Input Parameters for Metal Forming Simulation using LS-DYNA

Bradley N. Maker

Xinhai Zhu

Livermore Software Technology Corporation

April, 2000

LS-DYNA has been widely used to study automotive crash. Default input parameters are generally chosen to give efficient, accurate crash simulation results. These defaults are not necessarily optimal for metal forming simulations. The following presents a standard procedure for conducting metal forming simulations with LS-DYNA. 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**.

General Problem Setup

In explicit forming simulations, run time can 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.

More cycles per millimeter are required when the forming process allows large unrestrained sheet motion, such as in crash forming. When the sheet is heavily constrained with binders and punch support, fewer cycles per millimeter are necessary. 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.

It is recommended to select a maximum tool velocity of 2.0 mm/ms, and start and end the simulation with zero velocity. A simple trapezoidal velocity profile can be used, as shown in figure 1. Use the mass scaling time step size parameter **dt2ms** to achieve the required number of cycles per millimeter, according to the simple formula:

Time step size = 1.0 / (max tool velocity * cycles per millimeter)

Tool velocity, time step size, and termination time must be selected in a coordinated manner. Given the total tool travel, the procedure below can be used to set up the simulation:

Given:	total tool travel (mm): **D**	
	maximum tool velocity (mm/ms): 2.0	
	velocity profile:	trapezoidal with 2.0 ms rise and fall times (figure 1)
Choose:	number of cycles per millimeter: **ncpm**	
Compute:	termination time (ms):	**T** = 2.0 + **D**/2
	velocity data points:	(0.0, 0.0) (2.0, 2.0) (**T** – 2.0, 2.0) (**T**, 0.0)
	time step size (ms):	**dt2ms** = 1/(2***ncpm**)

The above calculation provides velocity profile data points for use in the tool motion section below. The termination time **term** and mass scaling step size **dt2ms** are input using the following keywords:

```
*CONTROL_TERMINATION
$     term
       0.0
$
*CONTROL_TIMESTEP
$   dtinit    tssfac      isdo     tslmt     dt2ms
       0.0       0.0         0       0.0       0.0
```

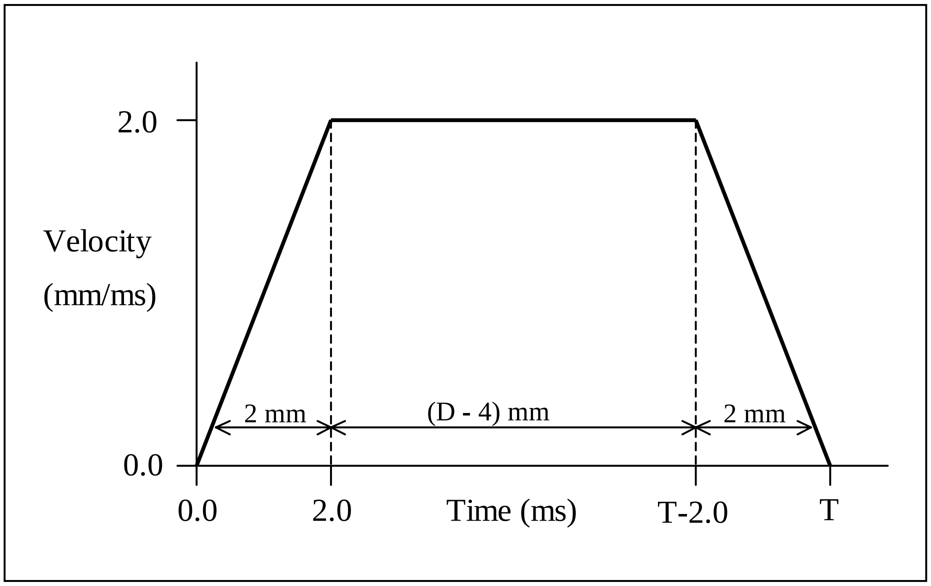

Figure 1: A trapezoidal punch velocity profile is shown with maximum velocity of 2.0 mm/ms. The rise and fall times are each 2.0 ms. Punch displacement is shown for each section of the curve. Total punch displacement = D mm.

Shell Element

The shell element formulation is selected using the **elform** parameter on the *SECTION_SHELL keyword. For formability analysis only, the default shell type #2 is recommended. If springback analysis will be performed after forming, the more expensive shell type #16 is recommended for both forming and springback simulations. The extra cost of the forming simulation will produce better accuracy and less convergence trouble during springback.

The number of thickness integration points **nip** is also selected using this keyword. Three points may be adequate for formability analysis, but if a springback analysis is to be performed at least 5 points must be used, and 7 points are better. The CPU cost of the forming analysis increases linearly with the number of thickness integration points.

Other required parameters on this keyword include the section id **secid** and part thickness **t1-t4**.

```
*SECTION_SHELL
$    secid    elform      shrf       nip     propt   qr/irid     icomp     setyp
         0        16       0.0         5       0.0       0.0         0         0
$       t1        t2        t3        t4      nloc
       0.0       0.0       0.0       0.0       0.0
```

The *CONTROL_SHELL keyword is used to select three important parameters. The use of triangle elements should be minimized. If some triangles are necessary, LS-DYNA will automatically apply the more accurate C0 triangle shell formulation when the **itrist** flag is selected. This element is more accurate than the degenerated quad element formulations, which are typically too stiff.

Activate shell element thickness change using the **istupd** flag. Otherwise elements will remain constant thickness. If element formulation #2 is used, activate the warping stiffness using the **bwc** parameter.

```
*CONTROL_SHELL
$   wrpang    itrist     irnxx    istupd    theory       bwc     miter      proj
       0.0         1         0         1         0         1         0         0
```

A stiffness form of hourglass control is recommended for metal forming. Select **ihq** as either type 4 or 5. Use the default value for hourglass coefficient.

```
*CONTROL_HOURGLASS
$      ihq    I   qh
         4       0.0
```

Materials

It is recommended to use the same material model for forming and springback simulation, even though springback is nominally an elastic unloading process. Several material models are currently available for implicit springback, including types 1, 3, 18, 24, 36, and 37. Of these, type 37 is most commonly used. Input the true physical density, and use mass and velocity scaling to artificially speed up simulation time, as described above. Avoid using material failure options with implicit springback, since these can create convergence problems.

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 , σy)**.
- ❑ Stress and strain must **increase monotonically**.
- ❑ **Slope** of each segment must vary smoothly.
- ❑ Avoid **too many data points**. Rely on at most four significant digits.

All tools should use shell elements and the rigid material type 20. Input reasonable elastic material properties for **ro**, **e**, and **pr**, such as those used for the sheet. Apply displacement constraints using **con1** and **con2** for translation and rotation, respectively. The third line is required but may be left blank.

```
*MAT_RIGID
$      mid        ro         e        pr
         0       0.0       0.0       0.0
$      cmo      con1      con2
       1.0         0         0
$      lco
         0
```

Contact

Penalty-based contact interfaces are recommended for most metal forming simulations. Set the penalty stiffness scale factor for all interfaces to 0.010 using **slsfac** on the *CONTROL_CONTACT keyword. Check results for significant penetrations around features with small radii or high contact pressure, and increase the penalty scale factor by up to 10x if necessary. This can be done separately for each interface using parameters in the *CONTACT keywords.

Activate thickness offsets for the sheet using **shlthk**. It is recommended to build tool mesh to the actual tool surfaces, so neglect thickness offsets in the rigid tools. See below for incorporating tool offsets into the contact interfaces when necessary.

```
*CONTROL_CONTACT
$   slsfac    rwpnal    islchk    shlthk    penopt    thkchg     orien
      0.01       0.0         0         1         0         0         0
$   usrstr    usrfac     nsbcs    interm    xpenen
         0         0         0         0       0.0
```

The “forming-one-way” type contact interface is recommended. This interface computes tooling curvature information for use in look-forward adaptive remeshing. This interface also allows disjoint mesh in the (master) tool surface. Tools must be meshed using 3- or 4-node shell elements. The sheet must be the slave side.

Surface definition by part ID is recommended (**sstyp**=**mstyp=3**), and must be used for the slave side if adaptive mesh is activated. For the forming simulation, 20 percent of critical viscous damping (**vdc**) is recommended to help eliminate high frequency dynamics. The shooting node logic must be disabled using the **snlog** flag on optional card B.

The thickness of the (slave) sheet is included in contact calculations. Contact occurs on an imaginary surface projected one-half thickness normal to each segment. It is recommended to build tool meshes which incorporate physical offsets, or die gaps. If this is not possible, the tool offset can be created by using a negative master thickness **mst**. If the master segment thickness is negative, a tool offset of one-half thickness is created. Large negative offsets will give poor results near small tool radii.

An alternative contact enforcement using the constraint method (not penalty method) can be selected using **soft=4** on optional card A. This is not generally necessary nor recommended.

Other required input parameters include the slave and master part IDs (**ssid**, **msid**), and the friction coefficient **fs**.

```
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE
$     ssid      msid     sstyp     mstyp    sboxid    mboxid       spr       mpr
         0         0         3         3         0         0         0         0
$       fs        fd        dc         v       vdc    penchk        bt        dt
       0.0       0.0       0.0       0.0      20.0         0       0.0       0.0
$      sfs       sfm       sst       mst      sfst      sfmt       fsf       vsf
       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
$ optional card A
$     soft
         0
$ optional card B
$   penmax    thkopt    shlthk     snlog
       0.0         0         0         1
```

Tool Motion

Tool motion is imposed using the *BOUNDARY_PRESCRIBED_MOTION_RIGID keyword. Indicate the tool part ID **pid** and the direction of motion **dir**. Velocity or displacement may be prescribed. Displacement control (**vad=2**) is recommended for implicit forming simulations where large time steps are used since displacement control produces the most accurate total tool travel. A curve giving motion vs. time is identified using **lcid**.

```
*BOUNDARY_PRESCRIBED_MOTION_RIGID
$      pid       dir       vad      lcid     scale
         0         0         2         0       0.0
```

Tool motion (displacement or velocity) should vary smoothly with time, starting and ending with a velocity of zero. Define this motion using *DEFINE_CURVE or *DEFINE_CURVE_SMOOTH. It is not necessary to include a zero velocity "hold time" at the end of an explicit forming simulation before proceeding to springback.

```
*DEFINE_CURVE
$       id
         0
$     time    motion
       0.0       0.0
       0.0       0.0
```

Binder Loads

Forces on binders and other floating tooling can be applied using either *LOAD_NODE or *LOAD_RIGID_BODY. If the node option is used, any tool node can be selected as long as rotational motion of the rigid tool is constrained. If the rigid body option is used, load is applied at the center of mass of the tool:

```
*LOAD_RIGID_BODY
$      pid       dir      lcid     scale
         0         0         0       0.0
```

Load variation with time is input using *DEFINE_CURVE. Load should begin from zero and increase to constant or starting values over the first few milliseconds of the simulation. Monitor the interface reaction forces to verify applied loads are free from dynamic effects.

```
*DEFINE_CURVE
$       id
         0
$     time      load
       0.0       0.0
       0.0       0.0
```

Adaptive Mesh Refinement

Adaptive mesh is recommended to automatically subdivide elements and create refined mesh in areas of high curvature. Begin with a regular mesh of rectangular elements with aspect ratios as near to one as possible. Avoid triangles, especially in critical areas. Select initial element size small enough to accurately resolve part details after three or four levels of refinement. For accurate springback, especially flange and wall opening in channels, at least 3-4 elements are needed around critical radii.

Use the adaptive mesh flag **adpopt** in the *PART keyword to indicate which parts are candidates for adapted mesh:

```
*PART
workpiece
$      pid       sid       mid     eosid      hgid      grav    adpopt      tmid
         0         0         0         0         0         0         1         0
```

The forming simulation is interrupted at the adaptive mesh frequency **adpfreq**, and the mesh is evaluated for adaptive refinement. Each element may be subdivided at most one level during each adaptive cycle. The adaptive frequency must be chosen small enough to prevent large elements from drawing over radii before being refined. Adapting the mesh every 2-4 mm of tool motion is usually adequate.

Two criteria are used to control element refinement: the angle between two neighboring elements, and the curvature of the approaching tooling. The adaptive tolerance **adptol** selects allowable angle change. Values near 4.0 degrees are reasonable. Smaller values will produce more refined mesh. The method used to measure this angle is selected using **adpopt**. Method 2 is recommended.

The maximum refinement level is given by **maxlvl**. Values of 3-4 are recommended, depending on the initial element size. The entire mesh can be refined one or more levels at the start of the simulation using the **ireflg** parameter.

After adaptively refining the mesh, the simulation can optionally be restarted from an earlier time to repeat a portion of the forming using the new mesh. This two-pass method is no longer recommended, since the new look-forward refinement criterion can initiate refinement before tool contact. The recommended method is therefore one-pass (**adpass=1**) with look-forward active (**adpene > 0.0**). The **adpene** parameter indicates the minimum allowable separation distance before tooling curvature is evaluated. This distance is also subject to the proximity limits used in contact detection, so values larger than several sheet thicknesses will have no additional effect.

```
*CONTROL_ADAPTIVE
$  adpfreq    adptol    adpopt    maxlvl    tbirth    tdeath     lcadp    ioflag
       0.0       4.0         2         3       0.0       0.0         0         0
$  adpsize    adpass    ireflg    adpene
       0.0         1         0       5.0
```

Adaptive refinement **level** may also be modified in certain regions of the mesh using an adaptive box definition:

```
*DEFINE_BOX_ADAPTIVE
$       id       xmn       xmx       ymn       ymx       zmn       zmx
         1       0.0       0.0       0.0       0.0       0.0       0.0
$      pid     level
         0         0
```

Drawbeads

Drawbeads can be included directly in the binder surfaces, or approximated by adding analytic drawbeads to flat binders. True drawbeads are sometimes necessary for accurate springback prediction on untrimmed panels. However, analytic drawbeads are most commonly used, with springback simulation performed only on the trimmed panels.

Analytic drawbeads are defined by a set of node points on the binder surface. Straight beads require only two nodes, one at each end of the bead. More nodes are necessary to accurately define curved beads. Additional points are automatically generated internally within LS-DYNA, interpolating between user-defined nodes, and drawbead forces are applied to the sheet as it passes these points.

Define a node set for the drawbead nodes. The set is used to attach the nodes to the binder, as well as in the drawbead definition. To attach the nodes to the binder, define them as extra nodes for the rigid body using keyword *CONSTRAINED_EXTRA_NODES_SET. Input the binder part id **pid** and the node set id **sid**.

```
*SET_NODE_LIST
$      sid
         0
$       n1        n2        n3        n4        n5        n6        n7        n8
         0         0         0         0         0         0         0         0
```

```
*CONSTRAINED_EXTRA_NODES_SET
$      pid      nsid
         0         0
```

The drawbead contact interface definition is similar to the standard forming type input, with the addition of card #4. The drawbead node set id is entered for **ssid**, and **sstyp** is chosen as 4. The sheet is defined as the master side of the interface.

Drawbead forces are generated in three ways. Frictional force is created if the friction coefficient **fs** is nonzero. Normal direction force follows the force-displacement curve given by **lcidnf**, in units of force per unit drawbead length. This normal direction force represents the force required to form the drawbead as the binder is closed. The drawbead restraining force due to bending and unbending of the sheet is given by curve **lcidrf**. This curve defines the maximum restraining force (per unit drawbead length) as a function of drawbead depth. As the binder closes, the drawbead depth increases to its maximum value **dbdth**. At time=0, binder separation should be at least **dbdth** to avoid initial penetration of the drawbead contact.

```
*CONTACT_DRAWBEAD
$     ssid      msid     sstyp     mstyp    sboxid    mboxid       spr       mpr
         0         0         4         3         0         0         0         0
$       fs        fd        dc         v       vdc    penchk        bt        dt
       0.0       0.0       0.0       0.0       0.0         0       0.0       0.0
$      sfs       sfm       sst       mst      sfst      sfmt       fsf       vsf
       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
$   lcidrf    lcidnf     dbdth     dfscl    numint
         0         0       0.0       0.0         0
```

A drawbead box should also be defined using **mboxid**. This box is automatically sized to contain all sheet nodes which may at some time enter the drawbead. Defining the box can reduce cpu time. Enter the sheet part ID **pid**, the drawbead node set id **nsid**, and the nominal direction of tool movement **idir**.

```
*DEFINE_BOX_DRAWBEAD
$    boxid       pid      nsid      idir
         0         0         0         0
```

Output

Many types of output can be generated by LS-DYNA. Of particular interest for metal forming applications are the binary plot databases including finite strain data, which are used in forming limit diagram analysis. Include strain in the plot database using the *DATABASE_EXTENT_BINARY flag **strflg**.

```
*DATABASE_EXTENT_BINARY
$    neiph     neips    maxint    strflg    sigflg    epsflg    rltflg    engflg
         0         0         0         1         0         0         0         0
$   cmpflg    ieverp    beamip     dcomp      shge     stssz    n3thdt
         0         0         0         0         0         0         0
```

Indicate the output frequency **dt** for binary plot and other ascii databases using the *DATABASE keywords. The RCFORC and BNDOUT databases are often requested to obtain reaction forces across interfaces and due to prescribed motion. See the user’s manual for an extensive list of available output databases.

```
*DATABASE_BINARY_D3PLOT
$       dt
       0.0
*DATABASE_RCFORC
$       dt
       0.0
*DATABASE_BNDOUT
$       dt
       0.0
```

Springback and Multi-Stage Forming

For springback or multi-stage forming simulations, LS-DYNA can output a keyword-formatted file containing the deformed mesh and stress state at the end of the forming simulation. This file, named “dynain”, will also contain strain information if the flag **strflg** above is active. The dynain file is requested using the *INTERFACE_SPRINGBACK_DYNA3D keyword. Input the id **psid** of a part set containing a list of parts to be included in the output file (usually just the sheet). An optional list of extra node constraints can be input, which are applied to the output model. These constraints provide a convenient way to eliminate rigid body motion in springback calculations.

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

To perform follow-on simulations such as springback, trimming, or additional forming, the “dynain” file can be included into a new LS-DYNA input deck for another independent simulation. This procedure avoids cumbersome binary restart databases.

Implicit Springback

Once a forming simulation has completed and a dynain file has been obtained, a second simulation can be run for springback prediction. The input deck for this simulation can be constructed using the part, section, and material information from the original forming model, and the node, element, and initial stress information contained in the dynain file. A few additional keywords must be added to control the springback process. These are described below.

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 using **dt0**, and can be chosen arbitrarily in most cases since the solution is static.

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

Springback of relatively stiff panels can sometimes be performed in a single step, but most simulations require multiple steps. Four steps is a reasonable starting point, so define the termination time **term=4*dt0** using *CONTROL_TERMINATION.

```
*CONTROL_TERMINATION
$     term
     0.004
```

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.

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”. The initial stiffness of these springs can be scaled using the input parameter **scale**. The default value of 1.0 is appropriate for thick, stiff panels. Most flexible panels will solve more rapidly if a value of **scale=0.001** is used, allowing more springback deformation during the first few steps of the simulation. If convergence of the first step is difficult, use a larger value for **scale**.

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

If convergence trouble is encountered, automatic time step control is available to back up the solution and try again 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**.

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

During each time step, the nonlinear solver searches iteratively to find equilibrium. This search is performed using a Newton-based method. By default, the BFGS method is used, but for difficult springback problems (flexible parts with large springback deformation) the full Newton method is better. To activate the full Newton method, set the iteration limit between stiffness reformations **ilimit=1**, and increase the maximum allowable stiffness reformations per time step to **maxref=100**. Activate the nonlinear solver print flag **nlprint=1** to see the progress of these iterations appear on the screen. The displacement and energy norms are displayed each iteration. These must be decreased below the 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. In some cases, the full Newton method will perform better if the line search is disabled using **lstol=99999**.

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

The stiffness matrix formed during implicit analysis requires a large amount of memory, and computing its inverse requires much CPU time. These operations are performed by the linear equation solver, whose control parameters are found on *CONTROL_IMPLICIT_LINEAR. A double precision solution to the linear system [K]{x}={f} can be selected using **lsolvr=3**, however this does not often improve results, and does increase memory requirements by 2x. A summary of memory usage is printed to the screen when the lprint flag is activated. Memory limits can be increased using the execution line argument “**memory=**”, where the default is memory=8500000.

```
*CONTROL_IMPLICIT_LINEAR
$   lsolvr    lprint    negeig
         0         0         0
```

The above recommended parameters provide a starting point for metal forming simulations using LS-DYNA. Experience and engineering judgement will likely lead to a modified set of parameters for special types of panels.