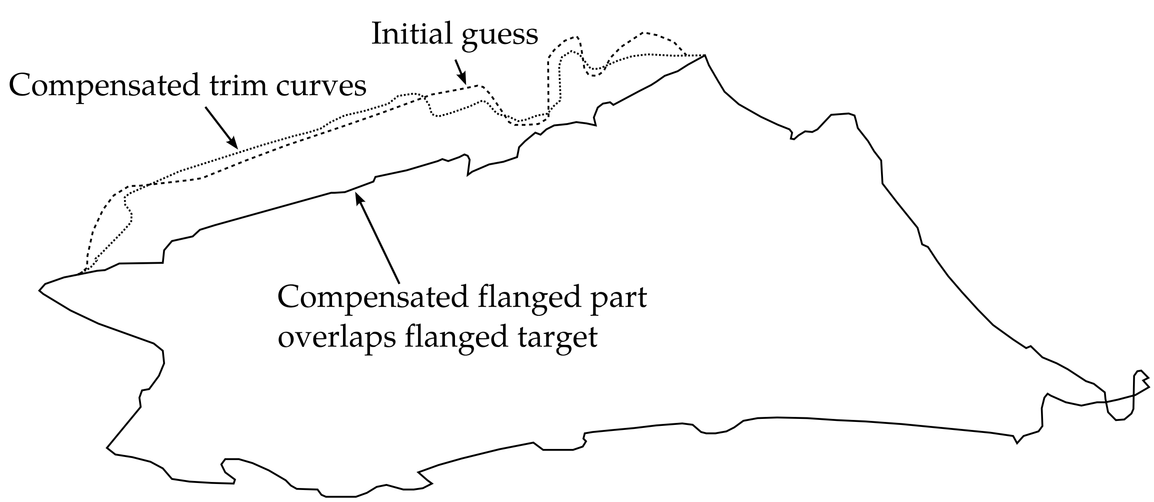
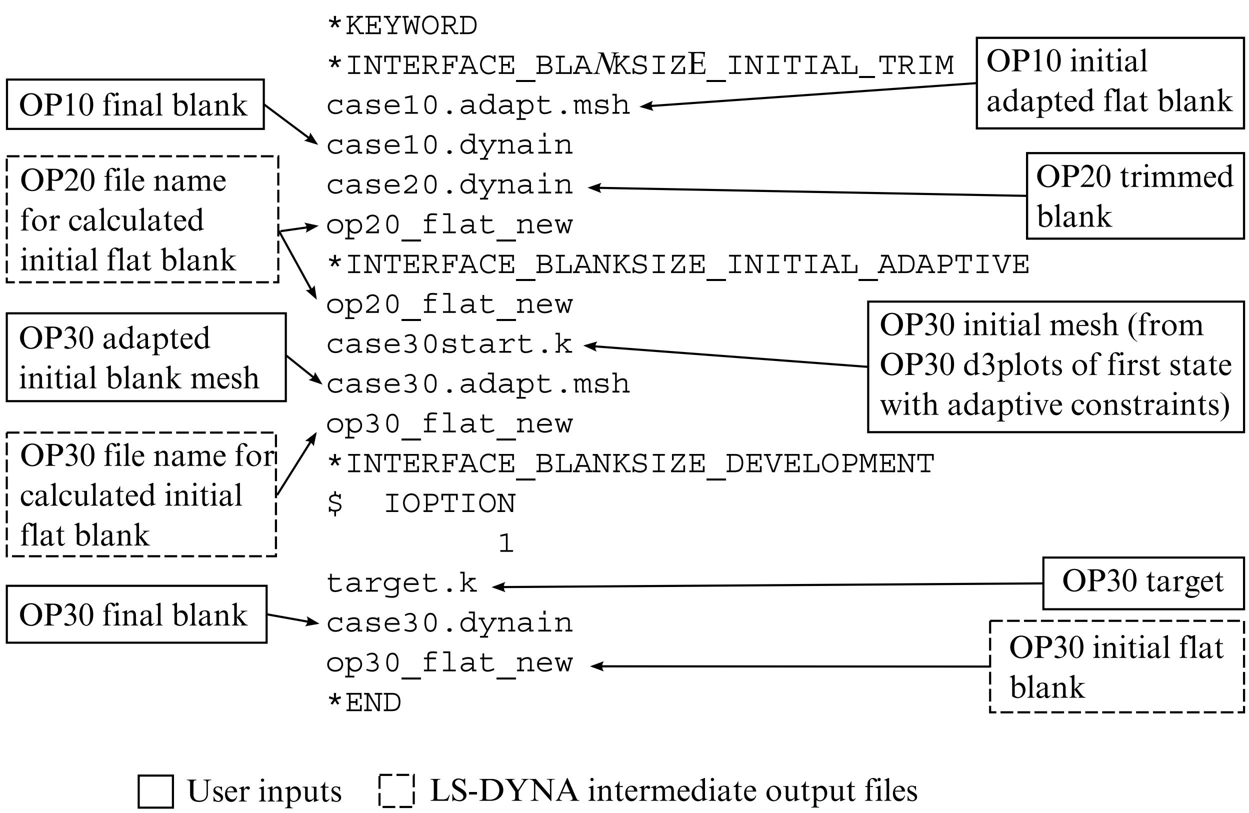


Figure 30-21. Compensated trim curves overlap with the target curves.



LS-DYNA simulation output: new trim line OP10 (file "trimcurves.ibo")

Figure 30-22. File structures for a multi-process blank development.

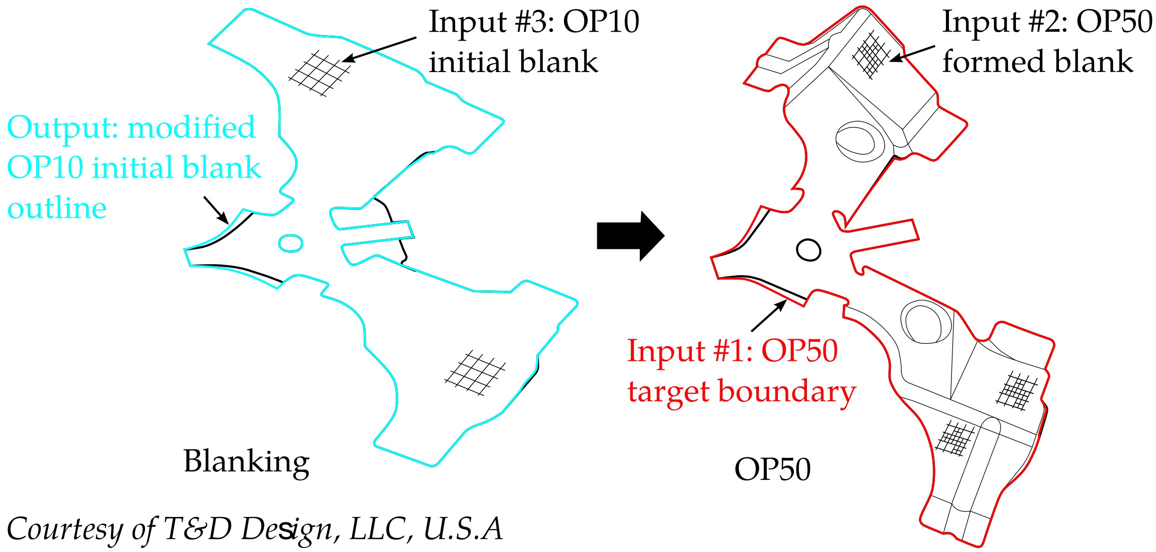


Figure 30-23. Inputs and output for the enhanced DEVELOPMENT feature.

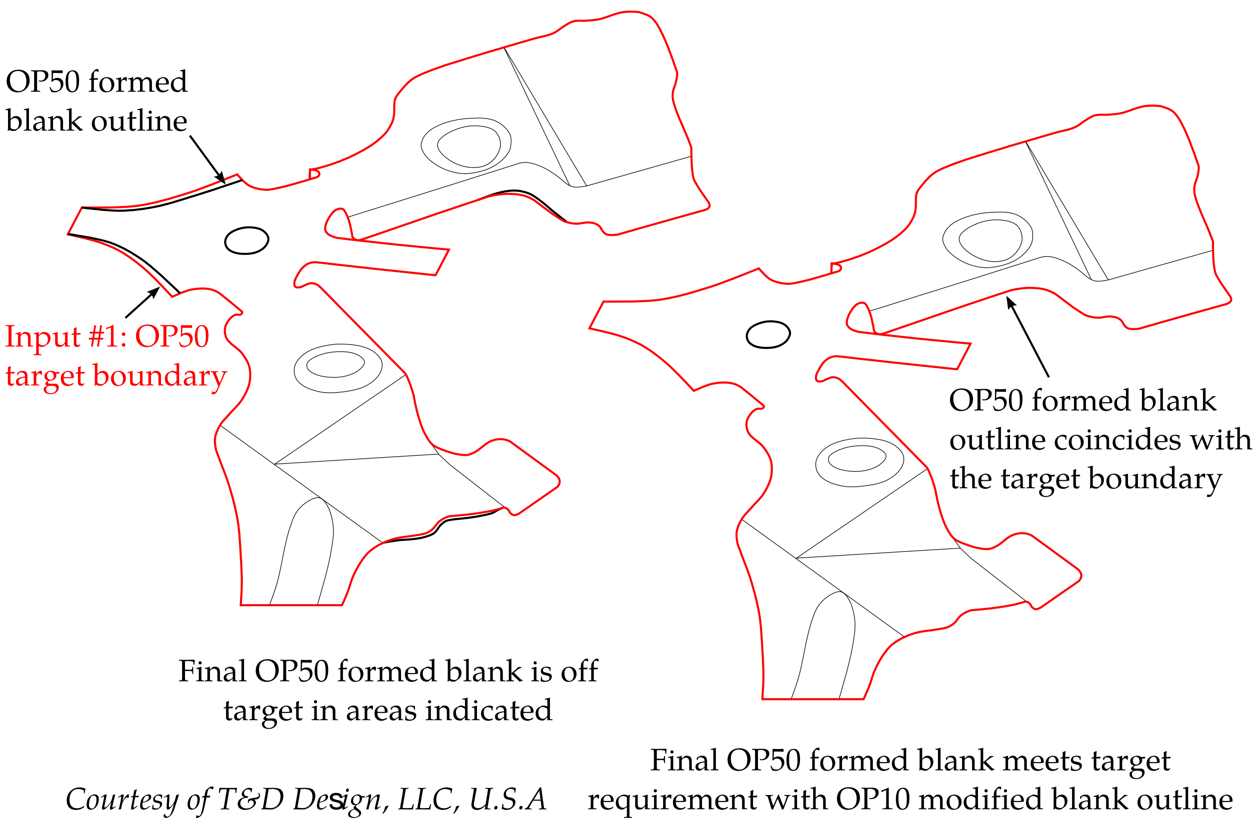


Figure 30-24. Verification simulation on a progressive die process.

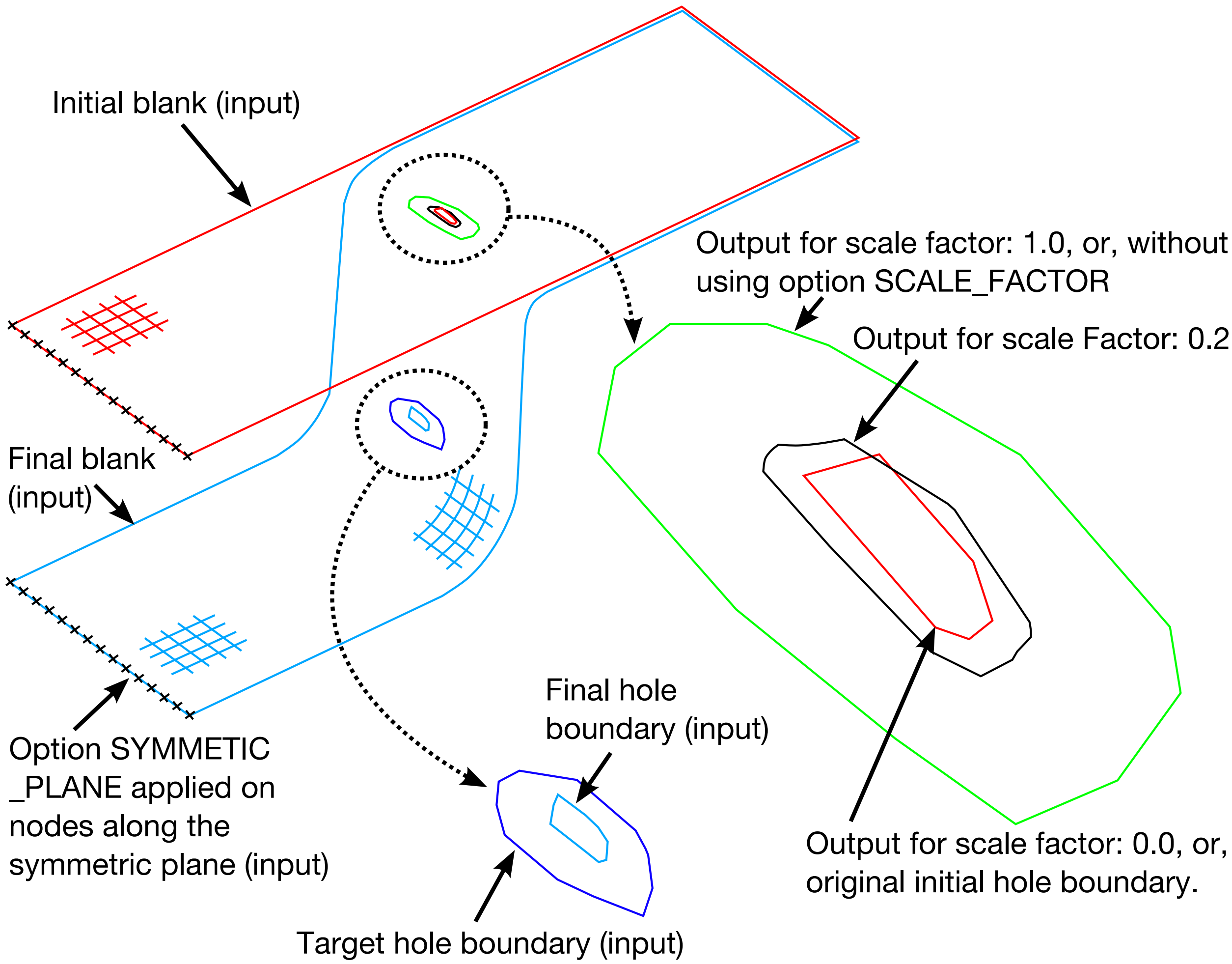


Figure 30-25. Options SCALE_FACTOR and SYMMETRIC_PLANE.

*INTERFACE_COMPENSATION_3D_{OPTION}

Available options include:

<BLANK>

ACCELERATOR

MULTI_STEPS

LOCAL_SMOOTH

PART_CHANGE

REFINE_RIGID

FLANGE

Purpose: The springback compensation module is implemented to automatically modify the rigid tools to compensate for the springback deviations. Since springback compensation is not a linear problem, an iterative approach is proposed that usually requires two to three iterations to bring the deviation below the tolerance.

To use this module, you need to provide the following information: a) the target, which is the deformed part with the original unmodified tool; b) the current tool file at its home or final position; c) the part shape formed with the current tool; d) the part shape after springback with the current tool; and e) a bridging file from the previous compensation, (for the first compensation the target file is used, then a file named disp.tmp is generated and will be used for the next iteration). To make the inclusion order-independent, five keywords are used to include the files mentioned above:

a) *INCLUDE_COMPENSATION_DESIRED_BLANK_SHAPE

b) *INCLUDE_COMPENSATION_CURRENT_TOOLS

c) *INCLUDE_COMPENSATION_BLANK_BEFORE_SPRINGBACK

d) *INCLUDE_COMPENSATION_AFTER_SPRINGBACK

e) *INCLUDE_COMPENSATION_COMPENSATED_SHAPE

Depending on the die set-up, different kinds of compensation are required. For example, sometimes the binder is not allowed to be modified. Various compensation methods, therefore, are available. The most used methods are 6, 7, and 8. Linear and nonlinear methods are also available to modify the tool surface outside the blank outline.

The applications for this card include: (1) calculating the deviation of the stamped part from its intended design and automatically compensating the tool to minimize the deviation; (2) mapping the existing trim curve to the modified tool; and (3) automatically detecting undercut. This keyword requires the double precision version of LS-DYNA. Note that the current methods sometimes fail to eliminate undercut.

NOTE:	The keyword *INTERFACE_COMPENSATION_3D replaces *INTERFACE_COMPENSATION_NEW starting in Revision 115517.

This feature is implemented and available in LS-PrePost under Application → Metal Forming →Easy Setup.

The ACCELERATOR option speeds up the convergence rate of compensation. This option also allows for a much simpler user interface.

The MULTI_STEPS option allows for tooling compensation of the next die process, based on target blank shape, compensated blank shape for the next step, and current tools. This feature is useful in line die process/tooling compensation.

The LOCAL_SMOOTH option features smoothing of a tool's local area mesh, which could become distorted because of a bad or coarse mesh of the original tool surface, a non-constant gap between tooling pairs (for example, flanging post and flanging steel), or a few compensation iterations. This option also requires using *SET_NODE_LIST_-SMOOTH.

The PART_CHANGE option allows for updating of the final compensated tool using the changed part or formed blank shape, thus eliminating the need for going through a new compensation iteration loop. This option is used together with *INCLUDE_COMPENSATION_UPDATED_BLANK_SHAPE, and *INCLUDE_COMPENSATION_UPDATED_RIGID_TOOL.

The REFINE_RIGID option refines the rigid tool mesh and aligns the nodes along the user-provided trim curves. The required input will be the original tool mesh and trim curves. This option ensures subsequent compensation of the tool to extend all the way to the trim curves. Note this option does not physically change the shape of the tool, but the refined and realigned tool mesh at the trim curves will greatly improve the compensation convergence in the iterative process. This option only needs to be done once before the iterative springback compensation begins.

The FLANGE option defines flanging steel's moving direction and can be used in springback compensation of a flanging process.

Card Summary:

Card 1a.1. This card is included if the option is <BLANK>, MULTI-STEPS, or LOCAL_-SMOOTH.

METHOD	SL	SF	ELREF	PSIDP	UNDCT	ANGLE	NLINEAR

Card 1a.2. This card is included if the option is <BLANK>.

TANGENT							

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

ISTEPS	TOLX	TOLY	TOLZ	OPTION			

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

MAXGAP							

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

FILENAME1

Card 1d.2. This card is included if the REFINE_RIGID option is used.

FILENAME2

Card 1e. This card is included if the FLANGE option is used. A maximum of two of this card may be inputted and the input ends with the next keyword ("*") card.

PID	VX	VY	VZ				

This card is included if the option is <BLANK>, MULTI-STEPS, and LOCAL_SMOOTH.

Card 1a.1	1	2	3	4	5	6	7	8
Variable	METHOD	SL	SF	ELREF	PSIDP	UNDCT	ANGLE	NLINEAR
Type	I	F	F	I	F	F	F	I
Default	6	5.0	0.75	1	none	0	0.0	1

VARIABLE	DESCRIPTION
METHOD	There are several extrapolation methods for the addendum and binder outside of trim lines; see Compensation Methods Overview.
SL	The smooth level parameter controls the smoothness of the modified surfaces. A large value makes the surface smoother. Typically, the value ranges from 5 to 10. If springback is large, the transition region is expected to be large. However, by using a smaller value of SL, the region of transition can be reduced.
SF	Shape compensation scale factor. The value scales the springback amount of the blank and the scaled amount is used to compensate the tooling. GT.0: Compensate in the opposite direction of the springback; LT.0: Compensate in the punch moving direction (for undercut). This scale factor determines how much the shape deviation is compensated, and a favorable value can reduce the number of iterations. However, the right value is case dependent. Usually, SF is chosen between the range of 0.75 and 1.25.
ELREF	Element refinement option: EQ.1: Special element refinement is used with the tool elements (default); EQ.2: Special element refinement is turned off.
PSIDP	Define the part set ID for primary parts of the tooling. Properly choosing the parts for the primary side is important since it affects what kinds of modifications will be made to the tooling. Usually, only one side of the tool will be chosen as the primary side, and the modifications made to the other side (secondary side) depend solely on the changes in the primary side. This specification allows the two sides to be coupled while maintaining a constant (tool) gap between the two sides. If both sides are chosen to be primary, the gap between the two sides might change and become inhomogeneous. When using METHOD 7, the choice of primary side will affect the result when applied to three-piece draw models. At this time, when the punch and binder are chosen as the primary side, the binder region will not be changed. Otherwise, when the die is chosen as primary side, the binder will be changed since the changes extend to the edges of the primary tool.

VARIABLE	DESCRIPTION
UNDCT	Tool undercut treatment option: EQ.0: No check (default) EQ.1: Check and fix undercut.
ANGLE	An angle defining the undercut.
NLINEAR	Activate nonlinear extrapolation

This card is included if the option is <BLANK>.

Card 1a.2	1	2	3	4	5	6	7	8
Variable	TANGENT							
Type	I							
Default	0							

VARIABLE	DESCRIPTION
TANGENT	A flag to maintain tangency during the compensation. Set TANGENT = 1 to maintain the tangential transition between the compensated and non-compensated areas of the rigid tool (for example, between addendum and binder in METHOD 7), and between the rigid tool area at the trim curves and addendum part of the tool. See Maintain Tangency.

This card is included if the ACCELERATOR option is used.

Card 1b	1	2	3	4	5	6	7	8
Variable	ISTEPS	TOLX	TOLY	TOLZ	OPTION			
Type	I	F	F	F	I			
Default	0	0.5	0.5	0.5	1			

VARIABLE	DESCRIPTION
ISTEPS	Number of compensation iterations for an accelerated compensation calculation. See Accelerated Springback Compensation.
TOLX	Part deviation tolerance between current blank and target blank shape in global x-direction.
TOLY	Part deviation tolerance between current blank and target blank shape in global y-direction.
TOLZ	Part deviation tolerance between current blank and target blank shape in global z-direction.
OPTION	Compensation acceleration method. Currently available only for method 1.

This card is included if the PART_CHANGE option is used.

Card 1c	1	2	3	4	5	6	7	8
Variable	MAXGAP							
Type	F							
Default	none							

VARIABLE	DESCRIPTION
MAXGAP	Maximum gap between the original part and changed part.

This card is used if the REFINE_RIGID option is used:

Card 1d.1	1	2	3	4	5	6	7	8
Variable	FILENAME1							
Type	A80							
Default	none							

VARIABLE	DESCRIPTION
FILENAME1	A keyword file name of the rigid tool mesh to be refined. This should be the tooling mesh used in the forming or flanging simulation, before any compensation is done. The refined rigid tool mesh will be in the file rigid_refined.tmp. See the Option REFINE_RIGID: Tool Mesh Refinement for a Better Convergence below.

This card is included if the REFINE_RIGID option is used.

Card 1d.2	1	2	3	4	5	6	7	8
Variable	FILENAME2							
Type	A80							
Default	none							

VARIABLE	DESCRIPTION
FILENAME2	A keyword file name with trim curves defined using *DEFINE_CURVE_TRIM_3D. The curves will be used to refine and realign the FILENAME1 to improve the convergence in the iterative compensation process. The refined rigid tool mesh will be in the file rigid_refined.tmp. See the Option REFINE_RIGID: Tool Mesh Refinement for a Better Convergence below.

This card is included if the FLANGE option is used. A maximum of two of this card may be inputted and the input ends with the next "*".

Card 1e	1	2	3	4	5	6	7	8
Variable	PID	VX	VY	VZ				
Type	I	F	F	F				
Default	none	none	none	none				

VARIABLE	DESCRIPTION
PID	The part ID of the flanging steel to be compensated.

VARIABLE	DESCRIPTION
VX, VY, VZ	The vector components of the flanging steel's (PID) moving direction.

Compensation Methods Overview:

After trimming, only a limited part of the tool has direct relationship with the springback of the blank part. Modifications of the rigid tool outside the trimmed region involves extrapolation. Unfortunately, extrapolating is unstable and tends to generate non-smooth surfaces. To resolve this problem, seven smoothing algorithms are implemented. The most frequently used methods are methods 7, 8 and -8, while the others are used only occasionally.

<u>Method 7</u>

If the punch is chosen as the primary side, the binder will not be changed. Aside from the region inside the punch opening, the rest of the model is untouched. Smoothing has little effect on method 7. The smoothness of the modified tool depends on the magnitude of the springback and the size of the addendum region. This method is nonlinear and, therefore, necessitates an iterative solve.

Advantages: The binder will not be changed.

Disadvantages: The change will be limited inside the addendum region, and the modified surface may not be smooth if the springback magnitude is large and the transition is small.

<u>Method 6</u>

The smoothness and the transition region of the modified surface will depend on the springback magnitude and the smoothing factor. If the springback magnitude is large, the transition region will be increased automatically. On the other hand, the transition region will be smaller if the springback magnitude is small. At the same time, a larger smoothing factor will result in a smaller transition region. Like method 7, this too is non-linear.

Advantages: The smoothness of the modified surfaces can be controlled.

Disadvantages: It is impossible to limit the transition region, and the binder surface (and therefore, draw beads) could change if the spring back is large.

<u>Method 3</u>

Similar to Method 6, however, it is a linear method and no iteration is necessary.

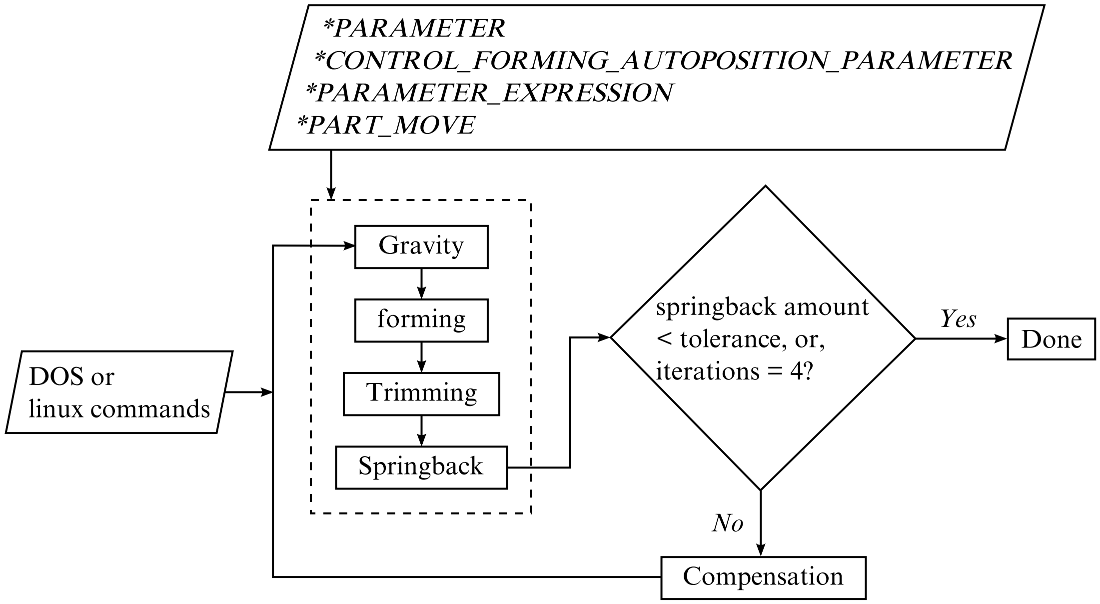


Figure 30-26. Iterative compensation flow chart

Method 8

This is an enhanced version of Method 6. It can account for addendum and binder changes. Usually, the upper tooling including addendum and binder (in an air draw) are included in the PSIDP definition.

Method -8

This method is a modification of Method 8. It is used for trim die nesting (from the drawn panel shape).

Methods 1, 2, 4, and 5

These methods are deprecated and may be removed in the future. They are included only for maintaining backwards compatibility.

Preventing Undercut:

When the draw wall is steep, it is likely that undercut will occur. Since undercut is not acceptable in real world die manufacturing, it must be prevented.

The compensation code can automatically detect undercut and issue a warning message. Additionally, LS-DYNA will write a list of undercut elements to a file called blankundercut.tmp so that the user can easily identify which elements may be problematic.

If the undercut is limited to only a few elements, it is possible to fix the problem manually.

Undercut can be reduced by compensating the springback only in the punch moving direction (by using a negative scale factor). This method is not 100% reliable and more robust solutions are being studied.

Iterative Springback Compensation:

Figure 30-26 illustrates the iterative springback compensation algorithm as applied to a typical stamping process. The first stamping process simulation is done following gravity → forming → trimming → springback (ITERATION 0). The stamping process simulation is set up using eZ-Setup in LS-PrePost. By using the parameterized automatic tool/blank positioning feature, the process simulation is fully automated (no user intervention required). Based on the calculated springback amount, tooling geometry is compensated through a compensation run. The stamping process simulation is conducted again, automatically, based on the new compensated tooling, followed by a second tooling compensation (ITERATION 1). Iterations 2, 3, and 4 follow the same pattern. The iteration process is repeated until the blank springback shape conforms to the target shape, or until it reaches 4 iterations (typically required to achieve part tolerance). With some shell scripting this iterative loop can be completed automatically. These tools allow the user to toggle between the single and double precision version of LS-DYNA.

The task of tracking the files involved in the iterative process can be daunting, especially in the advanced stage of the iterations. Figure 30-27 indicate what is written to storage during the process.

An input deck defining a springback compensation model is given below. The keyword file blank0.k includes node and element information of the blank shape before springback (after forming and trimming) with adaptive constraints (if exist). The keyword file spbk.k includes node and element information of the blank after springback, with adaptive constraints (if they exist). The blank shapes before and after springback (blank0.k and spbk.k) may be based on either the original die design (ITER0), or based on an intermediate compensated die design (say the n^{th} iteration).

The keyword file reference0.k is the blank shape before springback for iteration 0 (ITER0). This file is blank0.k and should *not* change from iteration to iteration. For iteration 0 the file reference1.k is also the same as blank0.k, but for iteration 1 reference1.k should be the disp.tmp generated from the compensation calculation during iteration 0 and so on and so forth for the subsequent iterations.

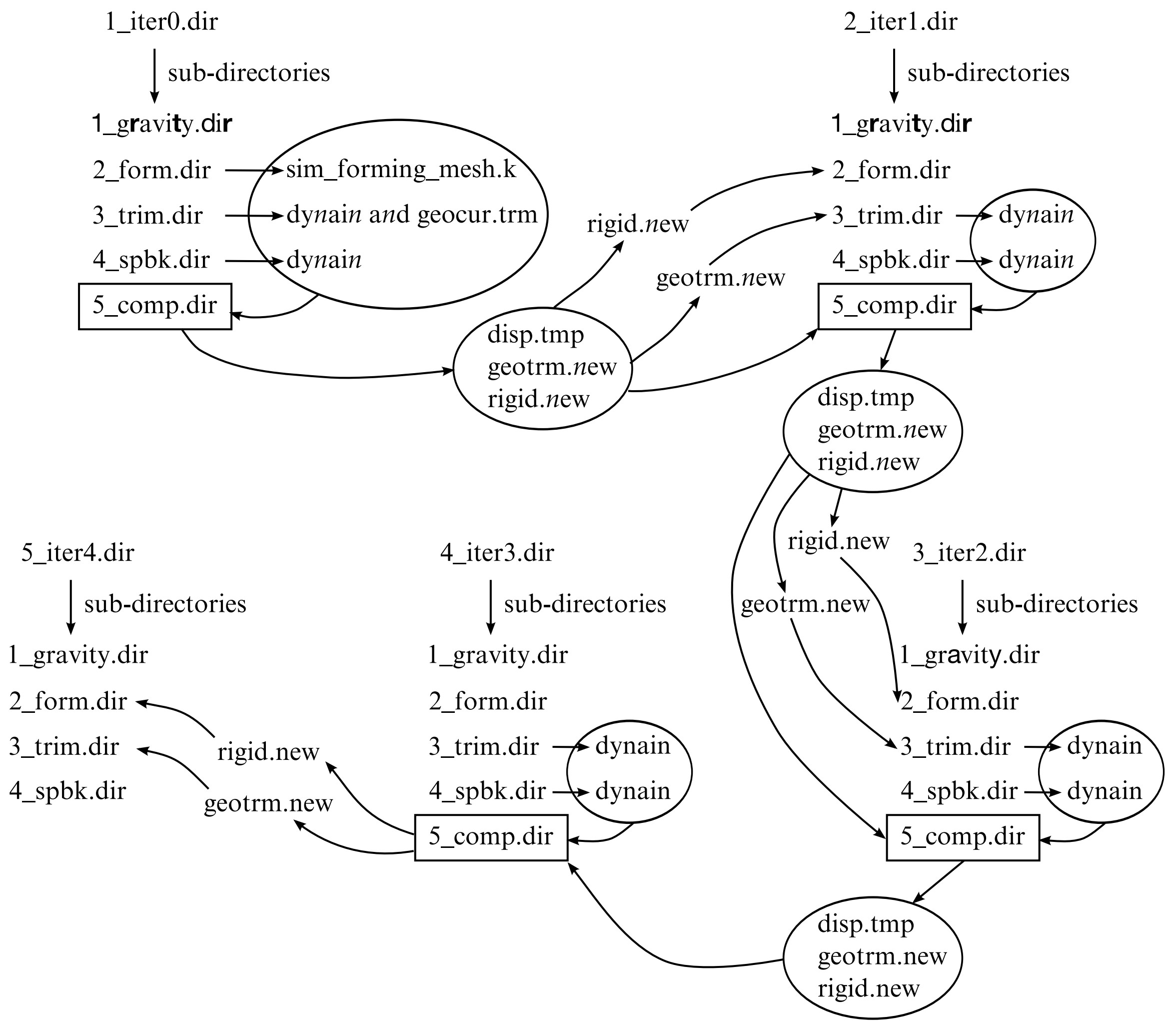


Figure 30-27. File structure for compensation

The keyword input tools.k must contain the mesh information for all of the stamping tools in their home positions. Compensated tools will be written to rigid.new with the original constant gap being maintained among the tools. During the baseline calculation, iteration 0, a keyword file called geocur.trm is generated during a LS-DYNA trimming simulation based on trimming curve input (usually in IGES format). During the compensation run of ITER1, geocur.trm is used to generate new trim curves which are output in geotrm.new. The new curves conform to the current compensated tools and are used for the next iteration. The new trim file, geotrm.new, is also in keyword format and contains a *DEFINE_CURVE_TRIM_3D card.

The example below models a three-piece air draw process. The upper die cavity (including binder) has a part ID 2, which is included in the part set ID 1 and is used for variable PSIDP. Method 8 will compensate all the tools included in file tools.k based on compensated shape for the upper cavity.

```
*KEYWORD
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+--
```

```
*INTERFACE_COMPENSATION_3D
$   METHOD        SL        SF     ELREF     PSIDP    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_SPRINBACK
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
geocur.trm
*SET_PART_LIST
$      PSID
         1
$      PID
         2
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+--
*END
```

An NUMISHEET 2005 Example:

In Figure 30-28, the NUMISHEET 2005 cross member is compensated following the flow chart. In two iterations the springback is reduced from 13mm to 1.7mm. Further iterations will reduce the part deviation down to a specific design target. Typically, four iterations are needed.

Iterative Compensation Applied During Die Construction:

The blank shape after springback can be obtained from the actual experimental shape of the springback panel, if available. For example, in hard tool construction, the trimmed panel can be scanned using white light technology and the panel shape can be written to an STL file. The STL format can be easily converted to LS-DYNA keyword format and the trimmed panel can be used as a rigid tool onto which the baseline (ITER0) trimmed panel (deformable) can be "pushed" using element normal pressure, and using *CONTROL_IMPLICIT_FORMING type 1. In this scenario, the adaptive refinement is turned off to maintain the one-to-one correspondence of the elements and nodes information. An advantage of this method is that the springback shape used for compensation will be exactly the same as the actual panel springback; therefore, the best tooling compensation result is expected. An example of such is shown in Figures 30-29 and 30-30.

Compensation of Localized Regions:

Compensation of a localized tooling region is possible, with the keyword *INCLUDE_-COMPENSATION_CURVE, by incorporating the following lines into the above example inputs:

```
*INCLUDE_COMPENSATION_CURVE
curves.k
```

The file curves.k defines the two enclosed "begin" and "end" curves using *DEFINE_-CURVE_COMPENSATION_CONSTRAINT_BEGIN/END. A more in depth explanation can be found in the corresponding keyword manual entries. In Figure 30-31, the NUMISHEET'05 decklid inner is compensated locally in the horizontal area above the backlite. Tangency of the compensated tool is maintained at the "End Curve" as shown in the section A-A. Also shown in Figure 30-32 includes color contours of part-separation distance throughout the iterations between compensated panel and the target design intent. Part tolerance is achieved in two iterations.

Accelerated Springback Compensation (ASC):

The option ACCELERATOR can be used in conjunction with *INCLUDE_COMPENSATION, with options ORIGINAL_DYNAIN and SPRINGBACK_INPUT to compensate springback with a faster convergence rate and a simplified user interface. A complete example is provided below. The example uses a spring back input file spbk.dyn, and a trimmed panel, with file name case20trimmed.dynain (including all stress and strain tensors and adaptive constraints). The variable ISTEPS is increased from 0 to 3, representing 3 compensation iterations. ISTEPS = 0 represents the baseline springback simulation (ITER0), while ISTEPS = 1, 2, 3 represent the compensation iterations. This feature requires the user to change only one variable (ISTEPS), and then submit the same input file to continue the next iteration.

Many scratch files, including a file named acceltmp.tmp, will be generated. *Do not delete them*. They are used to pass data between steps. A file, compensation.info, is generated and updated after each ISTEPS calculation. It contains iteration information, including maximum deviations in the x, y, and z directions. When the maximum deviation is within the tolerances specified in the TOLX, TOLY, and TOLZ fields, a message appears in the file indicating the compensation iterations have converged along with a message bootstrapping the next step. A file spbk.new is generated in the same directory and is used by the *INCLUDE_COMPENSATION_BLANK_AFTER_SPRINGBACK keyword with the scale factor for the tool compensation set to one. After the compensation, a verification calculation may be needed.

```
*KEYWORD
*INTERFACE_COMPENSATION_3D_ACCELERATOR
$   ISTEPS       TOLX       TOLY       TOLZ     OPTION
         3       0.20       0.20        0.2          1
*INCLUDE_COMPENSATION_ORIGINAL_DYNAIN
./case20trimmed.dynain
*INCLUDE_COMPENSATION_SPRING BACK_INPUT
./spbk.dyn
*END
```

Currently, mesh coarsening and checking are not supported in the accelerated mode. Also, the dynain file from the previous die process is not necessary.

An example of this feature is shown for a simple channel type of draw (one-half model) in Figure 30-33, which converged in three iterations; while four iterations were needed for the non-accelerated compensation.

Line Die Compensation:

The option MULTI_STEPS can be used together with *INCLUDE_COMPENSATION_-COMPENSATED_SHAPE_NEXT_STEP to enable compensation of tools for the next die process. An example is given below. In this example the target blank, named reference0.tmp, and the current tool named rigid.tmp come from the first die process. The disp.tmp file comes from the compensation in the second die process step. For example, a flanging die compensation can be a second die process step, preceded by a redraw die process as the first die process step.

```
*KEYWORD
*INTERFACE_COMPENSATION_3D_MULTI_STEPS
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$   METHOD        SL        SF     ELREF      PSID    UNDRCT     ANGLE   NLINEAR
         8     6.000      1.00         1         1         0         0         1
*INCLUDE_COMPENSATION_DESIRED_BLANK_SHAPE
reference0.tmp
*INCLUDE_COMPENSATION_COMPENSATED_SHAPE_NEXT_STEP
disp.tmp
*INCLUDE_COMPENSATION_CURRENT_TOOLS
rigid.tmp
*SET_PART_LIST
$     PSID
         1
$      PID
         2
*END
```

Compensation of Trim Dies (Trim Die Nesting):

The trim die can be compensated using the drawn panel's springback shape when METHOD is set to -8. In the example below, which is also shown in Figure 30-34, the draw panel, state1.k, is taken as the blank before springback, and draw panel springback shape, state2.k, is taken as the blank after springback. The tool shape for the draw process, drawtool.k, is used as the current tool. After the simulation, LS-DYNA will create a compensated tool named rigid.new, which can be used for the trim die shape.

```
*KEYWORD
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*INTERFACE_COMPENSATION_3D
$   METHOD        SL        SF     ELREF      PSID    UNDRCT     ANGLE   NLINEAR
        -8    10.000     1.000         2         1         0       0.0         1
*INCLUDE_COMPENSATION_BLANK_BEFORE_SPRINGBACK
state1.k
*INCLUDE_COMPENSATION_BLANK_AFTER_SPRINGBACK
state2.k
*INCLUDE_COMPENSATION_DESIRED_BLANK_SHAPE
ref0.tmp
```

```
*INCLUDE_COMPENSATION_COMPENSATED_SHAPE
ref1.tmp
*INCLUDE_COMPENSATION_CURRENT_TOOLS
drawtool.k
*INCLUDE_COMPENSATION_TRIM_CURVE
orginaltrim.k
*SET_PART_LIST
$      PSID
          1
$      PID
          3
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*END
```

Local Smoothing of Tooling Mesh:

The option LOCAL_SMOOTH can be used, along with a few more keywords, to smooth and restore the distorted tooling mesh after iterative compensation. In the example below, the keyword *INCLUDE_COMPENSATION_ORIGINAL_RIGID_TOOL includes an original tool, rigid.tmp, which has a good and smooth mesh. The keyword *INCLUDE_COMPENSATION_3D_RIGID_TOOL includes a compensated tool, rigidnew.bad, which could have a distorted mesh arising from the reasons listed when describing this option in the introduction section of this keyword.

The last keyword *SET_NODE_LIST_SMOOTH defines a node set that includes a distorted local area in the distorted mesh. Each node set defines a region needing smoothing. The node set should not include any boundary nodes of the tooling parts, otherwise position of the tooling may be altered. Smoothed tooling is stored in a file called rigid.new. In this example method 7 is active, the variable ELREF is set to 2, and PSID is left as undefined. Note also the *INCLUDE keyword is *not* supported here. For example, *SET_NODE_LIST_SMOOTH must not be in a separate file that is included in the main input file.

```
*KEYWORD
*INTERFACE_COMPENSATION_3D_LOCAL_SMOOTH
$   METHOD        SL        SF     ELREF      PSID    UNDRCT     ANGLE   NLINEAR
         7    10.000     1.000         2                   0       0.0         1
*INCLUDE_COMPENSATION_ORIGINAL_RIGID_TOOL
rigid.tmp
*INCLUDE_COMPENSATION_NEW_RIGID_TOOL
rigidnew.bad
*SET_NODE_LIST_SMOOTH
         1
     61057     61058     61059     61060     61061     61062     61063     61064
...
*SET_NODE_LIST_SMOOTH
         2
     56141     56142     56143     56144     56145     56146     56147     56148
...
*END
```

In an example shown in Figures 30-35 and 30-36, smoothing of the local mesh is performed in the draw bead area of the NUMISHEET 2005 cross member. In this case the die gap is not maintained throughout the tooling surface. Typically, this happens in the draw bead regions when male beads have lower bending radii (missing upper radii) and female beads have only upper bending radii (missing lower radii). Two node sets are defined for local areas of left and right female draw beads (Figure 30-37), which needed smoothing. It is important to include in the node sets some of the nodes on the relatively flat portion of the binder immediately off the bend radii. The smoothed results are shown in Figure 30-38.

In another example, a corner of a flanging die on a fender outer is being smoothed. The mesh becomes distorted after a few compensation iterations, as shown in Figure 30-39. In Figure 30-40, the result of local smoothing is shown, and the improvement is remarkable.

Compensation with Symmetric Boundary Condition:

A keyword example is provided in the manual pages related to *INCLUDE_COMPENSATION.

Global Compensation Using the Original Tool Mesh:

For some tooling meshes, the compensated die surfaces will be distorted. The keyword option *INCLUDE_COMPENSATION_ORIGINAL_TOOL causes the compensation code to use the original tooling mesh (starting in the second compensation) to extrapolate the addendum and binder in the compensated tooling surfaces. This minimizes the accumulative error, compared with using the last compensated tooling mesh, and therefore is a preferred method.

A complete keyword example is included below. In it, part ID 3 (included in part set ID 1) is compensated after ITERATION 3 (ITER3), using method 8, with a scale factor of 0.5. The dynain files from the trimmed ITER3 is taken as the “BEFORE” state (see the BEFORE_SPRINGBACK option) and the dynain file from the springback calculation is taken as the “AFTER” state. The “DESIRED” blank shape is given by the dynain is from the trimmed ITER0 output. The “COMPENSATED_SHAPE” is taken from the disp.tmp file of the last compensation run. “CURRENT_TOOL” is also from last compensation iteration. The “ORIGINAL_TOOL”, is taken from the tool mesh in ITER0. Updated trim curves geotrm.new are from the mapped trim lines of the last compensation.

It should be noted that, in an automatic compensation-loop calculation, as shown in the path of the input files, input files disp.tmp, rigid.new, and geotrm.new, which are the default file names of the previous compensation, should not be in the same directory as the current compensation run, as these files *will* be overwritten.

```
*KEYWORD
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*INTERFACE_COMPENSATION_3D
$   METHOD         SL         SF      ELREF       PSID     UNDRCT      ANGLE    NLINEAR
```

```
         8    10.000     0.500         2         1         1       0.0         1
*INCLUDE_COMPENSATION_BLANK_BEFORE_SPRINGBACK
../7_iter3.dir/2_trim.dir/dynain
*INCLUDE_COMPENSATION_BLANK_AFTER_SPRINGBACK
../7_iter3.dir/3_spbk.dir/dynain
*INCLUDE_COMPENSATION_DESIRED_BLANK_SHAPE
../1_iter0.dir/2_trim.dir/dynain
*INCLUDE_COMPENSATION_COMPENSATED_SHAPE
../6_compensation.dir/disp.tmp
*INCLUDE_COMPENSATION_CURRENT_TOOLS
../6_compensation.dir/rigid.new
*INCLUDE_COMPENSATION_ORIGINAL_TOOLS
../1_iter0.dir/sim_forming_mesh.k
*INCLUDE_COMPENSATION_TRIM_CURVE
../6_compensation.dir/geotrm.new
*SET_PART_LIST
$      PSID
         1
$      PID
         3
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*END
```

Updating Compensated Tool with Small Amount of Part Shape Change:

Often a part will have some small amount of shape change because of a product change. If the amount of shape change does not significantly alter the springback results, the compensated tools can be updated with the part mesh (inside the trim lines) or formed blank shape without going through another iterative compensation loop. This is accomplished using the PART_CHANGE option. Within the specified MAXGAP, the compensated tool shape can be updated. Changes to geometry involving sharp corners and transitions without fillets are not permissible. A complete keyword example is provided below, where a maximum gap of 5 mm is specified between the original shape and modified product shape. The updated part file name is updatepart.tmp and output file for the new rigid tool is newrigid.k.

```
*KEYWORD
*INTERFACE_COMPENSATION_3D_PART_CHANGE
$   MAXGAP
       5.0
*INCLUDE_COMPENSATION_DESIRED_BLANK_SHAPE
../1_iter0.dir/2_trim.dir/dynain
*INCLUDE_COMPENSATION_COMPENSATED_SHAPE
../6_compensation.dir/disp.tmp
*INCLUDE_COMPENSATION_CURRENT_TOOLS
../6_compensation.dir/rigid.new
*INCLUDE_COMPENSATION_UPDATED_BLANK_SHAPE
./updatedpart.tmp
*INCLUDE_COMPENSATION_UPDATED_RIGID_TOOL
$ file name to output the new rigid tools
./newrigid.k
*END
```

Option REFINE_RIGID: Tool Mesh Refinement for a Better Convergence:

The following keyword example refines the elements and aligns the nodes in the original tool mesh file rnominaltools.k at provided trim curves trimcurve106.k defined using the keyword *DEFINE_CURVE_TRIM_3D. The refined rigid tool mesh will be output as rigid_refined.tmp, which can be used to start the iterative springback compensation process. The aligned nodes will also be output in a file named bndnd0.tmp.

```
*KEYWORD
*INTERFACE_COMPENSATION_3D_REFINE_RIGID
Norminaltools.k
trimcurve106.k
*END
```

Maintain Tangency:

To maintain tangency between the addendum and binder (Method 7) in the draw die and tangential transitions immediately off the rigid tool at the trim curve area to the addendum, the variable TANGENT must be set to "1". In the keyword example below, the rigid_refined.tmp (from the option REFINE_RIGID) is used as both the current and the original tool. The aligned nodes file bndnd0.tmp is included under *INCLUDE_COMPENSATION_TRIM_NODE, which is used for tangency calculation. Note the variable TANGENT replaces the keyword *INCLUDE_COMPENSATION_TANGENT_CONSTRAINT.

```
*KEYWORD
*INTERFACE_COMPENSATION_3D
$   METHOD        SL        SF     ELREF       PSID    UNDRCT     ANGLE   NLINEAR
         7    10.000       1.0         2          1         0       0.0         1
$  TANGENT
         1
*INCLUDE_COMPENSATION_BLANK_BEFORE_SPRINGBACK
blank0A.tmp
*INCLUDE_COMPENSATION_BLANK_AFTER_SPRINGBACK
spbkA.tmp
*INCLUDE_COMPENSATION_DESIRED_BLANK_SHAPE
blank0A.tmp
*INCLUDE_COMPENSATION_COMPENSATED_SHAPE
blank0A.tmp
*INCLUDE_COMPENSATION_CURRENT_TOOLS
rigid_refined.tmp
*INCLUDE_COMPENSATION_ORIGINAL_TOOLS
rigid_refined.tmp
*INCLUDE_COMPENSATION_TRIM_NODE
bndnd0.tmp
*INCLUDE_COMPENSATION_TRIM_CURVE
../../1_iter0.dir/4_comp.dir/newtrmcurve.tmp
*INCLUDE_COMPENSATION_SYMMETRIC_LINES
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
     SYMID     SYMXY        X0        Y0
         1         2     -76.0       0.0
*SET_PART_LIST
$     PSID
         1
$      PID
         4
*END
```

Option FLANGE: Compensation of Flanging Process:

This option is developed specifically to compensate flanging tools for springback in a flanging process. A maximum of two cards can be input. The following keyword example defines flanging steels with PIDs 6 and 4 moving in the directions defined by the vectors [0.0,0.0, −1.0] and [0.0, −16.696, −5.674], respectively. The lower flanging post with PID2 is selected as the primary tool surface mesh and the flanging steels are compensated while maintaining the necessary tool gaps as in the original tool mesh iter0tools.k. Note in the compensation of flanging process the requirement on the even tool gap everywhere is removed, so flanging steels do not have to follow the flanging post, for example in the flanging steel radius area, etc.

```
*KEYWORD
*INTERFACE_COMPENSATION_3D
$   METHOD        SL        SF     ELREF      PSID    UNDRCT     ANGLE   NLINEAR
         8    10.000       1.0         2         1         0       0.0         1
*INCLUDE_COMPENSATION_BLANK_BEFORE_SPRINGBACK
blank0A.tmp
*INCLUDE_COMPENSATION_BLANK_AFTER_SPRINGBACK
spbkA.tmp
*INCLUDE_COMPENSATION_DESIRED_BLANK_SHAPE
blank0A.tmp
*INCLUDE_COMPENSATION_COMPENSATED_SHAPE
blank0A.tmp
*INCLUDE_COMPENSATION_CURRENT_TOOLS
iter0tools.k
*INCLUDE_COMPENSATION_ORIGINAL_TOOLS
iter0tools.k
*INTERFACE_COMPENSATION_3D_FLANGE
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$      PID        VX        VY        VZ
         6         0         0       -1.
         4         0   -16.696    -5.674
*SET_PART_LIST
$     PSID
         1
$      PID
         2
*END
```

Reference:

The manual pages related to *INCLUDE_COMPENSATION_{*OPTION*} can be further referenced for details.

Revision Information:

This keyword requires the double precision executable. Revision history is as follows:

1. Revision 61264: option ACCELERATOR.
2. Revision 61406: option MULTI_STEPS.
3. Revision 73850: option LOCAL_SMOOTH.

4. Revision 82698: option PART_CHANGE.

5. Revision 115517: *INTERFACE_COMPENSATION_3D replaces *INTERFACE_COMPENSATION_NEW.

6. Revision 113089: option REFINE_RIGID is available with *INTERFACE_COMPENSATION_NEW.

7. Revision 118400: option REFINE_RIGID is available with *INTERFACE_COMPENSATION_3D.

8. Revision 118490: option FLANGE.

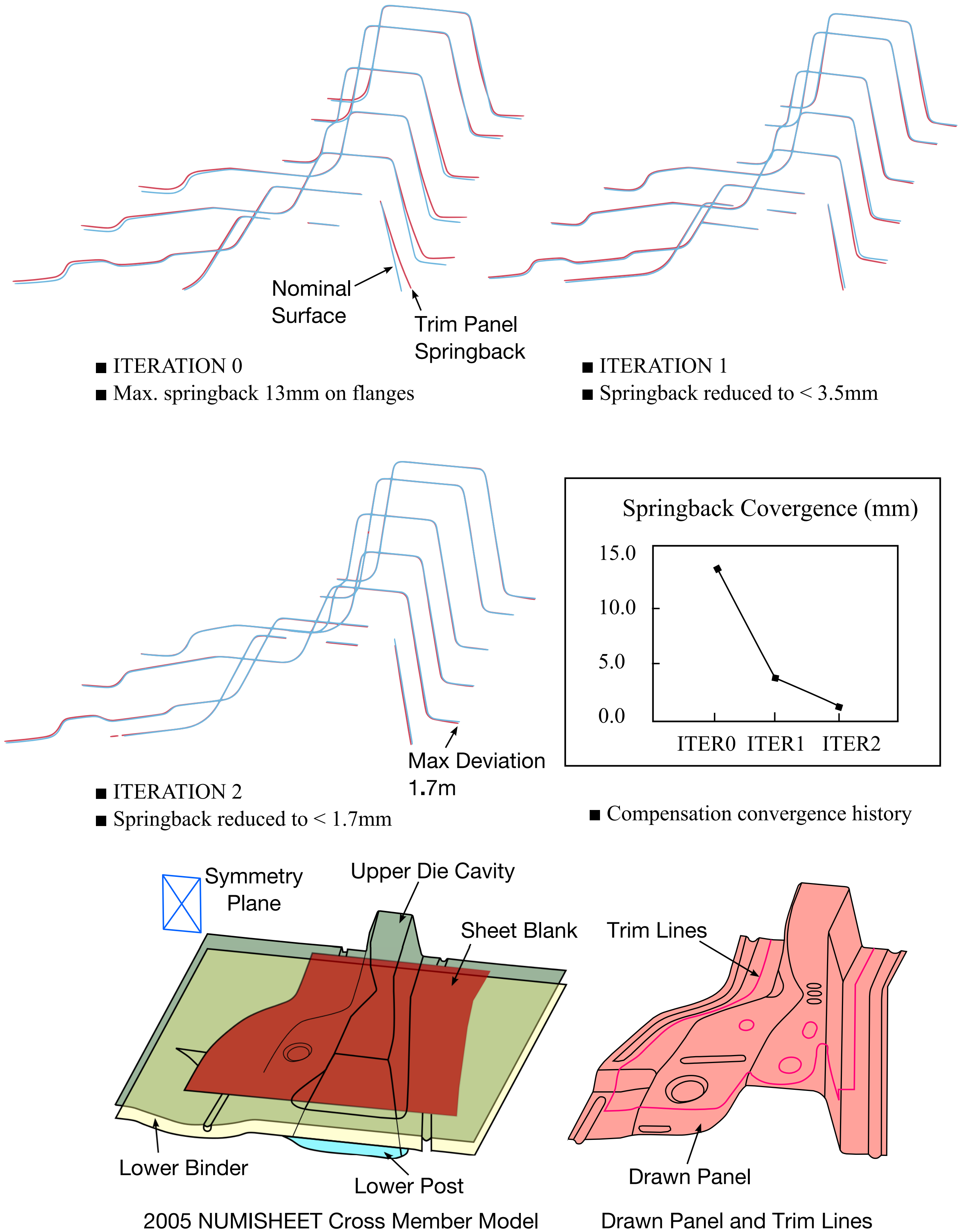


Figure 30-28. Iterative springback compensation on NUMISHEET'05 Xmbr(red – springback, blue – design intent)

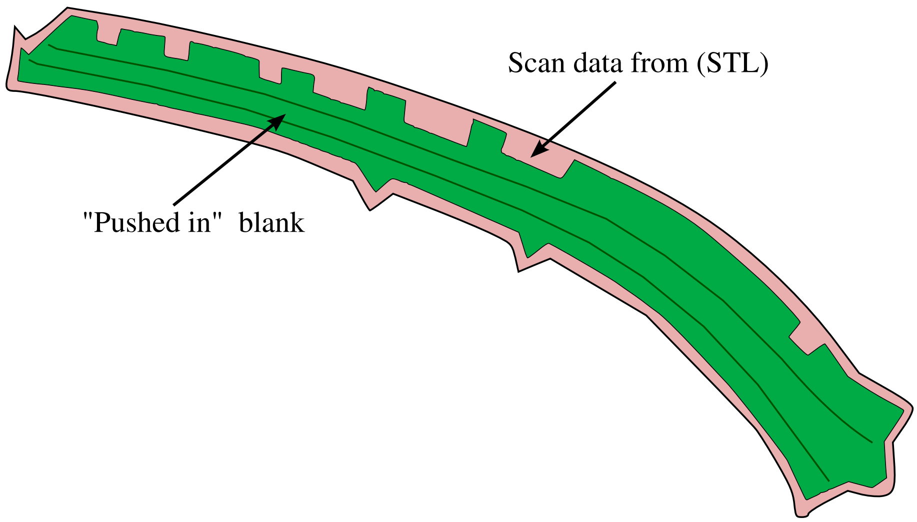


Figure 30-29. A trimmed panel “pushed” onto the scan data (rigid body).

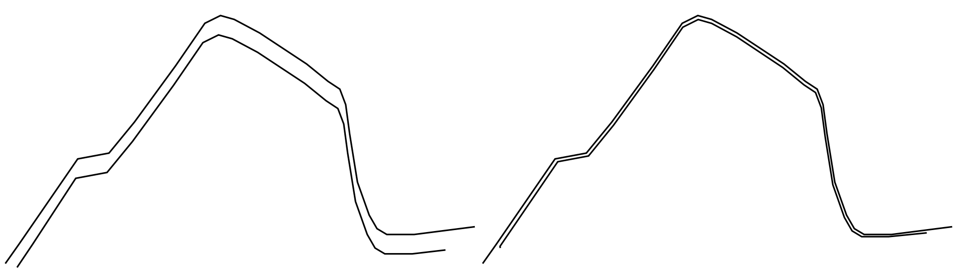


Figure 30-30. Section showing the “push” results – before and after.

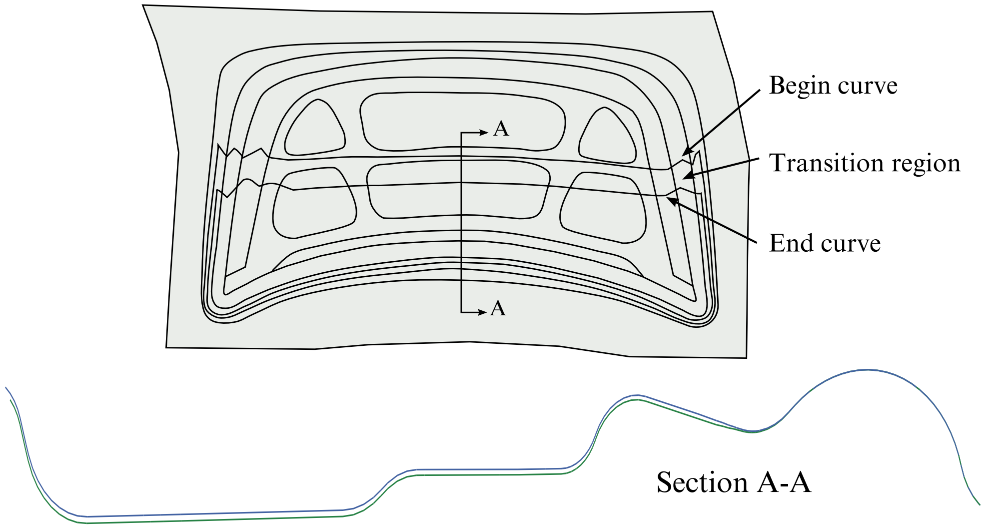


Figure 30-31. Two curves defining a localized area of a decklid inner.

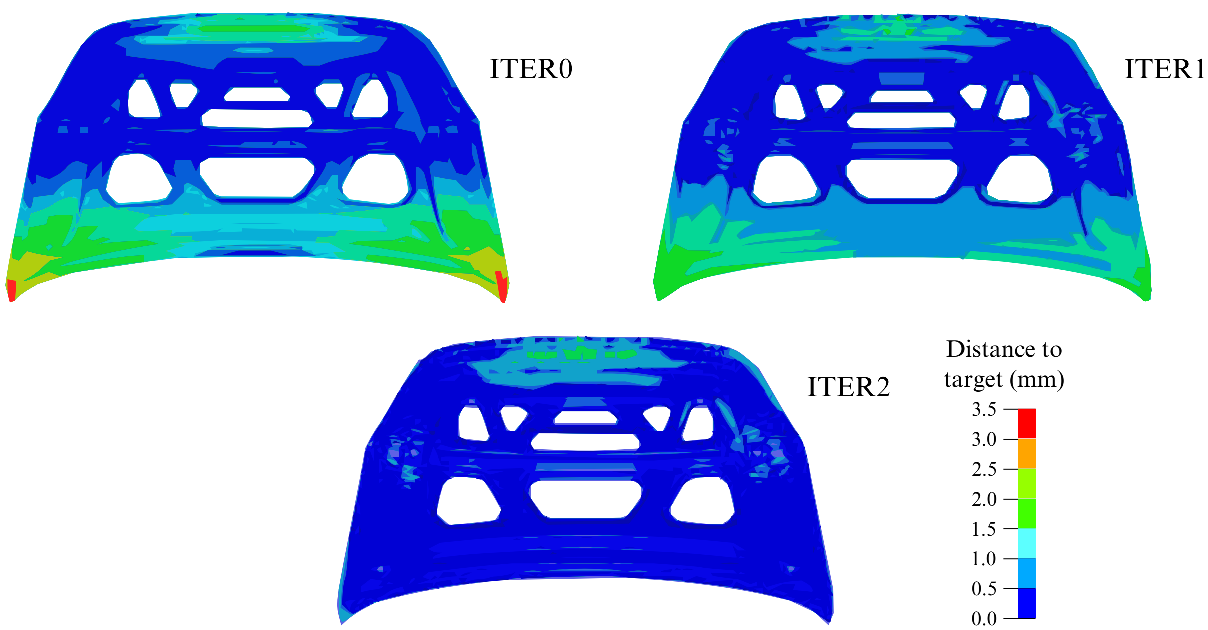


Figure 30-32. Iterative compensation for a localized (backlite) region.

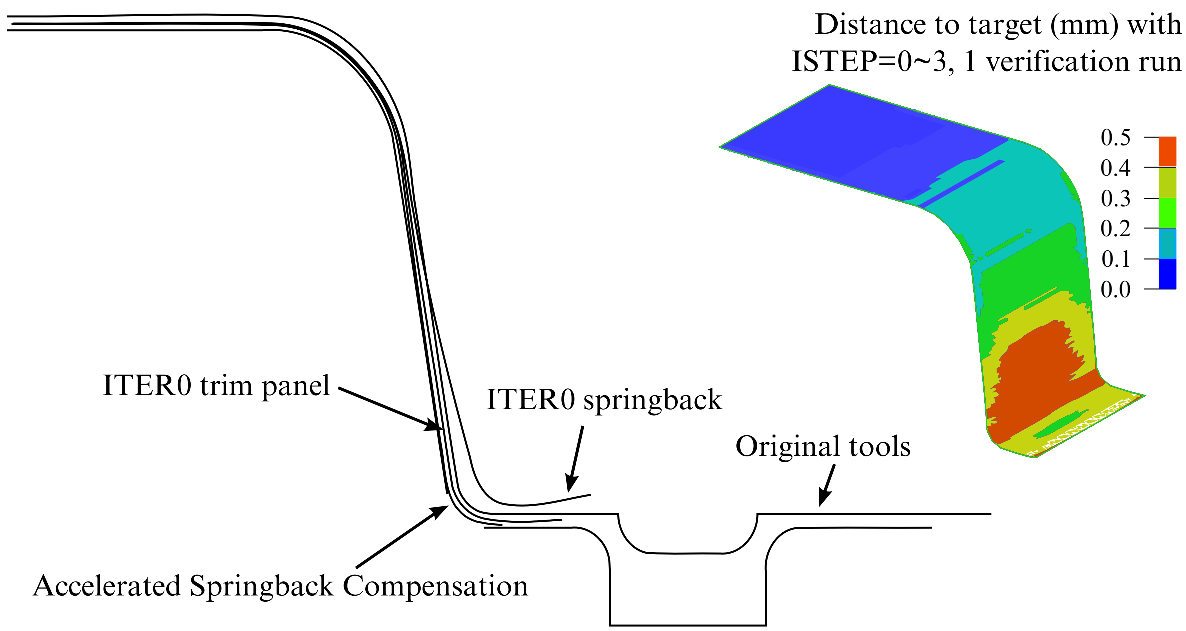


Figure 30-33. Accelerated Springback Compensation.

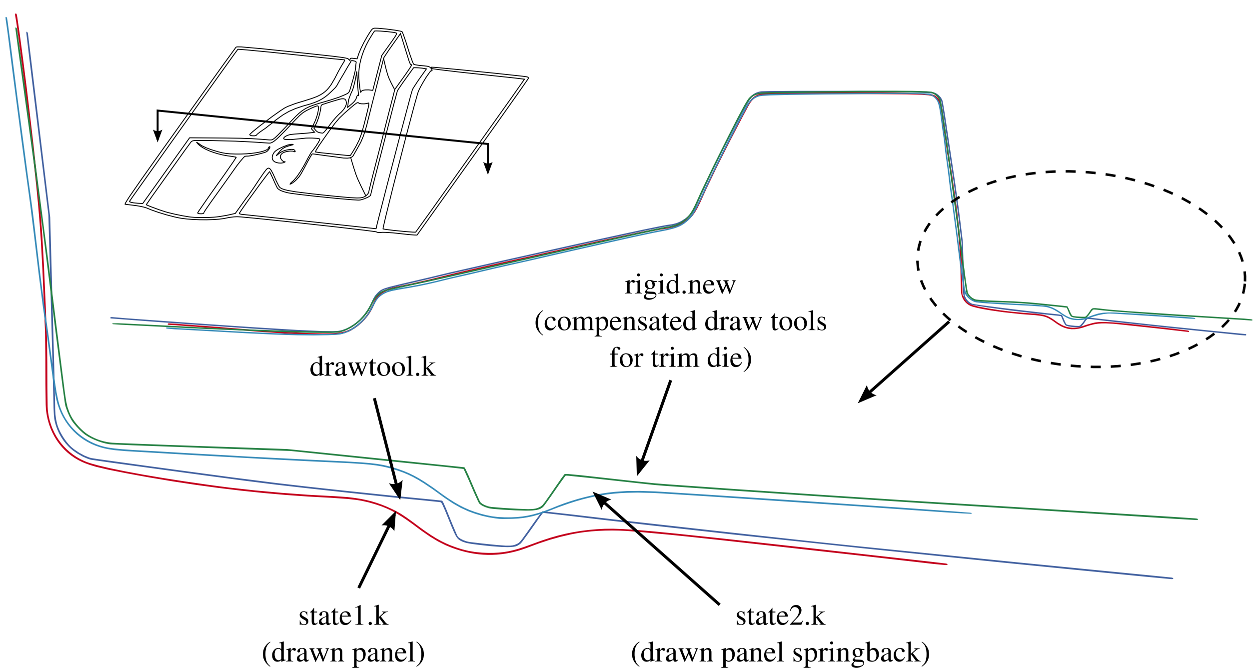


Figure 30-34. Trim die compensation with drawn panel springback shape.

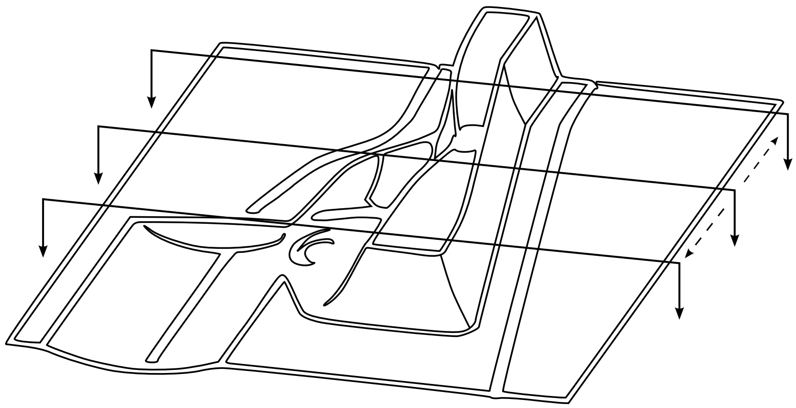


Figure 30-35. The NUMISHEET 2005 cross member.

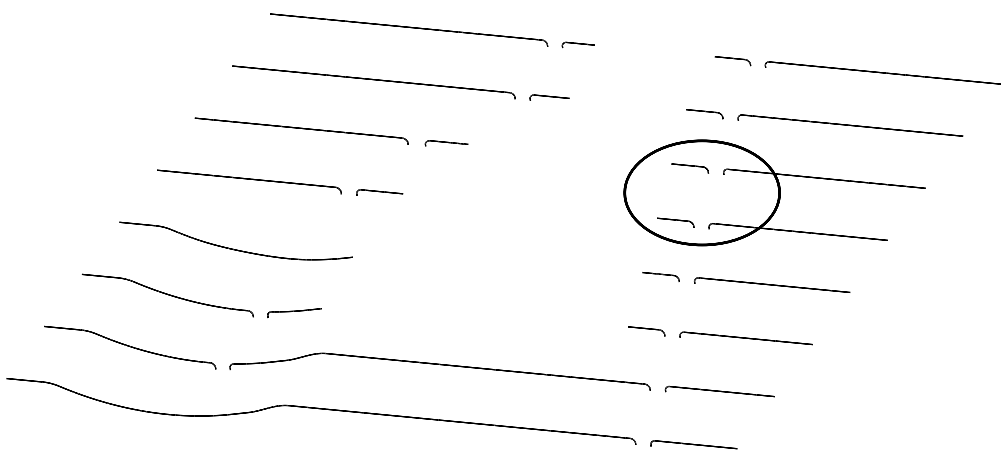


Figure 30-36. Multiple sections cut on the lower binder.

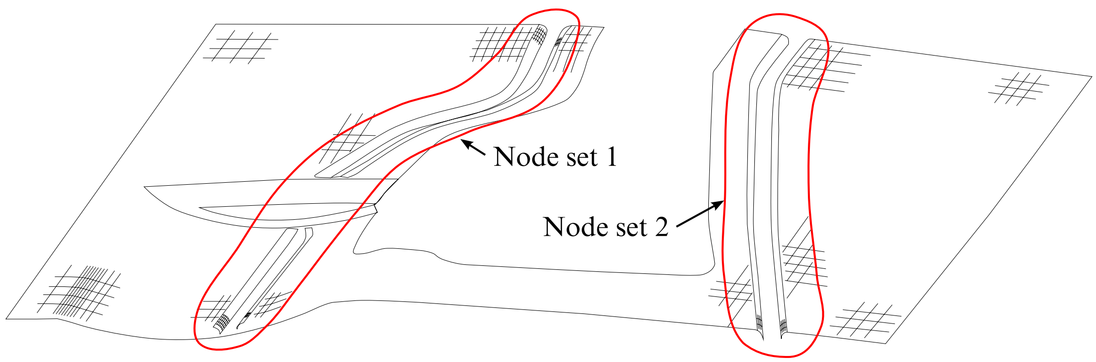


Figure 30-37. Local smoothing - two node sets defined including some nodes on the relatively flat binder area for both left and right draw beads.

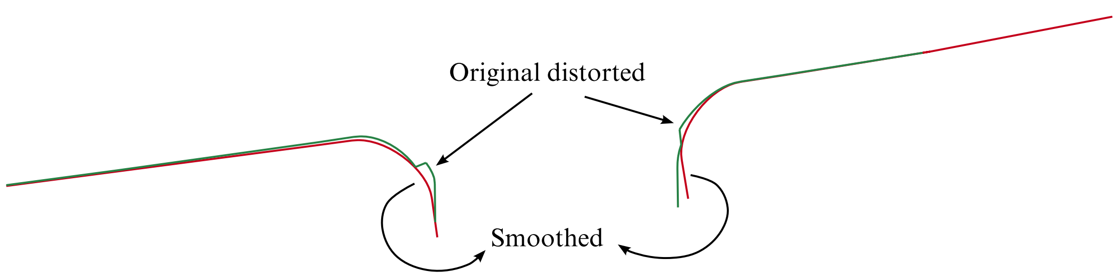


Figure 30-38. Comparison between original and smoothed tooling mesh.

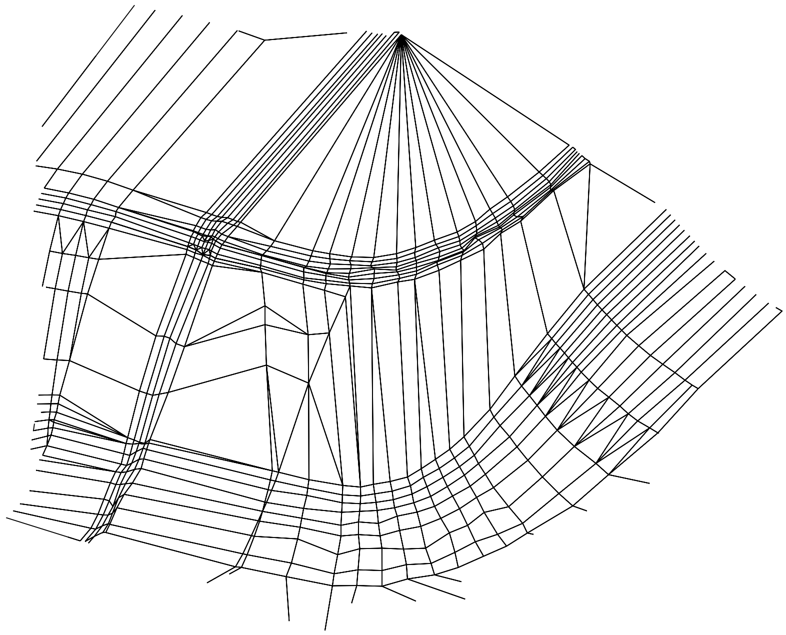


Figure 30-39. Original distorted tooling mesh.

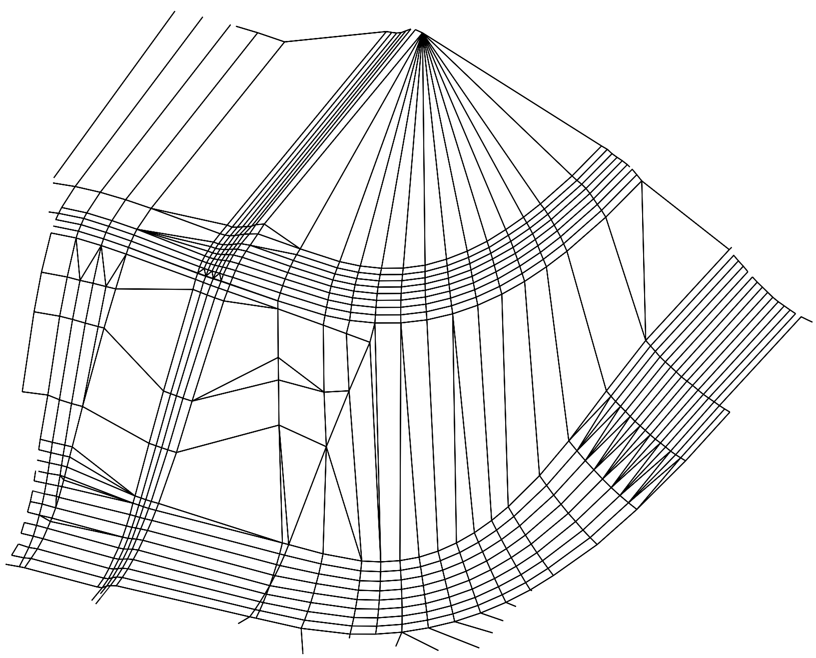


Figure 30-40. Smoothed tooling mesh.

*INTERFACE_COMPONENT_FILE

Purpose: Allow for the specification of the file where the component interface data should be written and the optional use of a new binary format for that data.

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

Optional Card.

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

VARIABLE	DESCRIPTION
FILENAME	Name of the file where the component data will be written
FORMAT	File format to use: EQ.1: Use old binary file format EQ.2: Use new LSDA file format
NDOF	Number of degrees of freedom per node that get written to the file: NE.3: Write as many as the model contains (3 or 6) EQ.3: Only write 3 even if the model has 6

Remarks:

If Z = is used on the command line, this card will be ignored. If this card is in effect, the new LSDA file format is the default format to be used. The new format has certain advantages and one possible drawback:

1. It allows for the use of the_TITLE modifier on all *INTERFACE_COMPONENT inputs so that subsequent *INTERFACE_LINKING cards can refer to components by a user-specified ID.

2. It is fully portable between machines with different precision and byte order.

3. It maintains the full precision of the coordinate vector. The internal coordinate vector has been in double precision for quite some time, even for single precision executables. The old binary format writes 32 bit data for single precision executables, losing some precision in the process.

4. Because of the maintained precision, the new format files will be significantly larger when running in single precision.

Of course, the new file format cannot be used for subsequent analysis with older versions of LS-DYNA, particularly those with a Product ID less than 50845. Executables that can read the new format for *INTERFACE_LINKING analysis automatically detect whether the new or old format is in use.

*INTERFACE_COMPONENT_*OPTION1*_{*OPTION2*}

Available values for OPTION1 include:

NODE

SEGMENT

SPH

OPTION2 only allows the value:

TITLE

Purpose: Create an interface for use in subsequent linking calculations. This command applies to the first analysis for storing interfaces in the interface file specified either by "Z=**isf1**" on the execution line or by the *INTERFACE_COMPONENT_FILE command. The output interval used to write data to the interface file is controlled by OPIFS on *CONTROL_OUTPUT. If OPIFS is not specified, the interval defaults to $1/10^{th}$ the value of DT specified in *DATABASE_BINARY_D3PLOT.

This capability allows the definition of interfaces that isolate critical components.

With the NODE and SEGMENT option, a database is created that records the motion of the interfaces. In later calculations the isolated components can be reanalyzed with arbitrarily refined meshes with the motion of their boundaries specified by the database created by this input. The interfaces defined here become the reference surfaces in the tied interface options.

With the SPH option, a database is created that records the FSI coupling forces of the interfaces. In later calculations the isolated components can be reanalyzed with arbitrarily refined meshes with external forces extracted from the database created by this input.

Each definition consists of a set of cards that define the interface. Interfaces may consists of a set of segments for later use with *INTERFACE_LINKING_SEGMENT, an ordered line of nodes for use with *INTERFACE_LINKING_EDGE, an unordered set of nodes for use with *INTERFACE_LINKING_NODE, or an unordered set of structure SPH particles for use with *INTERFACE_LINKING_FORCES.

Title Card. Additional card for TITLE keyword option.

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

VARIABLE	DESCRIPTION
ID	ID for this interface in the linking file
Title	Title for this interface

Card 2	1	2	3	4	5	6	7	8
Variable	SID	CID	NID					
Type	I	I	I					

VARIABLE	DESCRIPTION
SID	Set ID, see *SET_NODE (for NODE and SPH options) or *SET_-SEGMENT (for SEGMENT option).
CID	Coordinate system ID.
NID	Node ID.

Remarks:

CID and NID are optional. If CID appears, the transformation matrix for this coordinate system is written to the linking file at each output state. If NID appears, the displacement of this node is also written to the file. This information is then available to be used by the *INTEFACE_LINKING_NODE_LOCAL. If either of these is non-zero, then the linking file will be written in the LSDA format, as the old format cannot support this optional output.

If the old style binary format is used for the linking file (see *INTERFACE_COMPONENT_FILE) then the ID values are ignored and all components are numbered according to their input order, starting from 1.

*INTERFACE_DE_HBOND

Purpose: Define the failure models for bonds linking various discrete element (DE) parts within one heterogeneous bond (*DEFINE_DE_HBOND).

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

Bond Definition Cards. For each bond definition, include an additional card. This input ends at the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	PID1	PID2	PTYPE1	PTYPE2	FRMDL	FRGK	FRGS	DMG
Type	I	I	I	I	I	F	F	F
Default	none	none	none	none	1	none	none	1.0

VARIABLE	DESCRIPTION
IID	Interface ID. All interfaces should have a unique ID
PID1	First part ID.
PID2	Second part ID. PID1 and PID2 define the bonds that this fracture model is applied to. There are three combinations as Case a: PID1.EQ.0 This is the default model for all bonds, overriding the default model defined in Card 1.1 of *DEFINE_DE_HBOND. Case b: PID1.GT.0 and PID2.EQ.0 This model is applied to the bonds within part PID1, instead of the default model. Case c: PID1.GT.0 and PID2.GT.0

VARIABLE	DESCRIPTION
	This model is applied to the bonds between parts PID1 and PID2 only, but not to those within part PID1 or part PID2 (as in case b).
	Notes: 1. The default fracture model is applied to all parts that are not specified in case b. 2. The fracture model of the part with a smaller part id is applied to the bonds between two different parts if not specified in case c.
PTYPE1	First part type: EQ.0: DES part set EQ.1: DES part
PTYPE2	Second part type: EQ.0: DES part set EQ.1: DES part
FRMDL	Fracture model. (same as FRMDL in Card 1.1 of keyword *DEFINE_DE_HBOND.)
FRGK	Fracture energy release rate for volumetric deformation. (same as FRGK in Card 1.1 of keyword *DEFINE_DE_HBOND.)
FRGS	Fracture energy release rate for shear deformation. (same as FRGS in Card 1.1 of keyword *DEFINE_DE_HBOND.)
DMG	Continuous damage development. (same as DMG in Card 1.1 of keyword *DEFINE_DE_HBOND.)

INTERFACE_DE_HBOND EXAMPLE:

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$  *INTERFACE_DE_HBOND
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$  Using DE_HBOND to bond 4 parts with different failure models
$
*SET_PART_LIST_TITLE
DES BOND PARTS
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      sid       da1       da2       da3       da4
         1
$      id1       id2       id3       id4
       101       102       103       104
*SET_PART_LIST_TITLE
DES HBOND SUB PARTS
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      sid       da1       da2       da3       da4
         2
$      id1       id2
       103       104
*INTERFACE_DE_HBOND
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      iid
         1
$     pid1      pid2    ptype1    ptype2     frmdl      frgk      frgs       dmg
       101         0         1         0         1     1.0E1     1.0E1       1.0
         2         0         0         0         1     1.0E3     1.0E3       1.0
       102       103         1         1         2     1.0E2     1.0E2       1.0
$
*DEFINE_DE_HBOND
$
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>....8
$      sid     stype    bdform      idim
         1         2         2         3
$   pbk_sf    pbs_sf      frgk      frgs     bondr     alpha       dmg     frmdl
         1         1     1.0E0     1.0E0       2.5      0.01       1.0         1
$   precrk    cktype               itfid
         0         0                   1
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$ the fracture models for the bonds within each part are determined as
$
$   pid1    pid2    frgk    comments
$    101     101   1.0E1    Line 1 in the interface card
$    102     102   1.0E0    default (defined in hbond card)
$    103     103   1.0E3    Line 2 in the interface card
$    104     104   1.0E3    Line 2 in the interface card
$
$ for the bonds between two parts:
$    101     102   1.0E1    taken from Part 101 (smaller part id)
$    101     103   1.0E1    taken from Part 101 (smaller part id)
$    101     104   1.0E1    taken from Part 101 (smaller part id)
$    102     103   1.0E2    Line 3 in the interface card
$    102     104   1.0E0    taken from Part 102 (smaller part id)
$    103     104   1.0E3    taken from Part 103 (smaller part id)
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

INTERFACE_LINKING_DISCRETE_NODE_*OPTION

Available options include:

NODE

SET

Purpose: Link node(s) to an interface in an existing interface file. This link applies to all element types. The interface file is specified using *INTERFACE_LINKING_FILE or by including "L=**filename**" on the execution line.

With this command, nodes in a node set must be given in the same order as they appear in the interface file. This restriction does not apply to the more recent keyword *INTERFACE_LINKING_NODE_…

Card 1	1	2	3	4	5	6	7	8
Variable	NID/NSID	IFID						
Type	I	I						

VARIABLE	DESCRIPTION
NID	Node ID or Node set ID to be moved by interface file (see *NODE or *SET_NODE)
IFID	Interface ID in interface file

*INTERFACE_LINKING_EDGE

Purpose: Link a series of nodes to an interface in an existing interface file. The interface file is specified using *INTERFACE_LINKING_FILE or by including "L=**filename**" on the execution line.

Card	1	2	3	4	5	6	7	8
Variable	NSID	IFID						
Type	I	I						

VARIABLE	DESCRIPTION
NSID	Node set ID to be moved by interface file.
IFID	Interface ID in interface file.

Remarks:

The set of nodes defined will be constrained to follow the movement of the interface IFID, which should correspond to a curve output on a previous analysis via *INTERFACE_COMPONENT_NODE. The order of the nodes in the first analysis is important. The nodes, in the order specified in the first analysis, represent a curve. Each node in set NSID will be tied to the point on the curve nearest to its initial position, and then will be constrained to follow that point on the curve for the duration of the analysis. This option is intended to be used with beam or shell elements, as both translational and rotational degrees of freedom are constrained.

*INTERFACE_LINKING_FILE

Purpose: Allow for the specification of the file from which the component interface data should be read.

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

VARIABLE	DESCRIPTION
FNAME	Name of the file from which the component data will be read

Remarks:

If L= is used on the command line, this card will be ignored. There is no option to specify the file format, as the file format is automatically detected.

*INTERFACE_LINKING_FORCES

Purpose: Link segments to an interface in an existing interface file. The interface file is specified using *INTERFACE_LINKING_FILE or by including "L=**filename**" on the execution line.

Segment Set ID Card. Include as many cards as desired. Input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	SSID	IFID						
Type	I	I						

VARIABLE	DESCRIPTION
SSID	Segment set on which to apply forces from interface file
IFID	Interface ID in interface file

Remarks:

The set of segments defined will receive external forces coming from the interface IFID, which should correspond to an interface output on a previous analysis using the *INTERFACE_COMPONENT_SPH keyword. Positions from the interface file will be projected to the closest element in the segment set, and forces will be interpolated to their corner nodes.

INTERFACE_LINKING_NODE_*OPTION

Available options include:

SET

LOCAL

SET_LOCAL

Purpose: Link nodes(s) to an interface in an existing interface file. This link applies to all element types. The interface file is specified using *INTERFACE_LINKING_FILE or by including "L=**filename**" on the execution line.

WARNING:	Nodes constrained by the linking file should not be assigned additional constraints.

Node/Set ID Card. Include as many cards as desired. Input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	NID/NSID	IFID	FX	FY	FZ			
Type	I	I	I	I	I			

VARIABLE	DESCRIPTION
NID	Node ID or Node set ID to be moved by interface file. See *NODE or *SET_NODE.
IFID	Interface ID in interface file
FX	ID of a *DEFINE_FUNCTION which determines the x direction displacement scale factor. See Remarks.
FY	ID of a *DEFINE_FUNCTION which determines the y direction displacement scale factor. See Remarks.
FZ	ID of a *DEFINE_FUNCTION which determines the z direction displacement scale factor. See Remarks.

Card 2. This card appears after Card 1 when the LOCAL keyword option is used.

Card 2	1	2	3	4	5	6	7	8
Variable	LCID	LNID	USEC	USEN				
Type	I	I	I	I				

VARIABLE	DESCRIPTION
LCID	Local coordinate system ID for transforming displacements.
LNID	Local node ID for transforming displacements
USEC	Flag to indicate the use of the coordinate system in the linking file during displacement transformation. See Remarks. EQ.0: Linking file coordinate system is ignored. EQ.1: Linking file coordinate system is used.
USEN	Flag to indicate the use of the node displacement in the linking file during displacement transformation. See Remarks. EQ.0: Node displacement is not used. EQ.1: Node displacement is used.

Remarks:

This keyword constrains a set of nodes to follow the displacement of the interface having ID IFID in the linking file. Note that usually *INTERFACE_COMPONENT_NODE generates the linking file in a previous analysis.

The order of the nodes is not important. Each node in set NID will be tied to the nearest node in IFID using a bucket sort during the initialization phase. Nodes not found are reported and subsequently not constrained. Translational degrees of freedom are constrained. If the constrained analysis has rotational degrees of freedom, then the rotation degrees of freedom will be likewise constrained and the linking file *must* include rotational degrees of freedom.

The displacements in the linking file can be scaled upon input so that

$$\mathbf{u}_{\text{constrained}} = \begin{bmatrix} f_{\text{FX}}(\dots) & 0 & 0 \\ 0 & f_{\text{FY}}(\dots) & 0 \\ 0 & 0 & f_{\text{FZ}}(\dots) \end{bmatrix} \mathbf{u}_{\text{linking file}}$$

where $f_{\mathrm{FX}}(\dots)$ is the *DEFINE_FUNCTION function having ID FX and so on. When a scaling function is not specified, the corresponding component is imported unscaled as if the scaling function had a constant value of unity. These functions may take either 0, 1, 3, or 4 input arguments. The functions FX, FY, and FZ may be different and they may take different numbers of arguments. The data passed into the scaling function depends on the number of arguments that the function takes and the possibilities can be broken down into four cases:

1. *0 variables*. A function taking no inputs is constant over space and time. LS-DYNA evaluates such a function at the start of the calculation and uses that value for the duration of the run.

2. *1 variables.* LS-DYNA passes in the simulation time at each step and the resulting value is applied to all nodes in the set.

3. *3 variables.* LS-DYNA passes in the *initial position* of each constrained node as an (x, y, z) triple at the start of the calculation and then uses the result for the duration of the run.

4. *4 variables*. LS-DYNA passes in the *current* simulation time and the *initial position* of each of the constrained nodes as an (x, y, z, t) tuple. This function is updated at each time step. Using scaling functions of 4 variables may result in a performance penalty as each function must be evaluated for every constrained node every cycle.

If time dependent scaling functions are used, then the constrained nodes must start with coordinates identical to the constraining nodes in the linking file.

The LOCAL option and the values of the LCID, LNID, USEC, USEN flags, which was designed in conjunction with Honda R&D Co., Ltd., allow for the interface displacements to be transformed in various ways. By default, the scale factors FX, FY, and FZ act on the nodal displacements in the global coordinate system of the constrained calculation. This may be undesirable depending on how the global coordinate system of the linked calculation is defined. The most general transformation rule is:

$$\mathbf{u}_{\mathrm{constrained}} = \mathbf{Q}_2 \begin{bmatrix} f_{\mathrm{FX}}(\dots) & 0 & 0 \\ 0 & f_{\mathrm{FY}}(\dots) & 0 \\ 0 & 0 & f_{\mathrm{FZ}}(\dots) \end{bmatrix} \mathbf{Q_1}(\mathbf{u}_{\mathrm{linked}} - \mathbf{c_1}) + \mathbf{c_2}$$

where,

$\mathbf{c}_1$ = The displacement of the NID node in the linking file
$\mathbf{c_2}$ = The displacement of node LNID
$\mathbf{Q}_1$ = Rotation into the local coordinates of the linking file
$\mathbf{Q}_2$ = Rotation into the local coordinate system; if unset the inverse of $\mathbf{Q}_1$

If USEC = 0, then $\mathbf{Q}_1$ is the identity rotation and any coordinate system in the linking file is ignored. If USEN = 0, then $\mathbf{c}_1$ is set to $\mathbf{0}$ and NID in the linking file is ignored.

*INTERFACE_LINKING_SEGMENT

Purpose: Link segments to an interface in an existing interface file. The interface file is specified using *INTERFACE_LINKING_FILE or by including "L=**filename**" on the execution line.

Segment Set ID Card. Include as many cards as desired. Input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	SSID	IFID						
Type	I	I						

VARIABLE	DESCRIPTION
SSID	Segment set to be moved by interface file
IFID	Interface ID in interface file

Remarks:

The set of segments defined will be constrained to follow the movement of the interface IFID, which should correspond to an interface output on a previous analysis using the *INTERFACE_COMPONENT_SEGMENT keyword. The behavior will be the same as if set SSID is the SURFA side of a *CONTACT_TIED_SURFACE_TO_SURFACE with IFID as the SURFB side. Translational movement will be constrained but not rotations.

*INTERFACE_SPG_1

Purpose: Indicate that the current analysis is the first stage of a two-stage SPG analysis. This keyword does not affect the results of the SPG analysis.

The two-stage SPG analysis feature is available beginning with R13. Starting with R15, more internal variables other than the stresses and effective plastic strain can be transferred between stages by setting ISPGHIS = 1.

This card is optional.

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

VARIABLE	DESCRIPTION
ISPGHIS	Option to transfer more internal variables required by the material model other than the stresses and effective plastic strain: EQ.0: Only transfer the stresses and effective plastic strain. EQ.1: Include all other internal variables required by the material model.

Remarks:

1. **Output file.** When this keyword is included, LS-DYNA generates an ASCII file named 1234spg at normal termination. This file contains information (stress, strain, total displacement, etc.) about the SPG nodes at the end of the calculation.

2. **Dynain file for the second stage.** The second stage also requires a dynain file as input. Therefore, the first stage input file must include *INTERFACE_SPRINGBACK_LSDYNA to output this file. The part set for *INTERFACE_SPRINGBACK_LSDYNA must include the SPG parts since the SPG coordinates at termination are needed.

*INTERFACE_SPG_2

Purpose: Indicate the current analysis is the second stage of a two-stage SPG analysis.

The two-stage SPG analysis is only available for versions R13 and later. Starting with R15, more internal variables other than the stresses and effective plastic strain can be transferred between stages by setting ISPGHIS = 1.

This card is optional.

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

VARIABLE	DESCRIPTION
ISPGHIS	Option to read more internal variables other than the stresses and effective plastic strain if ISPGHIS = 1 in the first stage: EQ.0: Read only the stresses and effective plastic strain. EQ.1: Read all other internal variables required by the material model. The output file from the first stage only includes these internal history variables if ISPGHIS = 1 on *INTERFACE_SPG_1.

Remarks:

To perform a second stage analysis, the ASCII file 1234spg generated in the first stage calculation must be copied to the current folder and renamed as 1234spg0. The dynain file generated in the first stage calculation must also be included in the input deck for the second stage (see *INCLUDE_*{OPTION}*). The element connectivity and nodal coordinates for the second stage are taken from the dynain file.

*INTERFACE_SPRINGBACK_*OPTION1*_*{OPTION2}*

Available options included for *OPTION1* are:

LSDYNA

NASTRAN

SEAMLESS

EXCLUDE (see Remark 9)

and for *OPTION2:*

<BLANK>

NOTHICKNESS (see Remark 1)

Purpose: This keyword causes LS-DYNA to, upon termination, write the final state of the system to an output file so that the results can be used as input for other subsequent calculations, such as trimming, springback, flanging, and hemming. Also, if a sense switch "swb", "swc", or "swd" is issued while the analysis is in progress, the current state will be written to an output file (see "SENSE SWITCH CONTROLS" in the Getting Started section for details). In addition to the analysis results, the output can optionally contain additional nodal constraints. The keyword name, which contains SPRINGBACK, is due to historical reasons as this feature is not limited to springback applications. All the options, except SEAMLESS, control the output of the forming information.

With the SEAMLESS option, LS-DYNA automatically runs the springback analysis after the termination time of the explicit or implicit dynamic analysis. See Remarks for Seamless Springback for more details.

Card Summary:

Card 1. This card is required.

PSID	NSHV	FTYPE		FTENSR	NTHHSV	RFLAG	INTSTRN

Card 2. This card is optional. For LS-DYNA to interpret the card as this card, include the string "OPTCARD" in the first column. Otherwise, the input reader assumes the second card is Card 4.

OPTC	SLDO	NCYC	FSPLIT	NDFLAG	CFLAG	HFLAG	

Card 3.1. This card is optional. To include it, Card 2 and 3.2 must also be included. For LS-DYNA to interpret the card as this card, include the string "OPTCARD" in the first column. Otherwise, the input reader assumes the third card is Card 4.

OPTC	DTWRT						

Card 3.2. This card is optional. It must be included if using Card 3.1. For LS-DYNA to interpret the card as this card, include the string "OPTCARD" in the first column.

OPTC	NMWRT	IVFLG					

Card 4. Include as many instantiations of this card as needed. The next keyword ("*") card terminates this input.

NID	TC	RC					

Data Card Definitions:

Card 1	1	2	3	4	5	6	7	8
Variable	PSID	NSHV	FTYPE		FTENSR	NTHHSV	RFLAG	INTSTRN
Type	I	I	I		I	I	I	I

VARIABLE	DESCRIPTION
PSID	Part set ID. See *SET_PART.
NSHV	Number of shell or solid history variables (beyond the six stresses and effective plastic strain) to be initialized in the interface file. For solids, LS-DYNA writes one additional state variable (initial volume). If NSHV is nonzero, the element formulations, unit system, and constitutive models should not change between runs. If NHSV exceeds the number of integration point history variables required by the constitutive model, LS-DYNA only writes the number required. Thus, if in doubt, set NHSV to a large number, such as 100.
FTYPE	File type (see Remark 6): EQ.0: ASCII EQ.1: Binary EQ.2: Both ASCII and binary.

VARIABLE	DESCRIPTION
	EQ.3: LSDA format (for LSDYNA option only) EQ.10: ASCII large format (see *INITIAL_STRESS_SHELL) EQ.11: Binary large format EQ.12: Both ASCII and binary large format
FTENSR	Flag for dumping tensor data from the element history variables into the dynain file. EQ.0: Don't dump tensor data from element history variables EQ.1: Dump any tensor data from element history variables into the dynain file in GLOBAL coordinate system. Currently, only Material 190 supports this option.
NTHHSV	Number of thermal history variables.
RFLAG	Flag to carry over reference quantities, such as the reference geometry for hyperelastic materials: EQ.0: Default, do not output. EQ.1: Output reference coordinates and nodal masses.
INTSTRN	Output of strains for shells; see also *INITIAL_STRAIN_SHELL. EQ.0: Only at the innermost and outermost integration points, EQ.1: At all through-thickness integration points.

Irregular Optional Card 1. The keyword reader will interpret the card following Card 1 as optional Card 2 if the first column of the card is occupied by the string "OPTCARD". Otherwise, it is interpreted as the first Node Card; see below.

Card 2	1	2	3	4	5	6	7	8
Variable	OPTC	SLDO	NCYC	FSPLIT	NDFLAG	CFLAG	HFLAG	
Type	A	I	I	I	I	I	I	

VARIABLE	DESCRIPTION
SLDO	Output of solid element data as EQ.0: *ELEMENT_SOLID, or EQ.1: *ELEMENT_SOLID_ORTHO (only for anisotropic

VARIABLE	DESCRIPTION
	material).
NCYC	Number of process cycles this simulation corresponds to in the simulation of wear processes; see Remark 7.
FSPLIT	Flag for splitting of the dynain file (only for ASCII format): EQ.0: dynain file written in one piece. EQ.1: Output is divided into two files: dynain_geo including the geometry data and dynain_ini including initial stresses and strains.
NDFLAG	Flag to dump nodes into dynain file: EQ.0: Default, dump only SPH and element nodes EQ.1: Dump all nodes
CFLAG	Output contact state: EQ.0: Default, do not output. EQ.1: Output contact state, currently only Mortar segment pair information and selected tied contacts with restrictions; see Remark 8.
HFLAG	Output hourglass state, only valid for FTYPE = 3: EQ.0: Default, do not output. EQ.1: Output hourglass stresses for carrying over to the next simulation.

Irregular Optional Card 2. This card must have the string "OPTCARD" in the first column for LS-DYNA to interpret it as an optional card instead of the first Node Card.

Card 3.1	1	2	3	4	5	6	7	8
Variable	OPTC	DTWRT						
Type	A	F						

Irregular Optional Card 3. This card must have the string "OPTCARD" in the first column for LS-DYNA to interpret it as an optional card instead of the first Node Card.

Card 3.2	1	2	3	4	5	6	7	8
Variable	OPTC	NMWRT	IVFLG					
Type	A	I	I					

VARIABLE	DESCRIPTION
DTWRT	Time interval between two consecutive writes of the dynain.lsda file.
NMWRT	Maximum number of dynain.lsda files created. To save space, LS-DYNA periodically overwrites these files.
IVFLG	Flag to *not* write out the initial volume for solid elements. See IVEFLG on *INITIAL_STRESS_SOLID. EQ.0: IVEFLG = 1 for the written *INITIAL_STRESS_SOLID keywords. The output file contains the initial volume. EQ.1: IVEFLG = 0 for the written *INITIAL_STRESS_SOLID keywords. The output file does not contain the initial volume. .

Node Cards. Define a list of nodal points and constraints. For the SEAMLESS Option the constraints are used for the springback analysis. This input ends at the next keyword ("*") card.

Card 4	1	2	3	4	5	6	7	8
Variable	NID	TC	RC					
Type	I	F	F					

VARIABLE	DESCRIPTION
NID	Node ID, see *NODE.
TC	Translational Constraint: EQ.0: No constraints

VARIABLE	DESCRIPTION
	EQ.1: Constrained x displacement EQ.2: Constrained y displacement EQ.3: Constrained z displacement EQ.4: Constrained x and y displacements EQ.5: Constrained y and z displacements EQ.6: Constrained z and x displacements EQ.7: Constrained x, y, and z displacements
RC	Rotational constraint: EQ.0: No constraints EQ.1: Constrained x rotation EQ.2: Constrained y rotation EQ.3: Constrained z rotation EQ.4: Constrained x and y rotations EQ.5: Constrained y and z rotations EQ.6: Constrained z and x rotations EQ.7: Constrained x, y, and z rotations

Remarks:

1. **NOTHICKNESS option.** The NOTHICKNESS option is available when the keyword's first option is either LSDYNA or NASTRAN. With the NOTHICKNESS option the shell element thickness is not output.

2. **Filenames.** The file name for the LSDYNA option is dynain and for NASTRAN is nastin.

3. **Trimming.** Trimming is available for the adaptive mesh, but it requires manual intervention. To trim an adaptive mesh, use the following procedure:

   a) Generate the file, dynain, using the keyword *INTERFACE_SPRINGBACK_LSDYNA.

   b) Prepare a new input deck including the dynain file.

   c) Add the keyword *ELEMENT_TRIM to this new deck.

d) Add the keyword *DEFINE_CURVE_TRIM to this new deck.

e) Run this new input deck with i=**input_file_name**. The adaptive constraints are eliminated by remeshing and the trimming is performed.

f) In case this new trimmed mesh is needed, run a zero termination time job and output the file generated using the keyword, *INTERFACE_SPRING-BACK_LSDYNA.

4. **Temperature.** The file new_temp_ic.inc will be created for a thermal solution and a coupled thermal-mechanical solution. The file new_temp_ic.inc is a KEYWORD include file containing *new temperature initial conditions* for the nodes belonging to the PSID.

   a) For thermal user materials it is possible to dump thermal history variables. See the NTHHSV field.

5. **Velocity.** Nodal velocities from the final state are included in the dynain file when using the LSDA format (FTYPE = 3), or during a staged construction analysis (see *CONTROL_STAGED_CONSTRUCTION), but are otherwise not output.

6. **FTYPE.** The choice of format size in field FTYPE is only available for shell stresses and shell history data; see parameter LARGE on *INITIAL_STRESS_-SHELL. For solid and beam elements, the large format is always written to dynain; that is LARGE is automatically set to 1 on *INITIAL_STRESS_SOLID and *INITIAL_STRESS_BEAM, respectively.

   When you include a dynain file generated with a single precision executable in a subsequent double precision analysis, or vice versa, the dynain file should be in either ASCII or LSDA format. The binary file format is not compatible between single and double precision executables.

7. **NCYC.** When simulating wear processes, this represents the number of process cycles this particular simulation corresponds to and *INITIAL_CONTACT_-WEAR cards are generated accordingly in the dynain file (only ASCII format supported). Cards will only be generated for nodes in contact interfaces associated with a *CONTACT_ADD_WEAR, and having SAPR or SBPR set to 2 on the first card on *CONTACT. This wear data is in a subsequent simulation accounted for NCYC times when modifying the worn geometry, or alternatively processed in LS-PrePost.

8. **CFLAG.** Transferring contact state, particularly tied information, may be important for multistage simulations. Setting CFLAG to 1 will output Mortar contact segment pairs for sliding, tied, tiebreak, and tied weld options. The resulting dynain.lsda, which must use the LSDA format (FTYPE = 3), can be

included in a subsequent simulation to restore the contact state. The LS-DYNA version or core count as wells as the contact type (for instance changing from tied weld to simple tied) may be changed between simulations if the contact ID and node IDs making up the segments remain unchanged. If parts or elements are removed between simulations and these happen to contain segments that are in the dynain.lsda file, these pairs will be simply ignored. This option provides a flexibility that is appealing for multistage processes. For sliding contact, typically the ignore and friction state is transferred. Concerning non-Mortar tied contacts, the contacts recommended along with the implicit accuracy option, IACC on *CONTROL_ACCURACY, are supported for implicit-implicit transitions and non-groupable option. More specifically, the supported contact formulations are

*CONTACT_TIED_NODES_TO_SURFACE

*CONTACT_TIED_NODES_TO_SURFACE_OFFSET

*CONTACT_TIED_SHELL_EDGE_TO_SURFACE_BEAM_OFFSET

*CONTACT_TIED_SHELL_EDGE_TO_SURFACE_CONSTRAINED_OFFSET

9. **EXCLUDE.** This option is used to limit what data will be output to the LSDYNA dynain file. The input format is completely different and consists of any number of keyword cards WITHOUT the leading *. These cards and their associated data will not be output. For example:

```
*INTERFACE_SPRINGBACK_EXCLUDE
BOUNDARY_SPC_NODE
CONSTRAINED_ADAPTIVITY
```

would output all the normal dynain data, except for the SPC and adaptive constraints. The currently recognized keywords that can be excluded are:

BOUNDARY_SLIDING_PLANE

BOUNDARY_SPC_NODE

CONSTRAINED_ADAPTIVITY

DEFINE_COORDINATE_NODES

DEFINE_COORDINATE_VECTOR

ELEMENT_BEAM

ELEMENT_SHELL

ELEMENT_SOLID

INITIAL_STRAIN_SHELL

INITIAL_STRAIN_SOLID

INITIAL_STRESS_BEAM

INITIAL_STRESS_SHELL

INITIAL_STRESS_SOLID

INITIAL_TEMPERATURE_NODE

INITIAL_VELOCITY_NODE

NODE

REFERENCE_GEOMETRY

Remarks for Seamless Springback:

When seamless springback is invoked, the solution automatically and seamlessly switches from explicit or implicit dynamic to implicit static mode at the termination time and continues to run the static springback analysis. Seamless springback can be activated in the original LS-DYNA input deck or later using a small restart input deck. In this way, the user can decide to continue a previous analysis by restarting to add the implicit springback phase. (Another alternative approach to springback simulation is to use the keyword *INTERFACE_SPRINGBACK_LSDYNA to generate a dynain file after forming, and then perform a second simulation running LS-DYNA in fully implicit mode for springback. See Appendix P for a description of how to run an implicit analysis using LS-DYNA.)

The implicit springback phase begins when the forming simulation termination time ENDTIM is reached, as specified with the keyword *CONTROL_TERMINATION. Since the springback phase is static, its termination time can be chosen arbitrarily (unless material rate effects are included). The default choice is 2.0 × ENDTIM and can be changed using the *CONTROL_IMPLICIT_GENERAL keyword; see fields DT0 and NSBS.

Since the springback analysis is a static simulation, a minimum number of essential boundary conditions or Single Point Constraints (SPC's) can be input to prohibit rigid body motion of the part. These boundary conditions can be added for the springback phase using the input option on the *INTERFACE_SPRINGBACK_SEAMLESS keyword above.

If no boundary conditions are added with the SEAMLESS option, an eigenvalue computation is automatically performed using the Inertia Relief Option to find any rigid body modes and then automatically constrain them out of the springback simulation (see

*CONTROL_IMPLICIT_INERTIA_RELIEF). This approach introduces no artificial deformation and is recommended for many simulations.

An "SPR" option is available for several *CONTROL_IMPLICIT keywords to further control the implicit springback phase. Generally, default settings can be used, in which case the SPR option for *CONTROL_IMPLICIT keywords is not necessary.

To obtain accurate springback solutions, a nonlinear springback analysis must be performed. In many simulations, this iterative equilibrium search will converge without difficulty. If the springback simulation is particularly difficult, either due to nonlinear deformation, nonlinear material response, or numerical precision errors, a multi-step springback simulation will be automatically invoked. In this approach, the springback deformation is divided into several smaller, more manageable steps.

Two specialized features in LS-DYNA are used to perform multi-step springback analyses. The addition and gradual removal of artificial springs is performed by the artificial stabilization feature. Simultaneously, the automatic time step control is used to guide the solution to the termination time as quickly as possible, and to persistently retry steps where the equilibrium search has failed. By default, both of these features are active during a seamless springback simulation. However, the default method attempts to solve the springback problem in a single step. If this is successful, the solution will terminate normally. If the single step springback analysis fails to converge, the step size will be reduced, and artificial stabilization will become active. Defaults for these features can be changed using the following keywords:

- *CONTROL_IMPLICIT_GENERAL,
- *CONTROL_IMPLICIT_AUTO, and
- *CONTROL_IMPLICIT_STABILIZATION.

*INTERFACE_SSI_*{OPTION}*_ID

Purpose: This card creates a tied-contact soil-structure interface for use in a transient analysis of a soil-structure system subjected to earthquake excitation. This card allows the analysis to start from a static state of the structure, as well as to read in ground motions recorded on the interface in an earlier analysis.

Available options are:

<BLANK>

OFFSET

CONSTRAINED_OFFSET

LS-DYNA implements the effective seismic input method [Bielak and Christiano (1984)] for modeling the interaction of a nonlinear structure with a linear soil foundation subjected to earthquake excitation. Note that any nonlinear portion of the soil near the structure may be incorporated with the structure into a larger generalized structure, but the soil is assumed to behave linearly beyond a certain distance from the structure.

The effective seismic input method couples the dynamic scattered motion in the soil, which is the difference between the motion in the presence of the structure and the free-field motion in its absence, with the total motion of the structure. This replaces the distant earthquake source with equivalent effective forces adjacent to the soil-structure interface and allows truncation of the large soil domain using a non-reflecting boundary (e.g. *MAT_PML_ELASTIC) to avoid unnecessary computation. These effective forces can be computed using the free-field ground motion at the soil-structure interface, thus avoiding deconvolution of the free-field motion down to depth.

Nonlinear behavior of the structure may be modeled by first carrying out a static analysis of the soil-structure system, and then carrying out the transient analysis with only the structure initialized to its static state. Because the transient analysis employs the dynamic scattered motion in the soil, the soil cannot have any static loads only it − only the structure is subjected to static forces. Consequently, the structure must be supported by the static reactions at the soil-structure interface. Additionally, the soil nodes at the interface must be initialized to be compatible with the initial static displacement of the structure. LS-DYNA will do these automatically if the soil-structure interface is identified appropriately in the static analysis and reproduced in the transient analysis.

Thus, soil-structure interaction analysis under earthquake excitation may be carried out in LS-DYNA as follows:

1. Carry out a static analysis of the soil-structure system (e.g. using dynamic relaxation; see *CONTROL_DYNAMIC_RELAXATION), with the soil-structure interface identified using *INTERFACE_SSI_STATIC_ID

Optionally, carry out a free-field analysis to record free-field motions on the future soil-structure interface, using either *INTERFACE_SSI_AUX or *INTERFACE_SSI_AUX_EMBEDDED, for surface-supported or embedded structures respectively.

2. Carry out the transient analysis as a full-deck restart job (see *RESTART), with only the structure initialized to its static stress state (see *STRESS_INITIALIZATION), and the same soil-structure interface identified using *INTERFACE_-SSI_ID with the same ID as in static analysis:

   a) The structure mesh must be identical to the one used for static analysis.

   b) The soil mesh is expected to be different from the one used for static analysis, especially because non-reflecting boundary models may be used for transient analysis.

   c) The meshes for the structure and the soil need not match at the interface.

   d) Only the structure must be subjected to static loads, via *LOAD_BODY_-PARTS

   e) The earthquake ground motion is specified using *LOAD_SEISMIC_SSI:

      i) The NODE, SET, and POINT options allow specifying the ground motion at particular nodes, node sets, or coordinate points, respectively, on the soil-structure interface.

      ii) The AUX option employs motions recorded from a previous analysis using *INTERFACE_SSI_AUX or *INTERFACE_SSI_AUX_-EMBEDDED.

      iii) The DECONV option takes a motion specified at or below the top soild surface of the model and uses the SHAKE algorithm to deconvolve it to all points on the interface.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	HEADING						
Type	I	A70						

Card 2	1	2	3	4	5	6	7	8
Variable	STRID	SOILID	STRPR	SOILPR				
Type	I	I	I	I				
Default	none	none	0	0				

Optional card (superseded by LOAD_SEISMIC_SSI_AUX; see Remark 6)

Card 3	1	2	3	4	5	6	7	8
Variable	GMSET	SF	BIRTH	DEATH	MEMGM			
Type	I	F	F	F	I			
Default	none	1.	0.	10^{28}	2500000			

VARIABLE	DESCRIPTION
ID	Soil-structure interface ID. This is required and must be unique amongst all the contact interface IDs in the model.
HEADING	A descriptor for the given ID.
STRID	Segment set ID of base of structure at soil-structure interface.
SOILID	Segment set ID of soil at soil-structure interface.
STRPR	Include the structure side in the *DATABASE_NCFORC and the *DATABASE_BINARY_INTFOR interface force files: EQ.1: Structure side forces included. Note that for ncforc the forces for this side will be listed under surfa.
SOILPR	Include the soil side in the *DATABASE_NCFORC and the *DATABASE_BINARY_INTFOR interface force files: EQ.1: Soil side forces included. Note that for ncforc the forces for this side will be under surfb.
GMSET	Identifier for set of recorded motions from *INTERFACE_SSI_AUX or *INTERFACE_SSI_AUX_EMBEDDED

VARIABLE	DESCRIPTION
SF	Recorded motion scale factor.
BIRTH	Time at which specified recorded motion is activated.
DEATH	Time at which specified recorded motion is removed: EQ.0.0: Default set to 10^{28}
MEMGM	Size in words of buffer allocated to read in recorded motions

Remarks:

1. **Tied Contact Interface.** A tied contact interface (*CONTACT_TIED_SURFACE_TO_SURFACE) is created between the structure and the soil using the specified segment sets, with the soil segment set as the reference surface and the structure segment set as the tracked surface. Naturally, the two segment sets should not have merged nodes and can be non-matching in general. However, the area covered by the two surfaces should match.

2. **Keyword Options.** The options OFFSET and CONSTRAINED_OFFSET create the corresponding tied surface-to-surface contact interface.

3. **Interface ID.** The soil-structure interface ID is assigned as the ID of the generated contact interface.

4. **Soil Segment Set Orientation.** It is assumed that the soil segment set is oriented toward the structure.

5. **Multiple Soil-Structure Interfaces.** Multiple soil-structure interfaces are allowed, such as for bridge analysis.

6. **Obsolete Card.** Optional Card 3 has been superseded by *LOAD_SEISMIC_SSI_AUX, which allows for reading multiple ground motion files in the same model.

7. **Reading Recorded Motions.** The recorded motions are read in from a binary file named gmbin by default, but a different filename may be chosen using the option GMINP on the command line (see Getting Started, Execution Syntax).

8. **Recorded Motions.** If the motions from *INTERFACE_SSI_AUX or *INTERFACE_SSI_AUX_EMBEDDED were recorded on a segment set, then the free-field motions on each node in the soil segment set of the soil-structure interface are calculated from the nearest segment of the segment set used to record the motions. If, however, the motions were recorded on a node set, then the motions

on the soil segment set nodes are found by interpolation as is done for *LOAD_-SEISMIC_SSI.

INTERFACE_SSI_AUX_*{OPTION}

Available options are:

<BLANK>

NODE

Purpose: This card records the motion at a free surface, or on a set of nodes on a free surface, for the purpose of using the recorded motion as a free-field motion in a subsequent interaction analysis using *INTERFACE_SSI. By default, this card records motions on a segment set defining a surface, but can record motions on a node set using the option NODE. Only one of *INTERFACE_SSI_AUX and *INTERFACE_SSI_AUX_EMBEDDED is to be used for a particular soil-structure interface.

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

VARIABLE	DESCRIPTION
GMSET	Identifier for this set of recorded motions to be referred to in *INTERFACE_SSI. Must be unique.
SETID	Segment set or node set ID where motions are to be recorded.

Remarks:

1. **Output File.** The motions on the specified segment set or node set is recorded in a binary file named gmbin by default, but a different filename may be chosen using option GMOUT on the command line (see Getting Started, Execution Syntax).

2. **Output Interval.** The output interval for the motions may be specified using the parameter GMDT on the *CONTROL_OUTPUT card, with the default value being one tenth of the output interval for d3plot states.

*INTERFACE_SSI_AUX_EMBEDDED_*{OPTION1}*_*{OPTION2}*

Purpose: This card creates a tied-contact interface and records the motions and contact forces. The recorded data can then be used as free-field motion and reactions in a subsequent soil-structure interaction analysis using *INTERFACE_SSI, where the structure is embedded in the soil after part of the soil has been excavated. Only one of *INTERFACE_SSI_AUX and *INTERFACE_SSI_AUX_EMBEDDED is to be used for a particular soil-structure interface.

Available options for *OPTION1* are:

<BLANK>

OFFSET

CONSTRAINED_OFFSET

OPTION2 allows an optional ID to be given:

ID

ID Card. Additional card for ID keyword option.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	HEADING						
Type	I	A70						

Card 2	1	2	3	4	5	6	7	8
Variable	GMSET	STRID	SOILID	STRPR	SOILPR			
Type	I	I	I	I	I			
Default	none	none	none	0	0			

VARIABLE	DESCRIPTION
ID	Soil-structure interface ID. This is required and must be unique amongst all the contact interface IDs in the model.
HEADING	A descriptor for the given ID.

VARIABLE	DESCRIPTION
GMSET	Identifier for this set of recorded motions to be referred to in *INTERFACE_SSI. Must be unique.
STRID	Segment set ID at base of soil to be excavated
SOILID	Segment set ID at face of rest of the soil domain.
STRPR	Include the structure side in the *DATABASE_NCFORC and the *DATABASE_BINARY_INTFOR interface force files: EQ.1: Structure side forces included. Note that for ncforc the forces for this side will be listed under surfa.
SOILPR	Include the soil side in the *DATABASE_NCFORC and the *DATABASE_BINARY_INTFOR interface force files: EQ.1: Soil side forces included. Note that for ncforc the forces for this side will be listed under surfb.

Remarks:

1. **Output Binary File.** The motions on the specified segment set or node set is recorded in a binary file named gmbin by default, but a different filename may be chosen using the option GMOUT on the command line (see Getting Started, Execution Syntax).

2. **Output Interval.** The output interval for the motions may be specified using the parameter GMDT on the *CONTROL_OUTPUT card, with the default value being one tenth of the output interval for d3plot states.

*INTERFACE_SSI_STATIC_*{OPTION}*_ID

Purpose: This card creates a tied-contact soil-structure interface in order to record the static reactions at the base of the structure, which are to be used in a subsequent dynamic analysis of the soil-structure system subjected to earthquake excitation. This card is intended to be used with the initial static analysis of the structure subjected to gravity loads.

Available options are:

<BLANK>

OFFSET

CONSTRAINED_OFFSET

Card 1	1	2	3	4	5	6	7	8
Variable	ID	HEADING						
Type	I	A70						

Card 2	1	2	3	4	5	6	7	8
Variable	STRID	SOILID	STRPR	SOILPR				
Type	I	I	I	I				
Default	none	none	0	0				

VARIABLE	DESCRIPTION
ID	Soil-structure interface ID. This is required and must be unique amongst all the contact interface IDs in the model.
HEADING	A descriptor for the given ID.
STRID	Segment set ID of base of structure at soil-structure interface.
SOILID	Segment set ID of soil at soil-structure interface.
STRPR	Include the structure side in the *DATABASE_NCFORC and the *DATABASE_BINARY_INTFOR interface force files:

VARIABLE	DESCRIPTION
	EQ.1: Structure side forces included. Note that for ncforc the forces for this side will be listed under surfa.
SOILPR	Include the soil side in the *DATABASE_NCFORC and the *DATABASE_BINARY_INTFOR interface force files: EQ.1: Soil side forces included. Note that for ncforc the forces for this side will be listed under surfb.

Remarks:

See *INTERFACE_SSI_ID. The ID used for a particular interface in the static analysis must also be used for the same interface identified using *INTERFACE_SSI_ID during dynamic analysis.

*INTERFACE_THICKNESS_CHANGE_COMPENSATION

Purpose: Change the original tool gap between the upper and lower tools to variable gaps based on the formed blank thickness from a forming simulation. *This keyword must be used with *INCLUDE_COMPENSATION_CURRENT_TOOLS and *INCLUDE_COMPENSATION_BEFORE_SPRINGBACK.* This feature is useful for hot stamping simulations, where physical contacts are required between the upper/lower tools and the sheet blank to ensure a proper cooling rate. The rigid tools with the variable gaps will be output to the file rigid.new.

Compensation Card.

Card 1	1	2	3	4	5	6	7	8
Variable	IFLG	THK0						
Type	I	F						
Default	none	none						

VARIABLE	DESCRIPTION
IFLG	Activation flag; set to “1” to invoke the option.
THK0	Initial thickness of the sheet blank

Remarks:

1. **Overview.** This keyword causes LS-DYNA to modify the gap between tools due to the results of a forming simulation. *INCLUDE_COMPENSATION_CURRENT_TOOLS and *INCLUDE_COMPENSATION_BEFORE_SPRINGBACK are used to include the rigid tools and formed sheet blank files, respectively. The first file contains a set of upper and lower tools with the original gap. This gap is then modified to the formed blank thickness from a forming simulation provided in the second file, a dynain format file. The forming results from the dynain file must include the thicknesses of the formed sheet blank.

2. **Orientation of Surfaces.** For each tool, the surface normals must be consistent and must point towards the blank.

3. **Position.** The rigid tools and formed blank must be all in their punch home positions.

4. **Wrinkles.** To capture detailed tool gap change in case of wrinkles, the rigid body mesh size must be of the same order as the wrinkle wavelength.

Example:

A keyword example is provided below. The initial blank thickness is 1.2mm, the original tool is provided as rigid.inc, and the file drawn.dynain contains the formed blank with thickness information.

```
*KEYWORD
*INTERFACE_THICKNESS_CHANGE_COMPENSATION
$     IFLG      THK0
         1       1.2
*INCLUDE_COMPENSATION_CURRENT_TOOLS
rigid.inc
*INCLUDE_COMPENSATION_BEFORE_SPRINGBACK
drawn.dynain
```

Revision Information:

This feature was originally available in SMP (not available in MPP) Dev Revision 106357. Improvements are made in Dev Revision136166.

*INTERFACE_WELDLINE_DEVELOPMENT

Purpose: This keyword causes LS-DYNA to run a weld line development calculation instead of a finite element calculation. The input for this feature consists of (1) the formed blank from a completed metal forming simulation, (2) the corresponding initial blank, and (3) depending on which on desired whether the desired weld curve is on the formed or initial blank, the weld curve on the initial blank or on the final blank, respectively. Outputs include predicted final weld curve if input is the initial weld curve, or initial weld curve if input is the final welding curve; also nodes of any element edges that intersect the weld curve on the initial blank will be output to a file name affectednd_i.ibo and on the final blank to a file name affectednd_f.ibo.

Three additional keywords must be used together (and exclusively) with this keyword. They are: *INCLUDE_WD_INITIAL_BLANK, *INCLUDE_WD_FINAL_PART, and *INCLUDE_WD_WELDING_CURVE. See Remark 1 for a discussion of input order.

> **NOTE:** When this card is present, LS-DYNA *does not* proceed to the finite element simulation.

Development Parameter Card.

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

VARIABLE	DESCRIPTION
IOPTION	Welding curve development options: EQ.1: Calculate initial weld curve from final (given) weld curve, with output file name weldline.ibo, which will be on the initial blank mesh. EQ.-1: Calculate final weld curve from initial weld curve, with output file name weldline_f.ibo, which will be on the formed blank mesh.

Remarks:

1. **Input Order.** The input order of the keywords for a weld line development calculation is sensitive and must be in the order that follows:

```
*Keyword
*INTERFACE_WELDLINE_DEVELOPMENT
$  IOPTION
         1
*INCLUDE_WD_INITIAL_PART
$     filename
      initial.k
*INCLUDE_WD_FINAL_PART
$     filename
      final.k
*INCLUDE_WD_WELDING_CURVE
$     filename
      welding_curve.k
```

2. **Purpose of Weld Line Development Calculation.** For metal forming of tailor welded blanks, an initial straight weld line could become a curve on the formed part. The amount of deviation of the formed weld curve from its initial line depends on the part shape and forming conditions. Sometimes the formed weld curve is not desirable, thus a correction to the initial weld curve is needed. This keyword allows for determining the final weld curve given the initial weld curve and vice-versa, without performing another forming simulation.

3. **Mesh Adaptivity.** A mesh with adaptivity for the initial blank and final part is supported.

4. **Final (formed) Weld Curve.** The final formed weld curve should be projected onto the final blank mesh if it does not exactly lie on the mesh surface. This can be done with LS-PrePost4.2 via the menu option *GeoTol* → *Project* → *Project*, select *Closest Projection*, select *Project to Elements*, then define the destination mesh and source curves, and hit *Apply*. Sometimes the target curve may need enough points before projection; the points may be added via menu option *Curve* → *Spline* → *Method (Respace)* → *by number*. To write the curve out in *DEFINE_CURVE_TRIM_3D format, use *Curve* → *Convert* → *Method (To DEFINE_CURVE_TRIM)* → *To Key*, then write out the keyword using *FILE* → *Save Keyword*.

5. **Computed Weld Curve (and IGES).** Computed weld curves are written with a *DEFINE_CURVE_TRIM_3D keyword into a file called weldline.ibo (or, weldline_f.ibo, depending on IOPTION). The format of this file follows the keyword's specification. LS-PrePost4.0 can convert the computed curve to IGES; see the procedure in the manual pages for *INTERFACE_BLANKSIZE_DEVELOPMENT. After hitting Apply, the curves will show up in the graphics window, and *File* → *Save as* → *Save Geom as* can be used to write the curves out in IGES format.

Example:

As shown in Figure 30-41, given the initial (initialblank.k) and final blank (finalblank.k) configuration and a final formed weld curve (finalweldingcurve.k), the following input calculates a new initial weld curve on the initial blank. In this case, the final weld curve is specified as straight in the drawn panel.

```
*KEYWORD
*INTERFACE_WELDLINE_DEVELOPMENT
$OPTION
1
*INITIAL_BLANK
initialblank.k
*FINAL_PART
finalblank.k
*WELDING_CURVE
finalweldingcurve.k
*END
```

The output is the initial weld curve in the file weldline.ibo, Figure 30-41. Nodes of element edges that intersect the initial weld curve are output in affectednd_i.ibo; while nodes of element edges that intersect the final formed weld curve are output in affectednd_f.ibo, Figure 30-42.

To verify the predicted weld curve, the initial blank can be re-meshed according to the curve. A draw simulation can be performed again to confirm the final weld curve as straight, Figure 30-43.

Likewise, if given an initial weld curve (initialweldingcurve.k) and a final weld curve (weldline_f.ibo) can be calculated with the keyword inputs below:

```
*KEYWORD
*INTERFACE_WELDLINE_DEVELOPMENT
$OPTION
-1
*INITIAL_BLANK
initialblank.k
*FINAL_PART
finalblank.k
*WELDING_CURVE
initialweldingcurve.k
*END
```

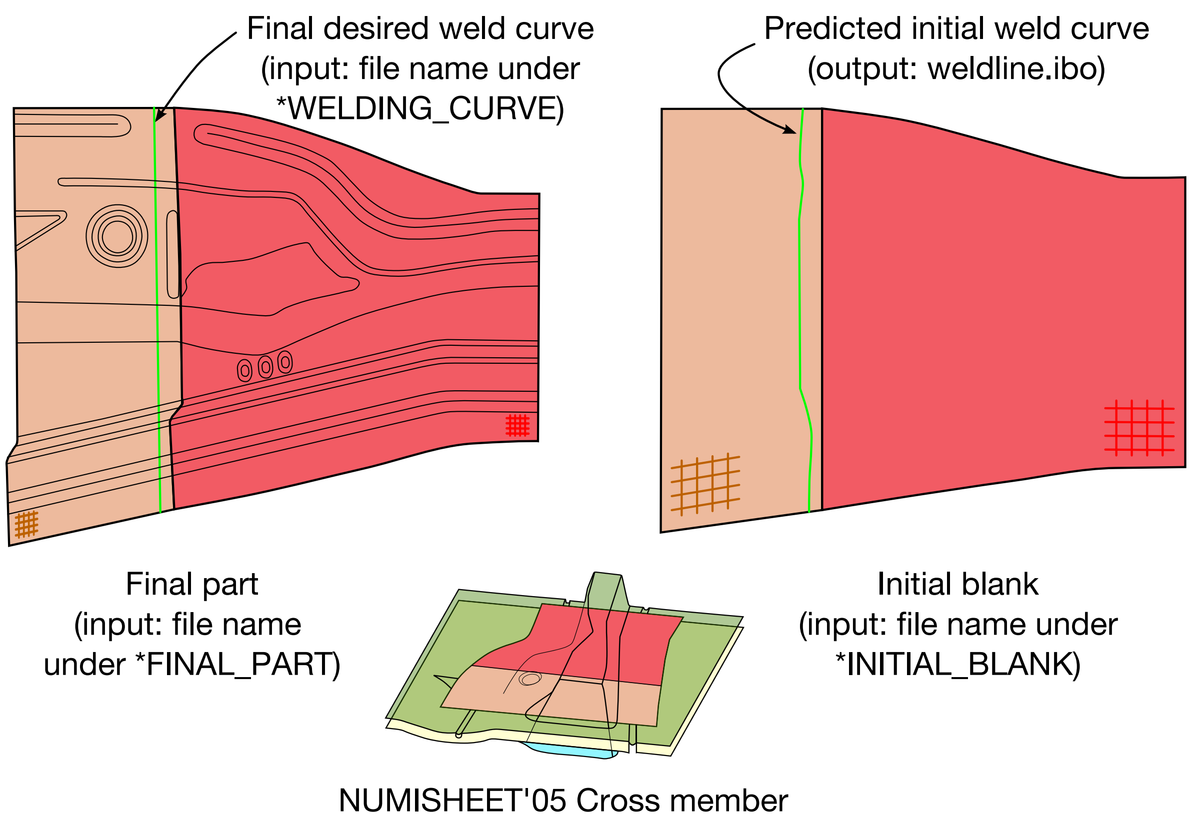


Figure 30-41. An example for a welding curve development.

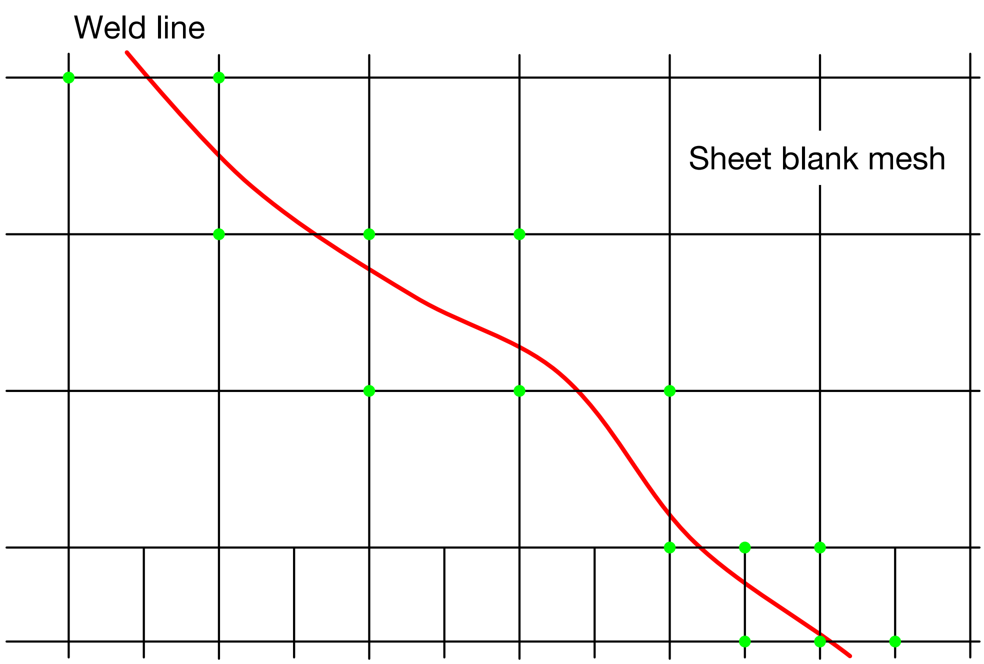


Figure 30-42. Nodes (in green) of element edges that intersect the weld curve are output in affectednd_i.ibo and affectednd_f.ibo, for initial and deformed mesh, respectively.

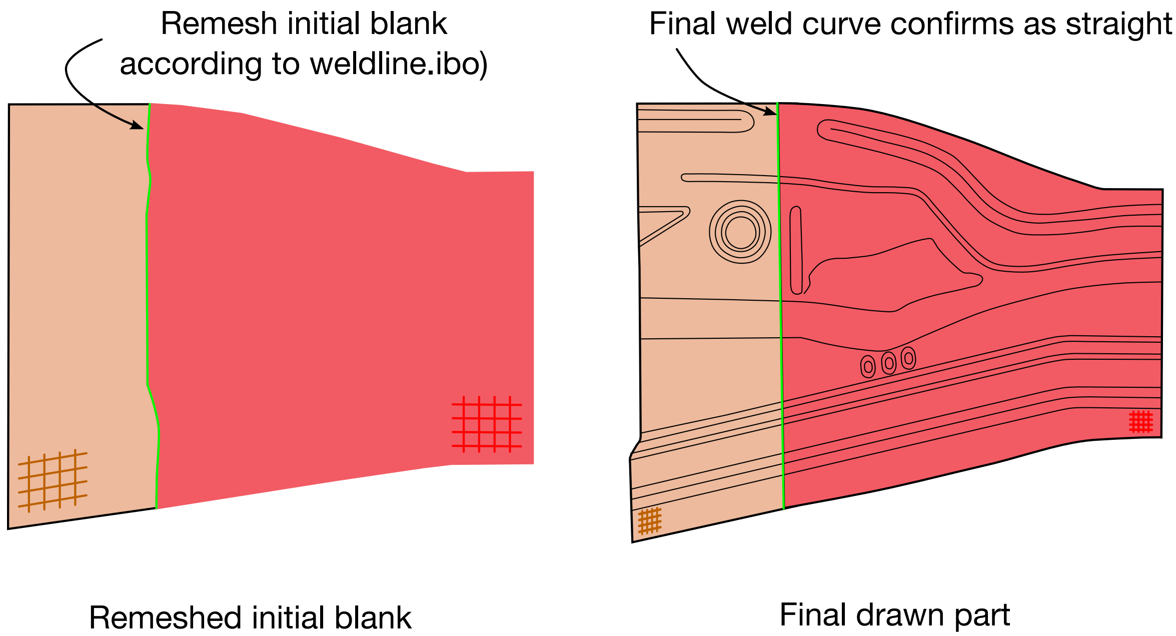


Figure 30-43. Verification run

Revision information:

1. IOPTION of “1”: Revision 105189 in both double precision versions of SMP and MPP.

2. IOPTION of “-1”: Revision 105727.

3. Output of nodes of element edges that intersect the weld curve: Revision 105727.

4. Later revisions may include improvements.