

**INPUT PARAMETERS FOR SPRINGBACK SIMULATION
USING LS-DYNA**

*Bradley N. Maker
Xinhai Zhu
Livermore Software Technology Corporation
June, 2001*

LS-DYNA has been applied to springback simulation by a large number of users, with generally mixed results. Some results have demonstrated 70% accuracy or better, while others have been entirely misleading. In order to eliminate inconsistent results, this report presents a standard procedure for conducting springback simulations with LS-DYNA. The “seamless” and “dynain” methods for springback are described, followed by a description of general implicit springback problem set-up. Recommendations are given for anticipating and improving springback prediction accuracy.

Wherever possible, LS-DYNA keyword input data is shown to clarify the presentation. Recommended input parameters are identified in boldface type and included in boxed keyword input syntax for quick reference. A boldface zero value is entered for required input data which is model specific, such as the termination time term.

The Forming Simulation

Results from the forming simulation provide the starting point for the springback simulation. The most important factor in springback accuracy is the accuracy of the forming simulation. This is essential! If trouble occurs during springback, look for the cause in the forming analysis.

In explicit forming simulations, run time can and should be greatly decreased using mass scaling and/or artificially high tool velocity. Both these methods introduce artificial dynamic effects, which must be minimized to reasonable levels in an engineering sense. A single independent parameter describing artificial dynamic effect is the number of explicit time steps (cycles) taken per millimeter of tool motion.

Relatively more cycles per millimeter are required when the forming process allows large unrestrained sheet motion. An example is the crash forming process, which uses no binders. Relatively fewer cycles per millimeter are necessary when the sheet is heavily constrained with binders and punch support. For most simulations, values of between 100 and 1000 cycles per millimeter produce reasonable results. If possible, or when it is otherwise necessary to repeat a simulation, use two different values and compare results to estimate sensitivity to artificial dynamic effects.

For an extensive description of input parameters for the forming simulation, see Maker and Zhu [1].

Springback Methods

LS-DYNA springback simulations can be performed by several methods. A standard explicit dynamic method may not be used since the objective is to obtain a static springback solution free from dynamic oscillations. Explicit dynamic relaxation is a viable method. The preferred approaches to springback employ the static implicit method. The two most common implicit approaches, the “seamless” and “dynain” methods, are described below.

Seamless Springback Method

In the seamless method, LS-DYNA begins by performing an explicit forming simulation. When the termination time is reached, LS-DYNA automatically and seamlessly switches to the implicit method, and continues with the springback simulation. At the time of switching, a user-specified list of parts (the sheet blank) are retained as active, and the remaining parts (the rigid tools) are deleted from the model. All contact interfaces are also automatically deleted. An optional list of nodal constraints are activated to eliminate rigid body motion after the tools are removed for the static springback simulation. (Required constraints are discussed later in this document.)

```
*INTERFACE_SPRINGBACK_SEAMLESS
$      psid
      0
$      nid          tcode          rcode
      0              0              0
      0              0              0
```

After switching seamlessly, LS-DYNA proceeds to perform a static implicit springback simulation. A special set of defaults are used which eliminate all requirements for *CONTROL_IMPLICIT keyword input. These alternate “springback” defaults are identified clearly in the user’s manual, and affect the time step size, artificial stabilization, and automatic time step control parameters. Default parameter values can be overridden by including optional *CONTROL_IMPLICIT keywords into the forming input deck.

Element Formulation Switching

An option is available to automatically switch shell element formulations when using the seamless springback method. When activated, all shell elements which are retained in the model during implicit springback are treated with the S/R Hughes-Liu element formulation #6. This option allows the user to reproduce previous results which were obtained using LS-DYNA3D for forming simulation and LS-NIKE3D for springback simulation, when default element formulations were chosen in both software. For best springback accuracy, however, it is recommended to use the Fast Shell element #16 for both the forming and springback simulations, so the element formulation switching option is not required.

The DYNAIN File Method

At the end of the forming simulation, LS-DYNA can output a keyword-formatted file named “dynain” containing the deformed mesh, stress, and strain state. The dynain file is requested using the keyword *INTERFACE_SPRINGBACK_DYNA3D. Input the id psid of a part set containing a list of parts to be included in the output file (usually just the sheet workpiece). An optional list of extra node constraints can be included, which are applied as the dynain file is written. These constraints provide a convenient way to eliminate rigid body motion in springback calculations. (Required constraints are discussed later in this document.)

```
*INTERFACE_SPRINGBACK_DYNA3D
$      psid
      0
$      nid          tcode          rcode
      0              0              0
      0              0              0
```

The dynain file can be used to perform many follow-on simulations such as springback, trimming, or additional forming. It can be included into a new LS-DYNA input deck so that each simulation is performed independently. This procedure avoids cumbersome binary restart databases, and separates multi-stage forming and springback jobs into more manageable pieces. For this reason, the dynain file method is the recommended method for springback simulation.

In the dynain file method, the input deck for springback simulation is easily constructed using the part, section, and material information from the original forming model, and the node, element, and initial stress and strain information from the dynain file. A few additional keywords must be added to control the implicit springback process.

Mesh Coarsening

Accurate forming simulation requires a very fine mesh over tool radii – typically at least four elements are needed around a ninety-degree radius. Surprisingly, much of this mesh refinement can be removed prior to springback analysis without significant loss of springback accuracy. Mesh coarsening is the procedure used in LS-DYNA to automatically combine neighboring elements in flat regions of the mesh.

Mesh coarsening can be applied to both uniform and adapted meshes. Mesh coarsening provides three significant benefits for implicit springback analysis: improved convergence behavior during nonlinear equilibrium iteration, due to reduced numerical truncation error; and reduced memory and cpu requirements due to the reduced model size.

The coarsening procedure is performed at the beginning of a simulation. Coarsening is applied to the input mesh, and then the simulation proceeds using the coarsened mesh. If a zero termination time is specified, and the keyword *INTERFACE_SPRINGBACK_DYNA3D is included, a new dynain file will be output containing the coarsened mesh and the simulation will terminate. (Be careful to rename the first dynain file to avoid overwriting it with the second dynain file.) This is the recommended procedure:

1. *forming simulation, output dynain file at termination time*
2. *mesh coarsening with zero termination time, output second dynain file*
3. *springback simulation using coarsened mesh*

Coarsening is activated using the keyword *CONTROL_COARSEN. The only required input parameter is the flatness tolerance **angle**, which limits coarsening to areas of the mesh where the angle between normal vectors of adjacent elements is less than the input value. A recommended value is **angle = 8** degrees, although values of up to 12 degrees have been used successfully. An optional list of nseed nodes are used to initialize the search for candidate groups of elements to be coarsened. Seed nodes can be used to assist the automatic searching logic in finding isolated regions of mesh within a part which need to be coarsened. Up to eight nodes may be defined. A seed nodes identifies the *center of a group of four elements* which may be combined into one. To avoid leaving a single row of fine elements around the perimeter in the coarsened mesh, seed nodes should not be chosen on mesh edges or refinement boundaries.

```
*CONTROL_COARSEN
$      icoarse      angle      nseed
      1            8.0         0
$      n1          n2          n3          n4          n5          n6          n7          n8
      0            0            0            0            0            0            0            0
```

An optional number of boxes may also be defined which protect regions of the mesh from coarsening, using the keyword `*DEFINE_BOX_COARSEN`. The parameter `iflag` indicates whether elements lying inside or outside the box will be protected.

```
*DEFINE_BOX_COARSEN
$   boxid   xmin   xmax   ymin   ymax   zmin   zmax   iflag
      0       0       0       0       0       0       0       0
```

Implicit Springback

Creating an input deck for implicit springback using a dynain file is simple. Keywords are required to activate the implicit method, and to select the time step size and the termination time. Extra constraints can be added using nodal SPCs to eliminate free rigid body motion of the sheet when the tools are removed. For difficult springback jobs, optional keywords are available to request multi-step springback unloading, to automatically adjust the time step size according to the difficulty of each step, and to control the linear and nonlinear equation solvers. A short template file can be used to save typical values for these keywords. Other necessary input, such as part, material, and section definitions, can be taken directly from the original forming input deck.

Activating The Implicit Method

Since springback is a static process, the implicit solver should be used. This solver is activated using the first parameter `imflag=1` on `*CONTROL_IMPLICIT_GENERAL`. The time step size is also input here using `dt0`, and can be chosen arbitrarily in most cases since the solution is static. A physically reasonable time step size should be chosen, so use `dt0=0.001` seconds:

```
*CONTROL_IMPLICIT_GENERAL
$   imflag   dt0   iefs  nstepsb   igso
      1      0.001   0       0         0
```

Choosing The Number Of Time Steps

The termination time and time step size determine the total number of springback steps. Springback of most reasonably stiff panels can be performed in a single step, so select the termination time `term=dt0` using `*CONTROL_TERMINATION`. Some difficult parts require several steps. A reasonable starting point for a difficult, multi-step analysis is four steps, or `term=4*dt0`.

```
*CONTROL_TERMINATION
$   term
      0.001
```

Required Constraints

All static simulations, including implicit springback analysis, require that rigid body motions be eliminated by defining constraints. These constraints are required since dynamic inertia effects are not included in a static analysis. Without constraints, a tiny applied load would cause the entire workpiece to move rigidly an infinite distance without creating any stresses. Mathematically, this means that without any constraints the global stiffness matrix for the model is singular, and the inverse can not be computed. When constraints are properly chosen this rigid body motion will be eliminated, and the model will deform freely without developing any reaction forces at the constraint points.

Constraints can be applied using the `*INTERFACE_SPRINGBACK` keywords, or, when the dynain file method is used, constraints can be added to the springback input deck using `*BOUNDARY_SPC_NODE`. Parameter `nid` indicates the constrained node ID, and a value of one is entered for each degree of freedom (`dx`, `dy`, `dz`) to be constrained:

```
*BOUNDARY_SPC_NODE
$   nid   cid   dx   dy   dz   rx   ry   rz
      0     0     0     0     0     0     0     0
```

Enough constraints must be defined to eliminate six rigid body motions in the model – three translations and three rotations. In theory, this could be accomplished by constraining all six degrees of freedom at a single shell element node point. In practice, numerical truncation error is introduced when rotational degrees of freedom are used to eliminate rigid body motion. The recommended method is therefore to constrain selected translational degrees of freedom at three nodes.

The three constraint nodes should be chosen well separated from each other, and away from edges and flexible areas in the part. The first node “A” receives constraints to all three translational degrees of freedom, and defines the reference point in the model where springback displacements are zero. The second node “B” is located away from node “A” along the global X-direction. Constraints are applied at node “B” to eliminate global Y- and Z-translation. The third node “C” is located away from node “A” along the global Y-direction. Only the global Z-translation is constrained at node “C”. Figure 1 shows a diagram of the location of these nodes on the model.

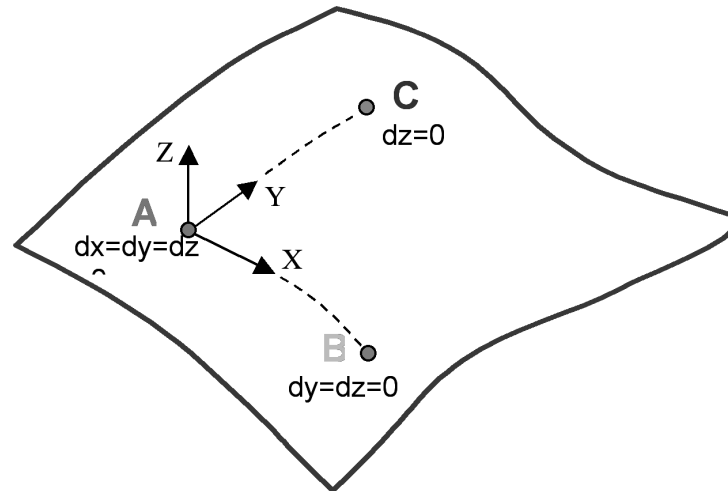


Figure 1 - Diagram showing location of constraint nodes on a typical springback model. Node “A” is the reference node, where all displacements are constrained. This eliminates the three translational rigid body motion of the part. Selected translational degrees of freedom are constrained at nodes “B” and “C” to eliminate the three rigid body rotations of the part about node “A”.

Constraints For Symmetric Models

Some stamping models include only one half of a symmetric panel, such as a hood or deck lid. In these cases, symmetry constraints are applied along one edge of the mesh. To eliminate rigid body motion during springback for these parts, constraints need only be added to two nodes chosen on the symmetry plane: completely constraining all translations for the first node, and eliminating one additional in-plane motion for the second node. Over-constraining a symmetric model by choosing three nodes according to figure 1 can lead to incorrect results. Figure 2 shows an example for the case of symmetry in the X-Z plane.

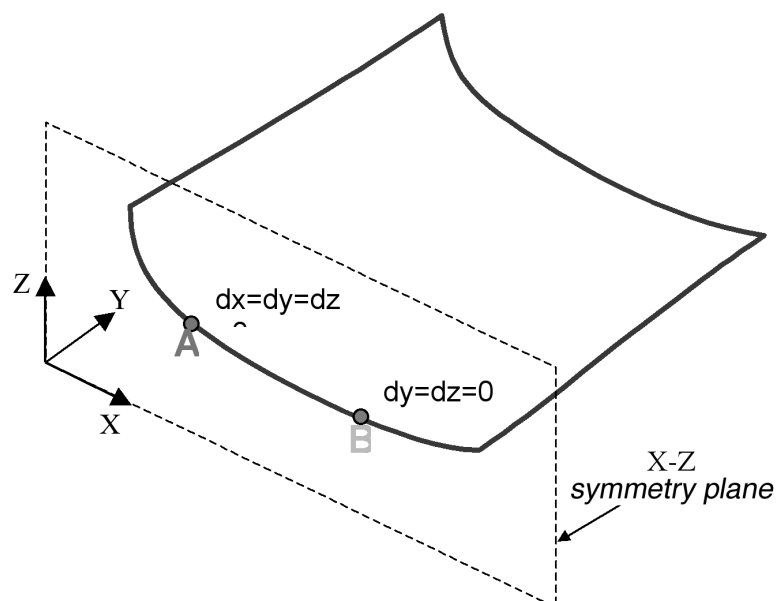


Figure 2 - Diagram showing location of constraint nodes for a symmetric part. The plane of symmetry in this example is the X-Z plane. Constraints must be added to *two nodes on the symmetry plane* to eliminate rigid body motion during static springback analysis. Node “A” is the reference node, where all displacements are constrained. This eliminates the three translational rigid body motion of the part. In addition to the standard symmetry constraints, selected translational degrees of freedom are constrained at node “B” to eliminate the three rigid body rotations of the part about node “A”.

Other LS-DYNA Input Parameters

The remaining necessary input parameters can be taken directly from the forming simulation input deck, and should not be modified for springback analysis. These include the *PART, *MAT_..., and *SECTION keywords which describe the workpiece. For recommended values of these parameters, see Maker and Zhu [1].

Running The Nonlinear Implicit Springback Simulation

Unlike explicit simulations where tiny time steps are completed very quickly, a large implicit simulation may take many minutes to complete a single time step. By default, LS-DYNA issues very little screen output information when running in implicit mode. Optional input parameters and interactive controls are available to produce more information about the progress of the simulation, as described below.

Equilibrium Iterations and Convergence

During each time step, the nonlinear solver searches iteratively to find static equilibrium. Activate the nonlinear solver print flag **nlprint=1** using the *CONTROL_IMPLICIT_SOLUTION keyword, or interactively type “<ctrl-c> nlprint” to see the progress of these iterations appear on the screen. The current displacement and energy norms are displayed each iteration, as shown in figure 3. These must both be decreased below their respective tolerances **dctol** and **ectol** before equilibrium is reached. The default values of these tolerances, 0.001 and 0.01 respectively, are generally good and need not be changed.

```
BEGIN implicit time step          3
=====
                time      =  1.09990E+00
    current step size  =  3.67821E-01

Iteration:   1      *|du|/|u| =  1.0894498E-01      *Ei/E0 =  1.8731172E+00

DIVERGENCE (increasing residual norm) detected:
  |{Fe} - {Fi}| ( 1.0547507E+07) exceeds |{Fe}| ( 9.1389570E+06)
automatically REFORMING stiffness matrix...

Iteration:   2      *|du|/|u| =  3.8969724E-03      *Ei/E0 =  3.3420090E-02
Iteration:   3      *|du|/|u| =  6.3582980E-03      *Ei/E0 =  3.3460971E-02
Iteration:   4      *|du|/|u| =  1.3780216E-03      *Ei/E0 =  6.2154527E-03
Iteration:   5      *|du|/|u| =  6.0081244E-03      *Ei/E0 =  7.7976128E-03
Iteration:   6      *|du|/|u| =  1.4377093E-03      *Ei/E0 =  8.9132953E-03
Iteration:   7      *|du|/|u| =  6.4089308E-03      *Ei/E0 =  1.7184228E-02
Iteration:   8      *|du|/|u| =  1.8267103E-03      *Ei/E0 =  1.9337881E-03
Iteration:   9      *|du|/|u| =  1.9491626E-03      *Ei/E0 =  2.3472405E-03
Iteration:  10      *|du|/|u| =  2.2147158E-03      *Ei/E0 =  1.5075735E-03
Iteration:  11      *|du|/|u| =  1.8921960E-03      *Ei/E0 =  1.9947323E-03
Iteration:  12      *|du|/|u| =  1.5758326E-03      *Ei/E0 =  7.9428701E-04

ITERATION LIMIT reached, automatically REFORMING stiffness matrix...

Iteration:  13      *|du|/|u| =  7.1106170E-04      *Ei/E0 =  3.0991789E-03

Equilibrium convergence summary for time step          3 at time =  1.0999005E+00
  Number of iterations to converge          =  13
  Number of stiffness reformations          =  2
```

Figure 3 – By selecting **nlprint=1** on *CONTROL_IMPLICIT_SOLUTION, or by interactively typing “<ctrl-c> nlprint”, the progress of the iterative equilibrium search will be displayed to the screen. Output is shown for a typical implicit step.

The equilibrium search is performed using a Newton-based method. By default, the “BFGS” method is used, where a new stiffness matrix is formed after every 11 iterations. For difficult springback problems (flexible parts with large springback deformation) the “Full Newton” method is better, since this method forms a new stiffness matrix after every iteration. To activate the Full Newton method, set the iteration limit between stiffness reformations to **ilimit=1**, and increase the maximum allowable stiffness reformations per time step to **maxref=100**. In some cases, the full Newton method will perform better if the line search is disabled using **lstol=99999**.

```
*CONTROL_IMPLICIT_SOLUTION
$  nlsolvr  ilimit  maxref  dctol  ectol  rctol  lstol
      0      1      100    0.0    0.0     0     99999.
$  dnorm  divflag  inistif  nlprint
      0      0      0      1
```

Solving The Linear System $[K]\{x\}=\{f\}$

The stiffness matrix formed during implicit analysis requires a large amount of memory, and computing its inverse requires most of the CPU time. These operations are performed by the linear equation solver, whose control parameters are found on *CONTROL_IMPLICIT_SOLVER. The default linear solver **lsolvr=0** is generally recommended. A double precision solution to the linear system $[K]\{x\}=\{f\}$ can be selected using **lsolvr=6**, however this alone does not often improve results, and does increase memory requirements by 2x. Solver #6 is very efficient at utilizing scratch files on disk to run in “out-of-core” mode, so it is recommended when computer memory resources are limited.

```
*CONTROL_IMPLICIT_SOLVER
$  lsolvr  lprint  negeig
      0      0      0
```

A summary of memory and CPU usage is printed to the screen when the **lprint** flag is activated, either by input using **lprint=1**, or interactively by typing “<ctrl-c> lprint”. The interactive control can be issued a second time to stop printing the memory information. Memory limits can be increased using the execution line argument “**memory=**”, where the default is **memory=8500000**. Note that 1 Mword = 4 Mbytes in single precision, and 1 Mword = 8 Mbytes in double precision.

```
SPARSE LINEAR EQUATION SOLVER STORAGE data(Mwords)
(225972 degrees of freedom)
pointer arrays:  initial = 11.523
                  actual = 6.413
stiffness coefficients = 6.187
Factorization Workspace (estimated)
                  symbolic = 14.015
                  numeric = 18.335
Final Storage Requirements (10% for pivoting)
                  incore          out-of-core
symbolic factorization = 5.276          5.276
numeric factorization = 69.772         5.292
numeric solution = 65.561          3.145
TOTAL = 87.648          23.168
TOTAL available = 98.196          98.196
```

an INCORE solution will be performed

```
Initialization CPU = 7.220E+00 seconds
Symbolic Factorization CPU = 1.065E+01 seconds
Numeric Factorization CPU = 8.539E+02 seconds
Forward/Backward CPU = 5.060E+00 seconds
```

Figure 4 – By selecting **lprint=1** on *CONTROL_IMPLICIT_SOLVER, or by interactively typing “<ctrl-c> lprint”, the memory and CPU requirements for the linear equation solver will be displayed to the screen. Output is shown for a production size springback model. This job will run in core memory since the total memory available (98.196 Mwords) is larger than the total required for incore solution (87.648 Mwords). The option “**memory=100m**” was used on the command line to request 100,000,000 words of memory.

Difficult Springback Simulations

The following section offers suggestions for solving difficult springback simulations. These typically involve very flexible parts, on which experimental springback measurements are also often difficult. In simulation, these parts usually present convergence trouble for the nonlinear equilibrium iteration process. A method is presented below for using several steps to simulate springback unloading, followed by a troubleshooting checklist with other modeling suggestions.

Multi-Step Springback for Difficult Parts

The applied load in a springback simulation results from the initial stress in the sheet, which is no longer in equilibrium once the tools have been removed. For difficult springback problems, this “load” must be applied slowly over several steps in order to divide the nonlinear springback response into manageable pieces. *Artificial stabilization* is the method used in LS-DYNA to distribute springback response over several steps. In this method, springs are artificially introduced to the model which restrict the motion of the sheet nodes. As the solution proceeds the spring stiffnesses are reduced, allowing more springback. When the termination time is reached the springs are completely removed, allowing completely unrestrained springback. It is important to reach the termination time completely, otherwise some artificial stabilization will remain in the model and the results will not be accurate.

To use multiple steps in a springback solution, the termination time must be extended. A good starting point for difficult jobs is four steps, so if the step size on `*CONTROL_IMPLICIT_GENERAL` is `dt0=0.001` then the termination time on `*CONTROL_TERMINATION` should be `term=0.004`.

Artificial stabilization is activated using `ias=1` on `*CONTROL_IMPLICIT_STABILIZATION`. When active, a message is printed to the screen at the start of each time step showing how much stabilization remains in the model. At the termination time, the message reports that artificial stabilization has been “completely removed”.

```
*CONTROL_IMPLICIT_STABILIZATION
$      ias      scale  tstart      tend
          1      0.001      0          0
```

The initial stiffness of these springs can be scaled using the input parameter `scale`. This parameter must be chosen using some engineering judgement about the flexibility of the panel being studied. Table 1 gives some guidelines on choosing `scale`.

type of panel	example application	scale
stiff, heavy gage	frame crossmember	1.000 (default)
stiff, standard gage	reinforced inner panel	0.100
flexible, curved	fender outer panel	0.010
flexible, flat	hood outer panel	0.001

Table 1 – When artificial stabilization is used for multi-step springback, the stabilization stiffness scale factor must be chosen according to the panel type. Note that stiff panels generally do not require multi-step springback, so the default value `scale=1.000` must nearly always be reduced.

A small value for `scale` gives softer springs, allowing more springback in the first few steps of the simulation. If convergence of the first step is difficult, use a larger value for `scale`. If the first few steps converge in very few iterations but the last step is difficult, use a smaller value for `scale`.

If convergence trouble is encountered during the iteration process, automatic time step control is available to repeat a failed step using a smaller step size. Automatic time step control is activated using `iauto=1` on the `*CONTROL_IMPLICIT_AUTO` keyword. For difficult springback simulations, an aggressive time step control strategy can be used. Increase the optimum number of iterations using `iteopt=200`, and restrict the maximum time step size using `dtmax=0.001`. In this way, the stepsize will always be increased after successfully converging, until the maximum stepsize is reached.

```
*CONTROL_IMPLICIT_AUTO
$      iauto  iteopt  itewin  dtmin  dtmax
          1      200      0      0.0      0.001
```


Troubleshooting Checklist

The following sections offer suggestions for common springback problems.

Poor Accuracy

Most accuracy problems result from errors which were introduced during the forming simulation. By closely examining the forming model and results, it may be possible to identify problems and anticipate poor springback predictions before they are submitted. Follow the guidelines described in Maker and Zhu [1]. In particular, look for:

- Insufficient mesh refinement. At least four elements are needed around ninety-degree tool radii.
- Poor element aspect ratio. Use elements which are as nearly square as possible.
- Artificial explicit dynamic effects. Running the forming simulation too slowly or too quickly can introduce error. Check the number of cycles taken per millimeter of tool motion.
- Changes in element formulation. For best accuracy, the more expensive element #16 must be used in the forming simulation as well as during springback, even though this adds significant cost to the forming simulation.
- Changes in thickness integration points. The number of thickness integration points must never be changed between forming and springback simulation.

Incorrect or Insufficient Material Data.

The effective stress – effective *plastic* strain curve must be carefully checked:

- The first data point must be at zero effective plastic strain and yield stress (0.0 , sy).
- Stress and strain must increase monotonically.
- Slope of each segment must vary smoothly.
- Data must fully include the range of strain seen in the part, including very large strains seen at the outer surface in sharp corners. Do not rely on LS-DYNA to extend your curve.
- Avoid too many data points. Rely on at most four significant digits.

Incomplete Solution

Beware that if convergence fails, LS-DYNA will issue an *error termination message*, and a d3plot state will be generated containing the last trial equilibrium geometry. These results are not accurate! Similarly, if the final step of a multi-step solution is not completed successfully, artificial stabilization will not be completely removed, an *error termination message* will be written, and the d3plot results will not be accurate. Accurate results can only be obtained after a *normal termination*.

Incorrect Constraints

The model must be adequately constrained to remove rigid body motion, but should not be over-constrained. Review the above section “Required Constraints”.

Gravitational Effects

The shape of large, flexible panels can be affected by gravity. Gravity effects can be easily included in springback simulations using *LOAD_BODY and *DEFINE_CURVE keywords. Be careful to employ a consistent system of units when defining gravitational acceleration.

Loose Convergence Tolerance

Nonlinear convergence tolerances can be increased to allow premature convergence, leading to poor accuracy. The default tolerance values for dctl and ectol on *CONTROL_IMPLICIT_SOLUTION are generally adequate, and should not be increased. Decreasing these tolerances to enforce equilibrium more strictly can be beneficial, especially when a double precision executable is used.

Single vs. Double Precision

Use of a double precision version of LS-DYNA improves convergence behavior in many implicit simulations. Merely activating a double precision linear equation solver has marginal benefit in an otherwise single precision executable. Contact LSTC to see if a double precision executable is available for your computer platform.

Mesh Coarsening

Mesh coarsening should be applied to most production size jobs to combine small elements into larger ones, reducing cpu time, memory requirement, and numerical truncation errors. If the number of elements in the formed workpiece exceeds 50,000 consider using mesh coarsening.

Extrusion and Coining

Significant errors result from situations where the workpiece is pinched between upper and lower tools to the extent that it is extruded or coined. This does not include the normal action of binders. Accurate simulation of extruded and coined parts may require layers of 8-node solid elements and advanced friction models which are beyond the scope of standard stamping simulation.

Clearance and Home Gap

Many springback simulations (and experiments!) are very sensitive to sidewall clearance in tools, and the “home gap” left at the bottom of the punch stroke. Carefully verify that your model accurately represents these details by making measurements directly on the model using the post-processor. Errors of less than one millimeter can have substantial effects on accuracy.

References

Maker, B.N. and Zhu, X, “Input Parameters for Metal Forming Simulation Using LS-DYNA,” April, 2000, available at www.feainformation.com/forming_parameters2.pdf or by email from support@lstc.com.