Copyright and Trademark Information

© 2009 SAS IP, Inc. All rights reserved. Unauthorized use, distribution or duplication is prohibited.

ANSYS, ANSYS Workbench, Ansoft, AUTODYN, EKM, Engineering Knowledge Manager, CFX, FLUENT, HFSS and any and all ANSYS, Inc. brand, product, service and feature names, logos and slogans are registered trademarks or trademarks of ANSYS, Inc. or its subsidiaries in the United States or other countries. ICEM CFD is a trademark used by ANSYS, Inc. under license. CFX is a trademark of Sony Corporation in Japan. All other brand, product, service and feature names or trademarks are the property of their respective owners.

Disclaimer Notice

THIS ANSYS SOFTWARE PRODUCT AND PROGRAM DOCUMENTATION INCLUDE TRADE SECRETS AND ARE CONFIDENTIAL AND PROPRIETARY PRODUCTS OF ANSYS, INC., ITS SUBSIDIARIES, OR LICENSORS. The software products and documentation are furnished by ANSYS, Inc., its subsidiaries, or affiliates under a software license agreement that contains provisions concerning non-disclosure, copying, length and nature of use, compliance with exporting laws, warranties, disclaimers, limitations of liability, and remedies, and other provisions. The software products and documentation may be used, disclosed, transferred, or copied only in accordance with the terms and conditions of that software license agreement.

ANSYS, Inc. is certified to ISO 9001:2008.

U.S. Government Rights

For U.S. Government users, except as specifically granted by the ANSYS, Inc. software license agreement, the use, duplication, or disclosure by the United States Government is subject to restrictions stated in the ANSYS, Inc. software license agreement and FAR 12.212 (for non-DOD licenses).

Third-Party Software

See the legal information in the product help files for the complete Legal Notice for ANSYS proprietary software and third-party software. If you are unable to access the Legal Notice, please contact ANSYS, Inc.

Published in the U.S.A.
# Table of Contents

## 1. Introduction
- 1.1. Starting ANSYS LS-DYNA ................................................................. 1
- 1.2. Overview of Steps in an Explicit Dynamic Analysis ......................... 1
- 1.3. Commands Used in an Explicit Dynamic Analysis ............................... 2
- 1.4. A Guide to Using this Document ..................................................... 4
- 1.5. Where to Find Explicit Dynamics Example Problems .......................... 5
- 1.6. Additional Information .................................................................. 5

## 2. Elements
- 2.1. Solid and Shell Elements .................................................................. 8
  - 2.1.1. SOLID164 .................................................................................. 8
  - 2.1.2. SHELL163 ................................................................................ 9
    - 2.1.2.1. General Shell Formulations .................................................. 9
    - 2.1.2.2. Membrane Element Formulation ......................................... 10
    - 2.1.2.3. Triangular Shell Formulations ............................................ 10
  - 2.1.3. PLANE162 .............................................................................. 13
  - 2.1.4. SOLID168 ............................................................................. 14
- 2.2. Beam and Link Elements .................................................................. 15
  - 2.2.1. BEAM161 .............................................................................. 15
  - 2.2.2. LINK160 ................................................................................ 16
  - 2.2.3. LINK167 ................................................................................ 16
- 2.3. Discrete Elements ........................................................................... 16
  - 2.3.1. COMBI165 Spring-Damper ..................................................... 16
  - 2.3.2. MASS166 .............................................................................. 17
- 2.4. General Element Capabilities ......................................................... 17

## 3. Analysis Procedure
- 3.1. Build the Model ............................................................................. 19
  - 3.1.1. Define Element Types and Real Constants ................................... 19
  - 3.1.2. Specify Material Properties ...................................................... 20
  - 3.1.3. Define the Model Geometry ...................................................... 20
  - 3.1.4. Mesh the Model ..................................................................... 20
  - 3.1.5. Define Contact Surfaces .......................................................... 21
  - 3.1.6. General Modeling Guidelines .................................................... 22
- 3.2. Apply Loads and Obtain the Solution ............................................... 22
  - 3.2.1. Loads ..................................................................................... 22
  - 3.2.2. Initial Velocities .................................................................... 23
  - 3.2.3. Constraints .......................................................................... 24
  - 3.2.4. DOF Coupling .................................................................... 24
  - 3.2.5. Data Smoothing ................................................................... 24
  - 3.2.6. Specify Explicit Dynamics Controls ........................................ 24
  - 3.2.7. Save Database and Solve ........................................................ 25
- 3.3. Review the Results ......................................................................... 25
- 3.4. The Definition of Part ...................................................................... 26
  - 3.4.1. Part Assemblies ................................................................. 29
- 3.5. Adaptive Meshing .......................................................................... 29

## 4. Loading
- 4.1. General Loading Options ................................................................. 33
  - 4.1.1. Components ........................................................................... 34
  - 4.1.2. Array Parameters ................................................................. 35
  - 4.1.3. Applying Loads ..................................................................... 36
  - 4.1.4. Data Curves .......................................................................... 38
# ANSYS LS-DYNA User’s Guide

## 4. Constraints and Initial Conditions

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1. Constraints</td>
<td>40</td>
</tr>
<tr>
<td>4.2. Constraints</td>
<td>41</td>
</tr>
<tr>
<td>4.2.1. Constraints</td>
<td>41</td>
</tr>
<tr>
<td>4.2.2. Welds</td>
<td>42</td>
</tr>
<tr>
<td>4.2.3. Initial Velocity</td>
<td>42</td>
</tr>
<tr>
<td>4.2. Initial Conditions</td>
<td>42</td>
</tr>
<tr>
<td>4.2.1. Constraints</td>
<td>42</td>
</tr>
<tr>
<td>4.2.2. Welds</td>
<td>42</td>
</tr>
<tr>
<td>4.2.3. Initial Velocity</td>
<td>42</td>
</tr>
<tr>
<td>4.2. Initial Conditions</td>
<td>42</td>
</tr>
<tr>
<td>4.3. Coupling and Constraint Equations</td>
<td>43</td>
</tr>
<tr>
<td>4.3.1. Constraints</td>
<td>43</td>
</tr>
<tr>
<td>4.3.2. Coupling and Constraint Equations</td>
<td>43</td>
</tr>
<tr>
<td>4.4. Nonreflecting Boundaries</td>
<td>44</td>
</tr>
<tr>
<td>4.5. Temperature Loading</td>
<td>44</td>
</tr>
<tr>
<td>4.6. Dynamic Relaxation</td>
<td>45</td>
</tr>
<tr>
<td>5. Solution Features</td>
<td>47</td>
</tr>
<tr>
<td>5.1. Solution Process</td>
<td>47</td>
</tr>
<tr>
<td>5.2. LS-DYNA Termination Controls</td>
<td>47</td>
</tr>
<tr>
<td>5.3. LS-DYNA Parallel Processing Capabilities</td>
<td>48</td>
</tr>
<tr>
<td>5.3.1. Shared Memory Parallel Processing</td>
<td>48</td>
</tr>
<tr>
<td>5.3.2. Massively Parallel Processing</td>
<td>49</td>
</tr>
<tr>
<td>5.4. Double Precision LS-DYNA</td>
<td>50</td>
</tr>
<tr>
<td>5.5. Solution Control and Monitoring</td>
<td>50</td>
</tr>
<tr>
<td>5.6. Plotting Small Elements</td>
<td>51</td>
</tr>
<tr>
<td>5.7. Editing the LS-DYNA Input File</td>
<td>52</td>
</tr>
<tr>
<td>5.7.1. Using a Preexisting File.K</td>
<td>54</td>
</tr>
<tr>
<td>6. Contact Surfaces</td>
<td>55</td>
</tr>
<tr>
<td>6.1. Contact Definitions</td>
<td>55</td>
</tr>
<tr>
<td>6.1.1. Listing, Plotting and Deleting Contact Entities</td>
<td>56</td>
</tr>
<tr>
<td>6.2. Contact Options</td>
<td>58</td>
</tr>
<tr>
<td>6.2.1. Definition of Contact Types</td>
<td>59</td>
</tr>
<tr>
<td>6.2.2. Definition of Contact Options</td>
<td>59</td>
</tr>
<tr>
<td>6.3. Contact Search Methods</td>
<td>60</td>
</tr>
<tr>
<td>6.3.1. Mesh Connectivity Tracking</td>
<td>60</td>
</tr>
<tr>
<td>6.3.2. Bucket Sort Method</td>
<td>61</td>
</tr>
<tr>
<td>6.3.3. Limiting the Contact Search Domain</td>
<td>61</td>
</tr>
<tr>
<td>6.4. Special Considerations for Shells</td>
<td>62</td>
</tr>
<tr>
<td>6.5. Controlling Contact Depth</td>
<td>63</td>
</tr>
<tr>
<td>6.6. Contact Stiffness</td>
<td>63</td>
</tr>
<tr>
<td>6.6.1. Choice of Penalty Factor</td>
<td>64</td>
</tr>
<tr>
<td>6.6.2. Symmetry Stiffness</td>
<td>65</td>
</tr>
<tr>
<td>6.7. 2-D Contact Option</td>
<td>65</td>
</tr>
<tr>
<td>7. Material Models</td>
<td>67</td>
</tr>
<tr>
<td>7.1. Defining Explicit Dynamics Material Models</td>
<td>68</td>
</tr>
<tr>
<td>7.2. Explicit Dynamics Material Model Descriptions</td>
<td>69</td>
</tr>
<tr>
<td>7.2.1. Linear Elastic Models</td>
<td>70</td>
</tr>
<tr>
<td>7.2.1.1. Isotropic Elastic Model</td>
<td>70</td>
</tr>
<tr>
<td>7.2.1.2. Orthotropic Elastic Model</td>
<td>70</td>
</tr>
<tr>
<td>7.2.1.3. Anisotropic Elastic Model</td>
<td>70</td>
</tr>
<tr>
<td>7.2.1.4. Elastic Fluid Model</td>
<td>71</td>
</tr>
<tr>
<td>7.2.2. Nonlinear Elastic Models</td>
<td>72</td>
</tr>
<tr>
<td>7.2.2.1. Blatz-Ko Rubber Elastic Model</td>
<td>72</td>
</tr>
<tr>
<td>7.2.2.2. Mooney-Rivlin Rubber Elastic Model</td>
<td>72</td>
</tr>
<tr>
<td>7.2.2.3. Viscoelastic Model</td>
<td>73</td>
</tr>
<tr>
<td>7.2.3. Nonlinear Inelastic Models</td>
<td>74</td>
</tr>
</tbody>
</table>

Release 12.1 - © 2009 SAS IP, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.
8. Rigid Bodies

8.1. Defining Rigid Bodies ................................................. ................................................... ................ 105
8.2. Specifying Inertia Properties ............................................... ................................................... ....... 105
8.3. Loading ............................................................................ ................................................... ............ 106
8.4. Switching Parts from Deformable to Rigid .................................................. ................................... 106
8.5. Nodal Rigid Bodies .................................................. ................................................... ............. 107
9. Hourglassing ................................................................. 109
10. Mass Scaling .................................................................. 111
11. Subcycling .................................................................... 113
12. Postprocessing ................................................................ 115
   12.1. Output Controls .......................................................... 115
   12.1.1. Results (Jobname.RST) vs. History (Jobname.HIS) Files ......................................................... 115
   12.1.2. Creating Components for POST26 ....................................................................................... 115
   12.1.3. Writing the Output Files for POST26 ..................................................................................... 116
   12.2. Using POST1 with ANSYS LS-DYNA ....................................................................................... 116
   12.2.1. Animating Results .................................................................................................................. 117
   12.2.2. Element Output Data .............................................................................................................. 117
   12.2.3. Postprocessing after Adaptive Meshing .................................................................................. 118
   12.3. Using POST26 with ANSYS LS-DYNA ...................................................................................... 120
   12.3.1. Nodal and Element Solutions .................................................................................................. 120
   12.3.2. Reading ASCII Files for Miscellaneous Output Data .............................................................. 121
   12.3.3. Data Smoothing ....................................................................................................................... 121
   12.4. Finding Additional Information ............................................................................................... 121
13. Restarting ........................................................................ 123
   13.1. The Restart Dump File .................................................... 123
   13.2. The EDSTART Command ................................................. 123
   13.2.1. A New Analysis ...................................................................................................................... 124
   13.2.2. A Simple Restart ........................................................ 124
   13.2.3. A Small Restart ....................................................................................................................... 124
   13.2.4. A Full Restart ........................................................................................................................... 125
   13.3. Effect on Output Files ...................................................... 127
14. Explicit-to-Implicit Sequential Solution ................................. 129
   14.1. Performing an Explicit-to-Implicit Sequential Solution ........................................................... 129
   14.2. Troubleshooting a Springback Analysis ....................................................................................... 132
   14.2.1. Springback Stabilization ......................................................................................................... 133
15. Implicit-to-Explicit Sequential Solution .................................... 135
   15.1. Structural Implicit-to-Explicit Solution for Preload ..................................................................... 135
   15.1.1. Special Considerations for Thermal Loading ............................................................................ 139
   15.2. Thermal Implicit-to-Explicit Solution ....................................................................................... 139
16. Arbitrary Lagrangian-Eulerian Formulation ............................... 145
   16.1. Performing an ALE Analysis ............................................... 147
17. Drop Test Module ................................................................ 149
   17.1. Typical Drop Test Procedure ............................................ 149
   17.1.1. Basic Drop Test Analysis Procedure ....................................................................................... 150
         17.1.1.1. STEP 1: Create or import the model ............................................................................... 150
         17.1.1.2. STEP 2: Set up the DTM .................................................................................................. 150
         17.1.1.3. STEP 3: Define the magnitude of (g) .............................................................................. 151
         17.1.1.4. STEP 4: Specify the drop height ....................................................................................... 151
         17.1.1.5. STEP 5: Orient the object .................................................................................................. 151
         17.1.1.6. STEP 6: Specify solution controls ..................................................................................... 151
         17.1.1.7. STEP 7: Solve .................................................................................................................... 151
         17.1.1.8. STEP 8: Animate results .................................................................................................. 152
         17.1.1.9. STEP 9: Obtain Time-History Results .............................................................................. 152
   17.1.2. Screen Coordinates Definition ............................................................................................. 152
   17.1.3. Additional Notes on the Use of the DTM .............................................................................. 153
17.2. Advanced DTM Features ................................................... 153
   17.2.1. Object Initial Velocity .................................................... 153
   17.2.2. Modifying the Target ............................................................................................................ 154
A. Comparison of Implicit and Explicit Methods ............................................... 167
B. Material Model Examples ............................................................................. 171

A.1. Time Integration ....................................................................................... 167
A.1.1. Implicit Time Integration ...................................................................... 167
A.1.2. Explicit Time Integration ..................................................................... 167
A.2. Stability Limit ......................................................................................... 168
A.2.1. Implicit Method .................................................................................. 168
A.2.2. Explicit Method ................................................................................... 168
A.3. Critical Time Step Size of a Rod ............................................................... 169
A.4. ANSYS LS-DYNA Time Step Size ............................................................. 169

B.1. ANSYS LS-DYNA Material Models ......................................................... 171
B.2. Material Model Examples ......................................................................... 173
B.2.1. Isotropic Elastic Example: High Carbon Steel ....................................... 173
B.2.2. Orthotropic Elastic Example: Aluminum Oxide ..................................... 174
B.2.3. Anisotropic Elastic Example: Cadmium ............................................... 174
B.2.4. Blatz-Ko Example: Rubber .................................................................. 174
B.2.5. Mooney-Rivlin Example: Rubber .......................................................... 174
B.2.6. Viscoelastic Example: Glass ................................................................ 174
B.2.7. Bilinear Isotropic Plasticity Example: Nickel Alloy .............................. 175
B.2.8. Transversely Anisotropic Elastic Plastic Example: 1010 Steel .............. 175
B.2.9. Transversely Anisotropic FLD Example: Stainless Steel .................... 175
B.2.10. Bilinear Kinematic Plasticity Example: Titanium Alloy ....................... 176
B.2.11. Plastic Kinematic Example: 1018 Steel ............................................. 176
B.2.12. 3 Parameter Barlat Example: Aluminum 5182 .................................... 176
B.2.14. Rate Sensitive Powerlaw Plasticity Example: A356 Aluminum ............ 177
B.2.15. Strain Rate Dependent Plasticity Example: 4140 Steel ....................... 177
B.2.16. Piecewise Linear Plasticity Example: High Carbon Steel .................. 178
B.2.17. Modified Piecewise Linear Plasticity Example: PVC ......................... 178
B.2.18. Powerlaw Plasticity Example: Aluminum 1100 .................................... 179
B.2.19. Elastic Viscoplastic Thermal Example ............................................... 179
B.2.20. Geological Cap Example: SRI Dynamic Concrete ............................... 180
B.2.21. Johnson-Cook Linear Polynomial EOS Example: 1006 Steel .......... 181
B.2.22. Johnson-Cook Gruneisen EOS Example: OFHC Copper .................. 181
B.2.23. Null Material Linear Polynomial EOS Example: Brass .................... 182
B.2.25. Steinberg Gruneisen EOS Example: Stainless Steel ......................... 183
B.2.26. Cable Material Example: Steel ........................................................... 183

17.3. Drop Test Set-up Dialog Box ................................................................. 156
17.3.1. Using the Drop Test Set-up Dialog Box ............................................ 156
17.3.2. Basic Tab of the Drop Test Set-up Dialog Box .................................. 157
17.3.3. Velocity Tab of the Drop Test Set-up Dialog Box ............................. 159
17.3.4. Target Tab of the Drop Test Set-up Dialog Box ............................... 160
17.3.5. Status Tab of the Drop Test Set-up Dialog Box ............................... 162

17.4. Picking Nodes ......................................................................................... 163
17.5. Postprocessing - Animation ................................................................. 163
17.6. Postprocessing - Graph and List Time-History Variables ......................... 164
List of Figures

2.1. Integration Points ........................................................................................................... 12
4.1. Constrained Shell to Solid ........................................................................................... 41
6.1. LS-DYNA Drawbead Representation ........................................................................... 63
7.1. Surface of the Two-invariant Cap Model ...................................................................... 88
9.1. Hourglass Deformations .............................................................................................. 109
11.1. Time Step Sizes Before and After Subcycling ............................................................... 113
16.1. High Speed Impact of a Metal Bar .............................................................................. 145
16.2. Lagrangian Impact Solution ....................................................................................... 146
16.3. Eulerian Channel Flow Solution .................................................................................. 146
16.4. ALE Impact Solution ................................................................................................ 147
17.1. Two Views of the Target ............................................................................................. 154
17.2. Drop Test Set-up Dialog Box - Basic Tab ................................................................. 157
17.3. Drop Test Set-up Dialog Box - Velocity Tab .............................................................. 159
17.4. Drop Test Set-up Dialog Box - Target Tab ................................................................. 160
17.5. Drop Test Set-up Dialog Box - Status Tab ................................................................. 162
17.6. Graph and Time-History Variables Dialog Box ............................................................. 165

List of Tables

3.1. Loads Applicable in an Explicit Dynamics Analysis ...................................................... 23
3.2. LS-DYNA Solution and Output Control Options ........................................................... 24
Chapter 1: Introduction

ANSYS LS-DYNA combines the LS-DYNA explicit finite element program with the powerful pre- and postprocessing capabilities of the ANSYS program. The explicit method of solution used by LS-DYNA provides fast solutions for short-time, large deformation dynamics, quasi-static problems with large deformations and multiple nonlinearities, and complex contact/impact problems. Using this integrated product, you can model your structure in ANSYS, obtain the explicit dynamic solution via LS-DYNA, and review results using the standard ANSYS postprocessing tools.

You can also transfer geometry and results information between ANSYS and ANSYS LS-DYNA to perform sequential implicit-explicit / explicit-implicit analyses, such as those required for droptest, springback and other applications.

The following ANSYS LS-DYNA general information topics are available:
  1.1. Starting ANSYS LS-DYNA
  1.2. Overview of Steps in an Explicit Dynamic Analysis
  1.3. Commands Used in an Explicit Dynamic Analysis
  1.4. A Guide to Using this Document
  1.5. Where to Find Explicit Dynamics Example Problems
  1.6. Additional Information

1.1. Starting ANSYS LS-DYNA

You can start ANSYS LS-DYNA from the ANSYS Launcher or from the command level. See "Running the ANSYS Program" in the Operations Guide for more information on these two methods.

If you are starting ANSYS LS-DYNA at the command level, you need to use the -dyn command option for ANSYS Structural and higher license levels that do not have LS-DYNA enabled. As an alternative to this command line option, you can set the environment variable ANSYS12_DYN to ON, or select the LS-DYNA (-dyn) Add-on Module from the launcher.

The following example shows the command line method for running ANSYS Multiphysics (feature name ane3fl) with LS-DYNA as an additional license:

```
ansys120 -p ane3fl -dyn
```

1.2. Overview of Steps in an Explicit Dynamic Analysis

The procedure for an explicit dynamic analysis is similar to any other analysis that is available in the ANSYS program. The three main steps are:

1. Build the model (with the PREP7 preprocessor)
2. Apply loads and obtain the solution (with the SOLUTION processor)
3. Review the results (with the POST1 and POST26 postprocessors)

This document describes procedures and concepts that are unique to an explicit dynamic analysis performed using the ANSYS LS-DYNA product. It does not describe all details of the three steps listed above. If you are
familiar with the ANSYS program, you already know how to perform these steps, and this document will provide additional information you need to perform an explicit dynamic analysis. If you have not used ANSYS before, you should review the following two manuals (in addition to this document) to learn the basic analysis procedures:

- Basic Analysis Guide
- Modeling and Meshing Guide

When using ANSYS LS-DYNA, it is recommended that you use the default settings provided by the program. Many times, these settings are appropriate for solving the problems.

### 1.3. Commands Used in an Explicit Dynamic Analysis

You use the same set of commands to build a model and perform an explicit dynamic analysis that you use to do any other type of ANSYS analysis. Likewise, you choose similar options from the ANSYS program’s graphical user interface (GUI) to build and solve models.

However, some commands are unique to an explicit dynamic analysis. These commands are:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDADAPT</td>
<td>Activates adaptive meshing</td>
</tr>
<tr>
<td>EDALE</td>
<td>Assigns mesh smoothing to explicit dynamic elements that use the ALE formulation</td>
</tr>
<tr>
<td>EDASMP</td>
<td>Creates a part assembly</td>
</tr>
<tr>
<td>EDBOUND</td>
<td>Defines a boundary plane for sliding or cyclic symmetry</td>
</tr>
<tr>
<td>EDBVIS</td>
<td>Specifies bulk viscosity coefficients</td>
</tr>
<tr>
<td>EDBX</td>
<td>Creates a box shaped volume to be used in a contact definition</td>
</tr>
<tr>
<td>EDCADAPT</td>
<td>Specifies adaptive meshing controls</td>
</tr>
<tr>
<td>EDCGEN</td>
<td>Specifies contact parameters</td>
</tr>
<tr>
<td>EDCLIST</td>
<td>Lists contact entity specifications</td>
</tr>
<tr>
<td>EDCMORE</td>
<td>Specifies additional contact parameters for a given contact definition</td>
</tr>
<tr>
<td>EDCNSTR</td>
<td>Defines various types of constraints</td>
</tr>
<tr>
<td>EDCONTACT</td>
<td>Specifies contact surface controls</td>
</tr>
<tr>
<td>EDCPU</td>
<td>Specifies CPU time limit</td>
</tr>
<tr>
<td>EDCRB</td>
<td>Merges two rigid bodies</td>
</tr>
<tr>
<td>EDCSC</td>
<td>Specifies whether subcycling will be used</td>
</tr>
<tr>
<td>EDCTS</td>
<td>Specifies mass scaling</td>
</tr>
<tr>
<td>EDCURVE</td>
<td>Specifies data curves</td>
</tr>
<tr>
<td>EDDAMP</td>
<td>Defines system damping</td>
</tr>
<tr>
<td>EDDBL</td>
<td>Selects numerical precision type</td>
</tr>
<tr>
<td>EDDC</td>
<td>Deletes or deactivates/reactivates contact entity specifications</td>
</tr>
<tr>
<td>EDDRELAX</td>
<td>Activates initialization to a prescribed geometry or dynamic relaxation for the explicit analysis</td>
</tr>
<tr>
<td>EDDUMP</td>
<td>Specifies output frequency for the restart file (d3dump)</td>
</tr>
<tr>
<td>EENERGY</td>
<td>Specifies energy dissipation controls</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EDFPLOT</td>
<td>Specifies plotting of load symbols</td>
</tr>
<tr>
<td>EDGCALE</td>
<td>Defines global ALE controls</td>
</tr>
<tr>
<td>EDHGLS</td>
<td>Specifies the hourglass coefficient</td>
</tr>
<tr>
<td>EDHIST</td>
<td>Specifies time-history output</td>
</tr>
<tr>
<td>EDHTIME</td>
<td>Specifies the time-history output interval</td>
</tr>
<tr>
<td>EDINT</td>
<td>Specifies number of integration points for output</td>
</tr>
<tr>
<td>EDIPART</td>
<td>Defines inertia for rigid parts</td>
</tr>
<tr>
<td>EDIS</td>
<td>Specifies stress initialization in a full restart analysis</td>
</tr>
<tr>
<td>EDLCS</td>
<td>Defines a local coordinate system</td>
</tr>
<tr>
<td>EDLOAD</td>
<td>Specifies loads</td>
</tr>
<tr>
<td>EDMP</td>
<td>Defines material properties</td>
</tr>
<tr>
<td>EDNB</td>
<td>Defines a nonreflecting boundary</td>
</tr>
<tr>
<td>EDNDTSD</td>
<td>Smooths noisy data and provides a graphical representation of the data</td>
</tr>
<tr>
<td>EDNROT</td>
<td>Applies a rotated coordinate nodal constraint</td>
</tr>
<tr>
<td>EDOPT</td>
<td>Specifies the type of output, ANSYS or LS-DYNA</td>
</tr>
<tr>
<td>EDOUT</td>
<td>Specifies LS-DYNA ASCII output files</td>
</tr>
<tr>
<td>EDPART</td>
<td>Creates, updates, or lists parts</td>
</tr>
<tr>
<td>EDPC</td>
<td>Selects and plots contact entities</td>
</tr>
<tr>
<td>EDPL</td>
<td>Plots time dependent load curves</td>
</tr>
<tr>
<td>EDPVEL</td>
<td>Applies initial velocities to parts or part assemblies</td>
</tr>
<tr>
<td>EDRC</td>
<td>Specifies rigid/deformable switch controls</td>
</tr>
<tr>
<td>EDRD</td>
<td>Switches a part from deformable to rigid or from rigid to deformable</td>
</tr>
<tr>
<td>EDREAD</td>
<td>Reads output from LS-DYNA ASCII files into POST26 variables</td>
</tr>
<tr>
<td>EDRI</td>
<td>Defines inertia properties for a new rigid body that is created when a deformable part is switched to rigid</td>
</tr>
<tr>
<td>EDRST</td>
<td>Specifies time increment for output to the .RST file</td>
</tr>
<tr>
<td>EDRUN</td>
<td>Specifies LS-DYNA serial or parallel processing</td>
</tr>
<tr>
<td>EDSHELL</td>
<td>Specifies shell computation controls</td>
</tr>
<tr>
<td>EDSOLV</td>
<td>Specifies &quot;explicit dynamic solution&quot; as the subsequent status topic</td>
</tr>
<tr>
<td>EDSP</td>
<td>Specifies small penetration checking for contact entities</td>
</tr>
<tr>
<td>EDSTART</td>
<td>Specifies status (new or restart) of the analysis</td>
</tr>
<tr>
<td>EDTERM</td>
<td>Specifies termination criteria</td>
</tr>
<tr>
<td>EDTP</td>
<td>Plots elements based on their time step size</td>
</tr>
<tr>
<td>EDVEL</td>
<td>Applies initial velocities to nodes or node components</td>
</tr>
<tr>
<td>EDWELD</td>
<td>Defines a massless spotweld or generalized weld</td>
</tr>
<tr>
<td>EDWRITE</td>
<td>Writes explicit dynamics input to an LS-DYNA input file</td>
</tr>
<tr>
<td>PARTSEL</td>
<td>Selects a subset of parts</td>
</tr>
<tr>
<td>RIMPORT</td>
<td>Imports initial stresses from an explicit run into ANSYS</td>
</tr>
</tbody>
</table>
Chapter 1: Introduction

REXPORT
Exports displacements from an implicit run to ANSYS LS-DYNA

UPGEOM
Adds displacements from a previous analysis and updates the
geometry to the deformed configuration

For detailed alphabetized descriptions of the ANSYS commands (including specific menu path information
for each command), see the Command Reference.

1.4. A Guide to Using this Document

This document contains both procedural and reference information. You may choose to read through it
from front to back. However, it may be more useful to read the chapters in a sequence that corresponds
with the process of planning and performing an explicit dynamic analysis.

Before you build a model, you must decide which element types and material models will best represent
your physical system. The following chapters provide the background information you need to make these
decisions:

Chapter 2, Elements (p. 7)
Chapter 7, Material Models (p. 67)

After choosing appropriate element types and material models, you are ready to build the model. The typi-
cal aspects of model building are presented in:

Chapter 3, Analysis Procedure (p. 19)
Chapter 6, Contact Surfaces (p. 55)
Chapter 8, Rigid Bodies (p. 105)
Chapter 4, Loading (p. 33)

Features related to solving and postprocessing are discussed in:

Chapter 5, Solution Features (p. 47)
Chapter 12, Postprocessing (p. 115)

Other more advanced capabilities that are not required to complete an analysis, but that may be useful in
some situations, are presented in:

Chapter 9, Hourglassing (p. 109)
Chapter 10, Mass Scaling (p. 111)
Chapter 11, Subcycling (p. 113)
Chapter 13, Restarting (p. 123)
Chapter 14, Explicit-to-Implicit Sequential Solution (p. 129)
Chapter 15, Implicit-to-Explicit Sequential Solution (p. 135)
Chapter 16, Arbitrary Lagrangian-Eulerian Formulation (p. 145)
Chapter 17, Drop Test Module (p. 149)

Finally, the appendices contain reference information on the following topics:

Appendix A (p. 167)
Appendix B (p. 171)
Appendix C (p. 185)
1.5. Where to Find Explicit Dynamics Example Problems

The Explicit Dynamics Tutorial describes a sample explicit dynamic analysis problem.

1.6. Additional Information

For a more detailed overview of the explicit dynamic analysis procedure, see the Structural Analysis Guide. For detailed information on explicit elements, see the Element Reference. For detailed theoretical information, see the Livermore Software Technology Corporation's LS-DYNA Theoretical Manual.
Chapter 2: Elements

The following elements can be used in an explicit dynamic analysis:

- LINK160
- BEAM161
- PLANE162
- SHELL163
- SOLID164
- COMBI165
- MASS166
- LINK167
- SOLID168

All of the explicit dynamic elements listed above are 3-D, except for PLANE162. All of the elements listed above use reduced integration by default, when applicable. Reduced integration means that the number of points for numerical integration in the element formation process is less than that necessary for exact integration. Therefore, one integration point is the default formulation for solid, shell, plane, and beam elements. Full integration options are also available for these elements. For LINK160 and LINK167 elements, one-point integration is enough for exact integration. For COMBI165 and MASS166 elements, no element integration is needed. For additional details, see Chapter 9, Hourglassing (p. 109) and also the LS-DYNA Theoretical Manual.

These elements assume a linear displacement function, with the following exceptions: SOLID168 elements, which have midside nodes, and MASS166 elements, which have only one node. The explicit dynamic elements are not available with extra shape functions or p-elements. Explicit elements with linear displacement functions and one point integration are best suited for nonlinear applications with large deformations and material failure.

It should be noted that the explicit elements are not linked directly to material properties. For example, the SOLID164 element supports more than 20 different material models, including elastic, plastic, rubber, and foam models. Unless stated otherwise (see Chapter 6, Contact Surfaces (p. 55)), the minimum material property requirements for all elements are density, Poisson’s ratio, and elastic modulus. See Chapter 7, Material Models (p. 67) of this document for detailed information on the material properties that can be used in an explicit dynamic analysis. See also the Element Reference for a complete description of each element, including details of input and output quantities.

The following explicit element topics are available:

2.1. Solid and Shell Elements
2.2. Beam and Link Elements
2.3. Discrete Elements
2.4. General Element Capabilities
2.1. Solid and Shell Elements

2.1.1. SOLID164

The SOLID164 element is an 8-node brick element. By default, it uses reduced (one point) integration plus viscous hourglass control for faster element formulation. One point integration is advantageous due to savings on computer time and robustness in cases of large deformations. A fully integrated solid formulation (KEYOPT(1) = 2) is also available; see the element description for SOLID164 in the Element Reference and Section 3.3 of the LS-DYNA Theoretical Manual. If hourglass phenomenon is a concern, such as with foams, the fully integrated formulation may perform better because hourglass control is not required; but it is about four times more costly in terms of CPU time.

KEYOPT(5) defines the element continuum treatment. Two different formulations are available: Lagrangian (default) and Arbitrary Lagrangian-Eulerian (ALE). See Arbitrary Lagrangian-Eulerian Formulation in the ANSYS LS-DYNA User's Guide for more information.

Wedge, pyramid, and tetrahedra shaped SOLID164 elements are simply degenerate bricks (i.e., some of the nodes are repeated). These shapes are often too stiff in bending and cause problems in some situations. Therefore, these degenerate shapes should be avoided. If a tetrahedron mesh is desired, it is recommended that SOLID168 elements be used instead of degenerated SOLID164 elements.

You can use any of the following material models for solid elements:

- Isotropic Elastic
- Orthotropic Elastic
- Anisotropic Elastic
- Bilinear Kinematic
- Plastic Kinematic
- Viscoelastic
- Blatz-Ko Rubber
- Bilinear Isotropic
- Power Law Plasticity
- Strain Rate Dependent Plasticity
- Composite Damage
- Concrete Damage
- Geological Cap
- Piecewise Linear Plasticity
- Honeycomb
- Mooney-Rivlin Rubber
- Barlat Anisotropic Plasticity
- Elastic-Plastic Hydrodynamic
- Rate Sensitive Powerlaw Plasticity
- Elastic Viscoplastic Thermal
- Closed Cell Foam
2.1.2. SHELL163

Twelve different formulations are available for the SHELL163 element. Use KEYOPT(1) to specify the desired formulation. As with solid elements, the number of integration points per element directly impacts CPU time. Therefore, for general analyses, the reduced integration shell formulations are recommended. The following is an outline of the different formulations available for SHELL163. Coordinate systems for output of quantities associated with SHELL163 may vary for different formulations. See the Element Reference for more information.

2.1.2.1. General Shell Formulations

- Belytschko-Tsay (KEYOPT(1) = 0 or 2) - default
  - Very fast and is recommended for most applications
  - Uses reduced integration (one point)
  - Should not be used when elements experience excessive warping
- Belytschko-Wong-Chiang (KEYOPT(1) = 10)
  - 25% slower than Belytschko-Tsay
  - Uses reduced integration (one point)
  - Generally gives correct results for warping
- Belytschko-Leviathan (KEYOPT(1) = 8)
  - 40% slower than Belytschko-Tsay
  - Reduced integration formulation (one point)
  - Includes physical hourglass control automatically
- Fully integrated Belytschko-Tsay (KEYOPT(1) = 12)
  - 2.5 times slower than reduced integration Belytschko-Tsay shell.
  - Has four integration points in plane and does not need hourglass control.
  - Shear locking is remedied by assumed strain for the transverse shear.
  - Recommended if hourglass modes are a problem in the analysis.
- Hughes-Liu (KEYOPT(1) = 1, 6, 7, 11) = > four different formulations which can offset the mid-plane of element away from the nodes.
  - KEYOPT(1) = 1 General Hughes-Liu. Has one-point integration and is 2.5 times slower that Belytschko-Tsay.
Chapter 2: Elements

- KEYOPT(1) = 11 Fast (corotational) Hughes-Liu. Has one-point integration and is 1.5 times slower than Belytschko-Tsay.
- KEYOPT(1) = 6 S/R Hughes-Liu. Has four integration points with no hourglassing but is 20 times slower than Belytschko-Tsay.
- KEYOPT(1) = 7 S/R corotational Hughes-Liu. Has four integration points with no hourglassing. 8.8 times slower than Belytschko-Tsay. Recommended if hourglassing is a problem in an analysis.

2.1.2.2. Membrane Element Formulation

- Belytschko-Tsay Membrane (KEYOPT(1) = 5)
  - Fast and recommended for most membrane applications
  - Reduced (one-point) integration
  - Good for fabrics where wrinkling is a concern (i.e., where large in-plane compressive stresses try to collapse the thin fabric elements)
- Fully integrated Belytschko-Tsay Membrane (KEYOPT(1) = 9)
  - Significantly slower than general membrane (KEYOPT(1) = 5)
  - Four integration points in plane
  - No hourglassing

2.1.2.3. Triangular Shell Formulations

- \( C^0 \) triangular shell (KEYOPT(1) = 4)
  - Based on Mindlin-Reissner Plate Theory
  - Formulation is rather stiff, so not recommended for entire mesh
  - Reduced integration formulation
- BCIZ Triangular Shell (KEYOPT(1) = 3)
  - Based on Kirchhoff Plate Theory
  - Slower than \( C^0 \) triangular shells
  - Reduced integration formulation

The description of SHELL163 in the ANSYS LS-DYNA User's Guide also has a complete list of available shell formulations.

Degenerate quadrilateral shell elements are prone to lock under transverse shear; therefore, \( C^0 \) triangular shell elements (based on work by Belytschko and coworkers) have been implemented. Triangular shells can be mixed with quadrilateral shells within the same material property set, provided that the element sorting flag (the \( ITRST \) field on the EDSHELL command) is set to 1 (which is the default).

You can use any of the following material models for shell elements:

- Isotropic Elastic
- Orthotropic Elastic
- Bilinear Kinematic
- Plastic Kinematic
- Blatz-Ko Rubber
• Bilinear Isotropic
• Power Law Plasticity
• Strain Rate Dependent Plasticity
• Composite Damage
• Piecewise Linear Plasticity
• Modified Piecewise Linear Plasticity
• Mooney-Rivlin Rubber (see note below)
• Barlat Anisotropic Plasticity
• 3 Parameter Barlat Plasticity
• Transversely Anisotropic Elastic Plastic
• Rate Sensitive Powerlaw Plasticity
• Transversely Anisotropic FLD
• Elastic Viscoplastic Thermal
• Johnson-Cook Plasticity
• Bamman

Note

When the Mooney-Rivlin Rubber material model is used with SHELL163 elements, the LS-DYNA code will automatically use a total Lagrangian modification of the Belytschko-Tsay formulation instead of using the formulation you specify via KEYOPT(1). This program-chosen formulation is required to address the special needs of the hyperelastic material.

All shell element formulations can have an arbitrary number of integration points through the thickness. Typically, 2 integration points (default) are required through the thickness for elastic behavior, while 3 or more integration points are required for plastic behavior. The number of integration points through the thickness is controlled using the second real constant:

R,NSET,R2 where R2 = number of integration points (NIP)

3-D plane stress constitutive subroutines are implemented for the shell elements; these update the stress tensor such that the stress component normal to the shell mid surface is zero. The integration points are stacked vertically at the centroid of the element, as shown in Figure 2.1: Integration Points (p. 12).
Through-thickness directions at each node are initially normal to the element surface but rotate with the nodes. The through-the-thickness integration is needed to calculate bending moments and in-plane forces. The distribution of strain is always linear; whereas the stress is more complicated in nature and depends on the material law.

Two integration points are sufficient for linear elastic materials, while more points are required for nonlinear materials. Stress output is given at the outermost integration points, not at the surfaces (despite the nomenclature of postprocessors, which refer to top and bottom surfaces), so care is needed in interpretation of results. For elastic materials, stresses can be extrapolated to the surfaces. For nonlinear materials the usual procedure is to choose four or five integration points through the thickness and to ignore the inaccuracy (i.e., the difference in stress between the surface and the outermost integration point). The locations of the outermost integration points of Gauss quadrature are given in the following table:

<table>
<thead>
<tr>
<th>Mid Plane</th>
<th>Outermost Point</th>
<th>Outer Surface</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2 points</td>
<td>1.0000</td>
</tr>
<tr>
<td></td>
<td>3 points</td>
<td>0.8611</td>
</tr>
<tr>
<td></td>
<td>4 points</td>
<td>0.9062</td>
</tr>
<tr>
<td></td>
<td>5 points</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

**Note**

Keep in mind that this integration could be made exactly beforehand when linear elastic material is used, but it is not done in ANSYS LS-DYNA because nonlinear behavior is generally being modeled.

Also, for elements using full integration, the output stress is the averaged stress values from the 2x2 integration points of the same layer.
2.1.3. PLANE162

The PLANE162 element is a 2-D, 4-node solid element that can be used either as a planar (X-Y plane) or as an axisymmetric (Y-axis of symmetry) element. KEYOPT(3) allows you to specify a plane stress, axisymmetric, or plane strain option for the element. For the axisymmetric element formulation, you can specify either an area or volume weighted option using KEYOPT(2).

KEYOPT(5) defines the element continuum treatment. Two different formulations are available: Lagrangian (default) and Arbitrary Lagrangian-Eulerian (ALE). See Arbitrary Lagrangian-Eulerian Formulation in the ANSYS LS-DYNA User's Guide for more information.

The PLANE162 element is typically defined by four-nodes. A three-node triangle option is also available, but not recommended because it is often too stiff. There are no real constants for this element. It is important to note that models containing PLANE162 elements must consist only of this element type. ANSYS LS-DYNA does not allow a mixed 2-D and 3-D finite element model. Furthermore, all PLANE162 elements in the model must be the same type (plane stress, plane strain, or axisymmetric).

The material models available to use with this element will depend on the KEYOPT(3) setting. For KEYOPT(3) = 0, 1, or 2 (plane stress, plane strain, or axisymmetric), you can choose the following materials:

- Isotropic Elastic
- Orthotropic Elastic
- Elastic Fluid
- Viscoelastic
- Bilinear Isotropic
- Temperature Dependent Bilinear Isotropic
- Bilinear Kinematic
- Plastic Kinematic
- Power Law Plasticity
- Rate Sensitive Powerlaw Plasticity
- Strain Rate Dependent Plasticity
- Piecewise Linear Plasticity
- Composite Damage
- Johnson-Cook Plasticity
- Bamman

For the plane stress option (KEYOPT(3) = 0), you can also choose the following materials:

- 3-Parameter Barlat Plasticity
- Barlat Anisotropic Plasticity
- Transversely Anisotropic Elastic Plastic
- Transversely Anisotropic FLD

For the axisymmetric and plane strain options (KEYOPT(3) = 1 or 2), you can also choose the following materials:

- Blatz-Ko Rubber
2.1.4. SOLID168

The SOLID168 element is a 10-node tetrahedron element. It uses either a quadratic displacement interpolation with five point integration (KEYOPT(1) = 0 or 1) or a composite assemblages of sub-tetrahedral shapes (KEYOPT(1) = 2); see the element description for SOLID168 in the Element Reference.

You can use any of the following material models for solid elements:

- Isotropic Elastic
- Orthotropic Elastic
- Anisotropic Elastic
- Bilinear Kinematic
- Plastic Kinematic
- Viscoelastic
- Blatz-Ko Rubber
- Bilinear Isotropic
- Power Law Plasticity
- Strain Rate Dependent Plasticity
- Composite Damage
- Concrete Damage
- Geological Cap
- Piecewise Linear Plasticity
- Honeycomb
- Mooney-Rivlin Rubber
- Barlat Anisotropic Plasticity
- Elastic-Plastic Hydrodynamic
- Rate Sensitive Power Law Plasticity
- Elastic Viscoplastic Thermal
- Closed Cell Foam
• Low Density Foam
• Viscous Foam
• Crushable Foam
• Johnson-Cook Plasticity
• Null
• Zerilli-Armstrong
• Bamman
• Steinberg
• Elastic Fluid

**SOLID168** elements are well suited to modeling irregular meshes such as those produced from various CAD/CAM systems and the ANSYS Workbench. It is important to note, however, that models made up entirely of **SOLID168** elements may not be as accurate as hexahedral (**SOLID164**) models. For this reason, we recommend using a combination of **SOLID168** and **SOLID164** elements within a model as a good modeling practice. The **SOLID168** elements could be used for the irregular portions of the model while the **SOLID164** elements could be used to mesh the more uniform features.

Although **SOLID164** and **SOLID168** elements can be used in the same model, you must be careful to develop an adequate mesh transition between these two element types. This is due to the fact that **SOLID168** elements contain mid-side nodes, which are not allowed in **SOLID164** elements.

### 2.2. Beam and Link Elements

#### 2.2.1. BEAM161

The **BEAM161** element has two basic formulations: **Hughes-Liu** and **Belytschko-Schwer**. **BEAM161** is best suited for rigid body rotations because it does not generate any strains. Three nodes must be used to define the element; a node at each endpoint and an orientation node are required. For both formulations, several beam cross sections can be defined using KEYOPTs (4) and (5). In general, **BEAM161** is efficient and robust for 2x2 gauss quadratic integration (default). Several different integration formulations can be defined with KEYOPT(2).

The **Hughes-Liu** beam (default) is a conventionally integrated element that can model rectangular or circular cross-sections using an array of integration points at the mid-span of the element. Alternatively, you can specify a cross-section integration rule to model arbitrary cross-sections. The beam effectively generates a constant moment along its length, so, as with the brick and shell elements, meshes need to be reasonably fine to achieve adequate accuracy. Because of the location of the integration points, yielding is detected only at the element center. Therefore, a cantilever beam model will yield at slightly too high a force because the fully plastic moment must be developed at the center of the clamped element rather than at the outer surface.

The **Belytschko-Schwer** beam (KEYOPT(1) = 2, 4, or 5) is explicitly formulated and generates a moment which varies linearly along the length of the beam. The elements have the "correct" elastic stresses and detect yielding at their ends. For example, a cantilever loaded statically at its tip can be represented accurately in both elastic and plastic regimes using a single element. As with the Hughes-Liu beam, the mass is lumped onto the nodes, so finer meshes may be required for dynamic problems where a correct mass distribution is important.

You can use any of the following material models for beam elements (with some restrictions for certain element formulations):
2.2.2. LINK160

The LINK160 spar (truss) element is similar to the Belytschko-Schwer beam, but carries axial loads only. This element assumes a straight bar, axially loaded at its ends, and of uniform properties from end to end. The materials you can use for this element are isotropic elastic, plastic kinematic (rate dependent), and bilinear kinematic.

2.2.3. LINK167

The LINK167 element is a tension only spar that can be used to model cables. It is similar to a spring element in that the relationships between force and deformation are input directly by the user. This element requires that the cable option be specified with the EDMP command (see the EDMP command description).

2.3. Discrete Elements

2.3.1. COMBI165 Spring-Damper

Spring elements generate a force which depends on displacement; that is, change of length of the element. The force is applied along the element axis. For example, a positive force (tension) acts along the positive axis direction on node 1 and along the negative axis direction on node 2. By default, the element axis lies along the line from node 1 to node 2. When the element rotates, the line of action of the force also rotates.

Damper elements are treated as a subset of spring elements: both linear viscous and nonlinear viscous dampers can be modeled.

Rotational (torsional) springs and dampers are also available. These are selected by setting KEYOPT(1) = 1. The rest of the input is the same as for translational springs; the given force-displacement relationship will be treated as moment-rotation (rotation in radians), and the moments are applied about the element axis (positive clockwise). Rotational springs affect only the rotational degrees of freedom of their nodes - they do not pin the nodes together.

The COMBI165 element can be used in combination with any other explicit element. However, because it does not have mass, COMBI165 cannot be the only element type in an analysis. To represent a spring/mass system, you must add mass by also defining MASS166 elements.

You cannot define both spring and damper properties for the same COMBI165 element. However, you can define separate spring and damper elements that use the same nodes (that is, you can overlay two COMBI165 elements).

You can use any of the following material models for COMBI165:

• Linear Elastic Spring
• Linear Viscous Damper
• Elastoplastic Spring
• Nonlinear Elastic Spring
• Nonlinear Viscous Damper
• General Nonlinear Spring
• Maxwell Viscoelastic Spring
• Inelastic Tension or Compression-Only Spring

When using COMBI165, be sure to specify a unique set of real constants, element type, and material properties (R, ET, and TB commands, respectively) for each part to ensure that parts are uniquely defined.

2.3.2. MASS166

The mass element is defined by a single node and a mass value (Force*Time²/Length). Mass elements can often be used in a model to realistically characterize the mass of a structure without incorporating a large number of solid or shell elements. In an automobile impact analysis, for example, mass elements can model engine components whose deformation behavior is not of primary interest. Such a use of mass elements will reduce the number of elements required in an analysis, hence reducing the total computation time required to obtain a solution.

You can also use the MASS166 element to define a lumped rotary inertia at a node. To use this option, set KEYOPT(1) = 1 in the MASS166 element definition and input six moment of inertia values (IXX, IXY, IXZ, IYY, IYZ, IZZ) via the element real constants. You cannot input a mass value for this option; therefore, you must define a second mass element at the same node (with KEYOPT(1) = 0) to account for mass.

2.4. General Element Capabilities

The following element types can be declared as rigid bodies: LINK160, BEAM161, PLANE162, SHELL163, SOLID164, LINK167, and SOLID168. Rigid bodies are covered in detail in Chapter 8, Rigid Bodies (p. 105).

The mass of each solid, shell, and beam element is distributed equally on the nodes of the element. In the case of shell and beam elements, a rotational inertia is also added to each node; a single value is used, so that the effect is to distribute the mass spherically around the node.
Chapter 3: Analysis Procedure

The procedure for an explicit dynamics analysis consists of three main steps:

1. Build the model
2. Apply loads and obtain the solution
3. Review the results

The following analysis procedure topics are available:

3.1. Build the Model
3.2. Apply Loads and Obtain the Solution
3.3. Review the Results
3.4. The Definition of Part
3.5. Adaptive Meshing

3.1. Build the Model

The first step in an explicit dynamic analysis is to create the model that will represent the physical system to be analyzed. Use the PREP7 Preprocessor to build the model.

If you perform your analysis through the ANSYS graphical user interface (GUI), it is important to first set the Preferences options (Main Menu> Preferences) to “LS-DYNA Explicit” so that menus are properly filtered to show explicit dynamics input options. (Note that setting the preference to LS-DYNA explicit does not activate an LS-DYNA solution. To do so, you must specify an explicit element type such as SHELL163.)

Once you have set the analysis preference, you can build the model as you normally would for any analysis type. First, enter the preprocessor by issuing the /PREP7 command or by picking Main Menu> Preprocessor in the GUI. In the preprocessor, perform the following tasks:

- Define the element types and real constants
- Specify material models
- Define the model geometry
- Mesh the model
- Define contact surfaces

If you have never used an ANSYS product before, you should review the Basic Analysis Guide and the Modeling and Meshing Guide to learn the general procedures used to build a model in ANSYS.

3.1.1. Define Element Types and Real Constants

Element types available for an explicit dynamic analysis are described briefly in Chapter 2, Elements (p. 7). More detailed descriptions of each explicit element can be found in the Element Reference. It is recommended that you read these descriptions carefully before deciding which element types to use in your model.

Once you have chosen appropriate elements to represent your physical system, you specify element types using the ET command (or in the GUI pick Main Menu> Preprocessor> Element Type).
Each element description in the *Element Reference* lists all real constants that are associated with the element. You must determine which real constants (if any) are required for each element type you include in the model. Then specify the real constants using the R command (or in the GUI pick **Main Menu> Preprocessor> Real Constants**).

### 3.1.2. Specify Material Properties

There are numerous material models available for use in an explicit dynamic analysis. You should refer to the element descriptions in the *Element Reference* to find out which material models are valid for a particular element. Also, review Chapter 7, *Material Models* (p. 67) of this document, which contains detailed descriptions for all of the available material models.

Once you have decided which material model (or models) to use in your analysis, you must define all of the properties associated with the model (as described in Chapter 7, *Material Models* (p. 67)). To do so in batch or command mode, you typically use the MP, TB, and TBDATA commands (and in some cases, the EDMP command). Temperature dependent models may also require the MPTEMP, MPDATA, and TBTEMP commands. In the GUI, material models are defined via the following menu path: **Main Menu> Preprocessor> Material Props> Material Models**

After you select the appropriate material model, you will be prompted to define the necessary properties for that model. See *Defining Explicit Dynamics Material Models* (p. 68) for more information on defining material models through the GUI.

In order to specify an orthotropic model that is not aligned with the global Cartesian coordinate system, you must first define the local coordinate system with the EDLCS command (menu path: **Main Menu> Preprocessor> Material Props> Local CS> Create Local CS**.)

For some material models, you may also need to use the EDCURVE command to define data curves associated with the material (e.g., a stress-strain curve). (To access EDCURVE in the GUI, pick: **Main Menu> Preprocessor> Material Props> Curve Options**.)

### 3.1.3. Define the Model Geometry

The easiest way to create the model geometry is with the solid modeling capabilities of the ANSYS program. Refer to *Solid Modeling* in the *Modeling and Meshing Guide* for detailed information on all solid modeling functions.

For simple models (e.g., line elements only), you may choose to use the direct generation modeling method. By this method, you define the nodes and elements of the model directly. See *Direct Generation* in the *Modeling and Meshing Guide* for details.

### 3.1.4. Mesh the Model

After building the solid model, you are ready to mesh the model with nodes and elements. The procedure for generating the mesh is outlined in *Generating the Mesh* in the *Modeling and Meshing Guide*. If you are not familiar with ANSYS meshing, you should review that chapter before meshing an explicit dynamics model. Since all of the details are contained in that chapter, we will discuss only a few key points in this section.

Meshing involves three main steps:

- Set the element attributes
- Set mesh controls
3.1.5. Define Contact Surfaces

An explicit dynamic analysis often includes contact between surfaces. Chapter 6, Contact Surfaces (p. 55) of this document describes the types of contact available and the procedure to define contact. A brief overview of that procedure is given here.

Defining contact involves four basic steps:

- Determine the type of contact which best defines the physical model (EDCGEN command)
- Identify contact surfaces (CM, or EDPART, or EDASMP commands, with the EDCGEN command)
- Specify friction coefficient parameters (EDCGEN command)
- Specify additional input required for the chosen contact type (EDCGEN and EDCONTACT commands)

If automatic contact is not being used, you must define contact surfaces by grouping the nodes on each surface into a component with the CM command. Once these components are created, use the EDCGEN command to specify contact between the desired surfaces (i.e., node components). You must also identify
the type of contact with the **EDCGEN** command. For the single surface contact algorithms, the outer surfaces of the contacting bodies are determined by ANSYS LS-DYNA.

**Note**

Contact surfaces can alternatively be identified by a PART number (using **EDPART**), or by an assembly of parts (using **EDASMP**) rather than by a node component. Part and assembly definitions are discussed in *The Definition of Part* (p. 26).

The **EDCGEN** command is also used to specify friction coefficient parameters and other input specific to different contact types. Another command, **EDCONTACT** is used to specify miscellaneous contact controls such as contact penetration checking and sliding interface penalties. (Contact options are found in the GUI under **Main Menu> Preprocessor> LS-DYNA Options> Contact**.)

### 3.1.6. General Modeling Guidelines

The following are guidelines that you should consider when planning your explicit dynamics model:

- Use rigid bodies whenever possible to represent relatively stiff, unyielding objects in the model. Using rigid bodies simplifies the solution and results in shorter run times. (See *Chapter 8, Rigid Bodies* (p. 105) for information on defining rigid bodies.)
- Use realistic values for material properties. For example, do not use an unrealistically high elastic modulus to represent a rigid body, and do not use unrealistic thickness values for shell elements.
- Consider using damping (**EDDAMP** command) to prevent unrealistic oscillations in your model's structural response. Refer to the **EDDAMP** command description in the **Command Reference** for detailed information.
- If you have performed 2-D dynamic analyses with the regular ANSYS program, consider extruding these models to 3-D and analyzing them with ANSYS LS-DYNA. You may achieve more accurate results in a shorter run time.
- Note that the submodeling and substructuring features of the ANSYS program cannot be used in ANSYS LS-DYNA.

### 3.2. Apply Loads and Obtain the Solution

To apply loads and obtain a solution, you need to enter the ANSYS solution processor. Issue the **/SOLU** command or pick **Main Menu> Solution** in the GUI.

From the solution processor, you can apply loads, initial velocities, constraints, and DOF coupling to the model. The following sections describe this process.

#### 3.2.1. Loads

In an explicit dynamic analysis, all loads must be specified over time using component logic or part IDs, array parameters, and the **EDLOAD** command. (In the GUI, load options are found under **Main Menu> Solution> Loading Options> Specify Loads**.) Basic input for this command is a component name or part number and two array parameter names or a load curve ID number (LCID). The component specified must contain the nodes or elements on which the load is being applied. The array parameters specified must contain time varying load data (one array for time values and one array for the corresponding load values; the two arrays must be the same length). As an alternative to inputting the array parameters on the **EDLOAD** command,
you can define the load curve using the **EDCURVE** command ([Main Menu> Solution> Loading Options> Curve Options](#)) and input the load curve ID on **EDLOAD**. Valid loads are shown in the table below.

**Table 3.1  Loads Applicable in an Explicit Dynamics Analysis**

<table>
<thead>
<tr>
<th>Load Type</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Displacements</td>
<td>UX, UY, UZ</td>
</tr>
<tr>
<td>Rotations</td>
<td>ROTX, ROTY, ROTZ</td>
</tr>
<tr>
<td>Forces</td>
<td>FX, FY, FZ</td>
</tr>
<tr>
<td>Moments</td>
<td>MX, MY, MZ</td>
</tr>
<tr>
<td>Velocities</td>
<td>VX, VY, VZ</td>
</tr>
<tr>
<td>Accelerations (on nodes)</td>
<td>AX, AY, AZ</td>
</tr>
<tr>
<td>Base Accelerations</td>
<td>ACLX, ACLY, ACLZ</td>
</tr>
<tr>
<td>Angular Velocities</td>
<td>OMGX, OMGY, OMGZ</td>
</tr>
<tr>
<td>Temperature</td>
<td>TEMP</td>
</tr>
<tr>
<td>Pressures (applied to elements)</td>
<td>PRESS</td>
</tr>
<tr>
<td>Displacements on Rigid Bodies</td>
<td>RBUX, RBUY, RBUZ</td>
</tr>
<tr>
<td>Rotations on Rigid Bodies</td>
<td>RBRX, RRY, RBHZ</td>
</tr>
<tr>
<td>Velocities on Rigid Bodies</td>
<td>RBVX, RBVY, RBVZ</td>
</tr>
<tr>
<td>Forces on Rigid Bodies</td>
<td>RBFX, RBFY, RBFZ</td>
</tr>
<tr>
<td>Moments on Rigid Bodies</td>
<td>RBMX, RBMY, RBMZ</td>
</tr>
<tr>
<td>Angular Velocities on Rigid Bodies</td>
<td>RBOX, RBOY, RBOZ</td>
</tr>
</tbody>
</table>

**Note**

We recommend that you specify velocity time histories instead of displacement time histories. A piecewise linear displacement time history may lead to discontinuous velocities and infinite accelerations.

The load symbol will appear automatically on the active graphics window. The load symbol is erased automatically when you replot. To turn the display of this symbol on or off, issue **EDFPLOT** (or pick [Main Menu> Solution> Loading Options> Show Forces](#)) in the GUI).

To visualize the applied load curve, use the **EDPL** command (or pick [Main Menu> Solution> Loading Options> Plot Load Curve](#)) in the GUI).

For more detailed information on applying loads with the **EDLOAD** command, see *General Loading Options* (p. 33).

### 3.2.2. Initial Velocities

In the ANSYS LS-DYNA program, initial velocities are defined using the **EDVEL** and **EDPVEL** commands. You can use these commands to apply both translational and rotational velocities to various entities. Use **EDVEL** to apply velocities to a nodal component or to a single node; use **EDPVEL** to apply velocities to a part or part assembly.
The **EDVEL** and **EDPVEL** commands provide two methods for specifying rotational velocities, **Option = VGEN** and **Option = VELO**. The VGEN method applies a rigid body rotation to the specified entity around a specified axis. The VELO method applies the rotational velocities directly to each node’s rotation degree of freedom. Since only shell and beam elements have rotation degrees of freedom, the rotations input with the VELO method are only applicable to **SHELL163** and **BEAM161** elements.

You can access all the initial velocity options in the GUI by picking **Main Menu > Solution > Initial Velocity**. For more information on how to apply initial velocities, see *Initial Velocity* (p. 42).

### 3.2.3. Constraints

In addition to loads and initial velocities, constraints can also be applied to the model. Using the **D** command, you can apply constraints only to the displacement (UX, UY, UZ) and rotation (ROTX, ROTY, ROTZ) degrees of freedom; the constraint value must be zero. (In the GUI, pick **Main Menu > Solution > Constraints > Apply**.)

Other specialized types of constraints are also available in an explicit dynamic analysis (such as symmetry boundary planes and welded constraints). See *Constraints* (p. 40) and *Welds* (p. 41) for more information.

### 3.2.4. DOF Coupling

DOF coupling (**CP** command) and constraint equations (**CE** command) are also allowed in an explicit dynamic analysis. (In the GUI, pick **Main Menu > Preprocessor > Coupling/Ceqn**.) Coupling is allowed only for the UX, UY, and UZ degrees of freedom. Constraint equations are allowed only for the UX, UY, UZ, and ROTX, ROTY, ROTZ degrees of freedom. For more information, see *Coupling and Constraint Equations* (p. 43).

### 3.2.5. Data Smoothing

If you’re working with noisy data (such as an earthquake excitation), you may want to "smooth" that data to a set of data that provides an accurate approximation of the data points.

To smooth data, you must first create four arrays. The first two arrays should contain the noisy data from the independent and the dependent variables. The second two arrays will be used to store the smoothed data. After these arrays are created, you then smooth the data using the **EDNDS** command (**Main Menu > Solution > Loading Options > Smooth Data**). See *Data Smoothing* (p. 121) later in this document for more information on data smoothing.

### 3.2.6. Specify Explicit Dynamics Controls

*Table 3.2: LS-DYNA Solution and Output Control Options* (p. 24) shows some basic LS-DYNA solution and output controls that you should specify for an explicit dynamic analysis.

<table>
<thead>
<tr>
<th>Option</th>
<th>Command</th>
<th>GUI Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Termination Time</td>
<td><strong>TIME</strong></td>
<td><strong>Main Menu &gt; Solution &gt; Time Controls &gt; Solution Time</strong></td>
</tr>
<tr>
<td>SHELL/BEAM Output</td>
<td><strong>EDINT</strong></td>
<td><strong>Main Menu &gt; Solution &gt; Output Controls &gt; Integ Pt Storage</strong></td>
</tr>
<tr>
<td>Substep Output Controls</td>
<td><strong>EDRST</strong></td>
<td><strong>Main Menu &gt; Solution &gt; Output Controls &gt; File Output Freq</strong></td>
</tr>
<tr>
<td>Output Interval (History File)</td>
<td><strong>EDH-TIME</strong></td>
<td><strong>Main Menu &gt; Solution &gt; Output Controls &gt; File Output Freq</strong></td>
</tr>
</tbody>
</table>
3.2.7. Save Database and Solve

After you have specified all loading and solution and output controls, save a backup copy of the database to a named file (SAVE command or Utility Menu > File > Save as in the GUI). Next, start the solution calculations (SOLVE command or Main Menu > Solution > Solve in the GUI).

For more details on explicit dynamic solution controls and other solution features, see Chapter 5, Solution Features (p. 47) later in this document.

3.3. Review the Results

You can review explicit dynamic analysis results using POST1, the general postprocessor, and POST26, the time-history processor.

- POST1 is used to review results over the entire model at specific time-points.
- POST26 is used to track specific nodal and element result items over a more detailed load history.

The following are basic guidelines for postprocessing explicit dynamic results.

- The database must contain the same model for which the solution was calculated.
- The results file (Jobname.RST) must be available for POST1.
- The history file (Jobname.HIS) must be available for POST26.
- All stresses and strains output from LS-DYNA are in the global Cartesian coordinate system. However, if you are using composite materials, stresses can be in a local (element) coordinate system.

---

### Option | Command | GUI Path
---|---|---
Energy Options | EDEN-ERGY | Main Menu> Solution> Analysis Options> Energy Options

The above table does not show all controls available in an explicit dynamic analysis. However, most of the default settings for the LS-DYNA control options (output controls, file controls, damping options, etc.) are sufficient for most explicit dynamic analyses and need not be modified.

A brief description of those options that are recommended follows:

- **Termination Time [TIME]**
  
  This option specifies time at the end of the analysis.

- **Substep Output Controls [EDRST]**
  
  This option specifies the number of results written to the Jobname.RST file. Because explicit dynamics analyses are only solved over very small time increments (i.e., 1e-7 seconds), only a relatively small number of solutions should be written to the Jobname.RST file.

- **Output Interval [EDHTIME]**
  
  This option specifies the number of output steps for the history file (Jobname.HIS). The history file results are typically saved for a small subset of nodes or elements [EDHIST], but at a much higher frequency than the results file (Jobname.RST) results.
For more information on postprocessing explicit dynamic results, see Chapter 12, Postprocessing (p. 115) later in this document. For a complete description of all postprocessing functions, see "An Overview of Postprocessing" in the Basic Analysis Guide.

### 3.4. The Definition of Part

A group of elements having a unique combination of MAT, TYPE, and REAL set IDs is designated as a part and is assigned a PART ID. There are a number of ANSYS LS-DYNA commands which refer to the PART ID directly (for example, EDCGEN, EDDC, EDLOAD, EDDAMP, EDCRB, and EDREAD). The PART IDs are automatically created and written to the LS-DYNA input file Jobname.K at the start of the model solution by the ANSYS LS-DYNA program. They can also be created and listed with the EDPART command.

When the program generates PART IDs, the IDs are created from a sequential list of selected elements. If any of the MAT, TYPE, or REAL IDs change in the element list, the next PART ID number will be assigned to that group of elements. For example, if the first ten elements have MAT = 3, TYPE = 1, and REAL = 2 then these ten elements will be assigned part number 1; if the next one hundred elements have MAT = 2, TYPE = 2, and REAL = 1, these elements will be assigned part number 2; and so on.

The part list can be created and listed using the CREATE/UPDATE/ADD/LIST options on the EDPART command. It is also created and remains permanent if the SOLVE command is issued or if the EDWRITE command is used to write the input file, Jobname.k. If the part list already exists, these latter commands will only update the list, not overwrite it. For more information on the EDPART command, refer to the description of this command in the Command Reference.

You can assign a specific PART ID to a group of elements by using the ADD option on EDPART. To do so, you must first select a group of elements which have the same combination of MAT, TYPE, and REAL IDs (ESEL command). You can name those elements as a component, or assign the PART ID to the currently selected set. You can then use the EDPART,ADD option to specify an arbitrary number to be used as the PART ID for those elements. It is important to note that user-specified PART IDs will not be overwritten by subsequent EDPART,CREATE or EDPART,UPDATE commands. However issuing EDPART,ADD with a different PART ID will overwrite existing PART IDs for the specified elements. To delete a user-specified PART ID completely, issue the EDPART,DELE,PARTID command. The deleted PART ID is immediately overwritten by a program-generated PART ID (similar to EDPART,UPDATE).

EDPART,CREATE creates new PART IDs, but does not overwrite any user-specified PART IDs. These PART IDs can be listed using the EDPART,LIST command. The list shows the status of parts at the time of creation or update (EDPART,UPDATE). If the EDPART,CREATE command is issued repeatedly, the part list is overwritten. In order to get the actual part list after modifications or additions to the model, issue the EDPART,UPDATE command. This extends the existing part list without changing its order and allows elements to be added to an existing part comprised of the same MAT, TYPE, and REAL IDs. Any combination of MAT, TYPE, and REAL which is not referred to by any selected element will render the corresponding part unused. This will be evident from the zero value printed on the fifth (USED) column of the part list. A warning is issued at the SOLVE or EDWRITE command if a previously defined part-related command refers to an unused part.

Any combination of user-specified and automatically generated PART IDs can be used within a model. For example, you may choose to input PART IDs (using the ADD option) for one or more groups of elements and then use EDPART,CREATE or EDPART,UPDATE to define PART IDs for the rest of the model.

The following example shows a 14-element model with 2 MATs, 3 TYPEs and 3 REALs. In this model, element 10 originally had MAT = TYPE = REAL = 2, but was deleted after the part list was automatically created by issuing EDPART,CREATE. An ELIST command produces the following element list:
To define a user-specified PART ID, first select elements 3 and 5 and create an element component E35. **EDPART**,ADD,10,E35 specifies PART ID 10 for elements 3 and 5. PART ID 10 (for the component E35) will not change if you issue **EDPART**,CREATE or **EDPART**,UPDATE after reselecting all of the elements in the model.

Consider the definition of a second user-specified PART ID 60 for element 14. Selecting element 14 and issuing **EDPART**,ADD,60 changes the PART ID for element 14 from 6 to 60. After selecting all of the elements, **EDPART**,UPDATE shows the following part list.

<table>
<thead>
<tr>
<th>PART</th>
<th>MAT</th>
<th>TYP</th>
<th>REAL</th>
<th>USED</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>60</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

To define a user-specified PART ID, first select elements 3 and 5 and create an element component E35. **EDPART**,ADD,10,E35 specifies PART ID 10 for elements 3 and 5. PART ID 10 (for the component E35) will not change if you issue **EDPART**,CREATE or **EDPART**,UPDATE after reselecting all of the elements in the model.

Consider the definition of a second user-specified PART ID 60 for element 14. Selecting element 14 and issuing **EDPART**,ADD,60 changes the PART ID for element 14 from 6 to 60. After selecting all of the elements, **EDPART**,UPDATE shows the following part list.

<table>
<thead>
<tr>
<th>PART</th>
<th>MAT</th>
<th>TYP</th>
<th>REAL</th>
<th>USED</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>60</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

In the above list, elements 1,2 and 4 have MAT = TYPE = REAL = 1 and form PART 1. Elements 3 and 5 also have MAT = TYPE = REAL = 1 but have the user-specified PART ID 10. Element 14 with MAT = 1, TYPE = 3 and REAL = 2 has the user-specified PART ID 60. The PART IDs for the rest of elements were generated by the ANSYS LS-DYNA program. Element 10 had MAT = TYPE = REAL = 2, but was deleted after the part list was created initially. Therefore, USED for PART ID 4 is zero. The USED field in the part list indicates the number of elements having the PART ID. USED is calculated from all currently selected elements. If the element selection changes, USED will change. To get the correct number of USED, select the correct number of elements and issue **EDPART**,UPDATE.
Note

For this example, issuing \texttt{EDPART,UPDATE} will not change the PART IDs in the part list. However, issuing \texttt{EDPART,CREATE} will create a new part list. PART 5 will become the new PART 4 and PART 6 will become the new PART 5. The user-specified PART IDs 10 and 60 will not change. Because the automatically generated PART IDs can change as a result of \texttt{EDPART,CREATE}, this can invalidate previously defined part-based loading, contact specifications, etc.

The following procedure is recommended:

a. Build the model as usual until you have to specify a PART ID number on a part-related command.

b. Create the part list (\texttt{EDPART,CREATE}) or specify PART IDs for some groups of elements (\texttt{EDPART,ADD}); then list the parts (\texttt{EDPART,LIST}).

c. Use the appropriate PART ID number(s) from the list for the command(s).

d. Continue modeling.

e. Update the part list (\texttt{EDPART,UPDATE}) if the elements or their attributes have changed, which includes the addition of new elements.

f. List the updated PART ID numbers and make use of them on further part-related command(s).

Issuing \texttt{ELIST} with the part listing key on (\texttt{PTKEY = 1}) will show the PART ID for each elements as follows.

<table>
<thead>
<tr>
<th>ELEM</th>
<th>PART</th>
<th>MAT</th>
<th>TYP</th>
<th>REL</th>
<th>ESY</th>
<th>SEC</th>
<th>NODES</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>21</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>31</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>41</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>51</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>61</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>71</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>81</td>
</tr>
<tr>
<td>11</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>101</td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>111</td>
</tr>
<tr>
<td>13</td>
<td>6</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>121</td>
</tr>
<tr>
<td>14</td>
<td>60</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>131</td>
</tr>
<tr>
<td>15</td>
<td>6</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>141</td>
</tr>
</tbody>
</table>

In addition to listing PART IDs in the element list, PART IDs can also be plotted with the \texttt{ELEPLOT} command by turning on the part display key (\texttt{/PNUM,PART,1}). Element selection using the \texttt{ESEL} command can also be performed based on PART IDs. These two capabilities replace the obsolete \texttt{PARTSEL} command.

The \texttt{CDWRITE} command writes the part information to the \texttt{Jobname.CDB} file. This information can then be automatically read into ANSYS with the \texttt{CDREAD} command. However, if more than one \texttt{Jobname.CDB} file is read, the part list from the last \texttt{Jobname.CDB} file overwrites the existing part list, if any. In general, you must issue the \texttt{EDPART,CREATE} command to recreate the part list of the total model. This will affect all
part-related commands contained in the Jobname.CDB file. Therefore, you can join models, but not part-related inputs, which must be modified using the newly-created PART ID numbers.

In limited cases, an update of the part list (EDPART,UPDATE) is possible. This requires that no used combination of MAT/TYP/REAL appears more than once in the list. However, partial changes to the part-related commands may be necessary.

### 3.4.1. Part Assemblies

Part assemblies, which are created using the EDASMP command, are entities that are made up of several different parts. Part assemblies can be used as input on several ANSYS LS-DYNA commands. They can be particularly useful when defining contact between two entities consisting of several parts (see Chapter 6, Contact Surfaces (p. 55) in the ANSYS LS-DYNA User’s Guide). Using the EDASMP command, you can define as many as 16 parts in a part assembly after you provide an ID number. The part assembly number can be any arbitrary number, but cannot be a currently assigned PART ID number (use EDPART,LIST to determine the currently used numbers). You can also list and delete part assemblies using the EDASMP,LIST and EDASMP,DELE options, respectively. As an example, to specify part assembly ID number 10 consisting of parts 1,2 and 4, the following command would be issued:

```
EDASMP,ADD,10,1,2,4
```

### 3.5. Adaptive Meshing

In metal forming and high-speed impact analyses, a body may experience very large amounts of plastic deformation. Single point integration explicit elements, which are usually robust for large deformations, may give inaccurate results in these situations due to inadequate element aspect ratios. To counteract this problem, the ANSYS LS-DYNA program has the ability to automatically remesh a surface during an analysis to improve its integrity. This capability, known as adaptive meshing, is controlled with the EDADAPT and EDCADAPT commands.

The EDADAPT command activates adaptive meshing within a specific PART ID. (Use the EDPART command to create and list valid PART IDs.) For example, to turn on adaptive meshing for PART ID #1, you would issue the following command:

```
EDADAPT,1,ON
```

**Note**

Adaptive meshing is only valid for parts consisting of SHELL163 elements.

When adaptivity is turned on for a part, the mesh for that part will be automatically regenerated during an analysis to ensure adequate element aspect ratios are maintained throughout the deformation process. Adaptive meshing is most commonly used in the analysis of large deformation processes such as metal forming (adaptive meshing is typically used for the blank). To activate adaptive meshing for more than one part within a model, you must reissue the EDADAPT command for each different PART ID. By default, adaptive meshing is off for all parts within a model.

After specifying which parts will be adaptively meshed, you must also define adaptive meshing parameters using the EDCADAPT command. The EDCADAPT command globally sets the control options for all PART IDs that are to be adaptively meshed as defined by the EDADAPT command. The parameters controlled by the EDCADAPT command are listed below.

- Frequency (**FREQ**) - The time interval between which adaptive mesh refinements will take place. For example, if **FREQ** is set to 0.01, elements will be adaptively remeshed every .01 seconds (assuming the
time unit used is seconds) if they are above the specified angle tolerance. Since the default value for the \( \text{FREQ} \) field is 0.0, this field must be specified when using adaptive meshing in an analysis.

- **Angle Tolerance (\( \text{TOL} \))** - The adaptive angle tolerance (in degrees) for which adaptive meshing will occur (default = 1e31). The \( \text{TOL} \) field controls the aspect ratio between elements and is very important for ensuring accurate results. If the relative angle between elements exceeds the specified \( \text{TOL} \) value, the elements will be automatically refined.

- **Adaptivity Option (\( \text{OPT} \))** - There are two different angle options available for mesh adaptivity. For \( \text{OPT} = 1 \), the angle change that is compared to the specified \( \text{TOL} \) value is computed based on the original mesh configuration. For \( \text{OPT} = 2 \), the angle change that is compared to the specified \( \text{TOL} \) value is computed based on the previously refined mesh.

- **Mesh Refinement Levels (\( \text{MAXLVL} \))** - The \( \text{MAXLVL} \) field controls the number of times an element can be remeshed during the entire analysis. For a single original element, a \( \text{MAXLVL} \) of 1 would allow the creation of one additional element, a \( \text{MAXLVL} \) of 2 would allow the creation of up to four additional elements, and a \( \text{MAXLVL} \) of 3 would allow the creation of up to 16 additional elements. High values of \( \text{MAXLVL} \) will yield more accurate results, but can dramatically increase the model size.

- **Remeshing Birth and Death Times (\( \text{BTIME} \) and \( \text{DTIME} \))** - The remeshing birth and death times control when adaptive meshing is turned on and off in an analysis. For example, if you set \( \text{BTIME} = .01 \) and \( \text{DTIME} = .1 \), adaptive meshing will only take place in the analysis between .01 and .1 seconds (assuming the time unit used is seconds).

- **Interval of Remeshing Curve (\( \text{LCID} \))** - This data curve (specified on the \text{EDCURVE} command) defines the interval of remeshing as a function of time. The abscissa of the data curve is time, and the ordinate is the varied adaptive time interval. If this option is nonzero, the adaptive frequency (\( \text{FREQ} \)) will be replaced by this data curve. Note, however, that a nonzero \( \text{FREQ} \) value is still required to initiate the first adaptive loop.

- **Minimum Element Size (\( \text{ADPSIZE} \))** - Minimum element size to be adapted based on element edge length. If this parameter is undefined, the edge length limit is ignored.

- **One or Two Pass Option (\( \text{ADPASS} \))** - If \( \text{ADPASS} = 0 \), two pass adaptivity is used, and the calculation is repeated after adaptive remeshing (this is the default). If \( \text{ADPASS} = 1 \), one pass adaptivity is used, and the calculation is not repeated after adaptive remeshing. See Figures 30.9(a) and 30.9(b) in the \text{LS-DYNA Theoretical Manual} for graphical representations of these two options.

- **Uniform Refinement Level Flag (\( \text{IREFLG} \))** - Values of 1, 2, 3, etc. allow 4, 16, 64, etc. elements, respectively, to be created uniformly for each original element.

- **Penetration Flag (\( \text{ADPENE} \))** - Depending on whether the value of \( \text{ADPENE} \) is positive (approach) or negative (penetrate), the program will adapt the mesh when the contact surfaces approach or penetrate the tooling surface by this value. The tooling adaptive refinement is based on the curvature of the tooling. If \( \text{ADPENE} \) is positive, the refinement generally occurs before contact takes place; consequently, it is possible to use one pass adaptivity (\( \text{ADPASS} = 1 \)).

- **Shell Thickness Level (\( \text{ADPTH} \))** - Absolute shell thickness level below which adaptivity should begin. This option works only if the adaptive angle tolerance (\( \text{TOL} \)) is nonzero. If thickness based adaptive remeshing is desired without angle change, set \( \text{TOL} \) to a large angle. (If \( \text{ADPTH} = 0.0 \), this option is not used.)

- **Maximum Element Limit (\( \text{MAXEL} \))** - Maximum number of elements at which adaptivity will be terminated. Adaptivity is stopped if this number of elements is exceeded.

For most problems, you should do an initial analysis without adaptive meshing. If the results of the initial analysis show a highly distorted mesh and if the solution appears to be incorrect, then you can rerun the analysis with adaptive meshing turned on. Adaptive meshing may also be used when an analysis terminates in \text{LS-DYNA} with a “negative volume element” error.
When adaptive meshing is turned on, the number of elements within the model will change during the solution. After each adaptive loop, the mesh will be updated and a new results file will be written with the extension RSnn, where nn is the adaptive mesh level. (An adaptive loop will occur at every time increment specified by FREQ or at the time intervals specified by LCID.) For example, a model that is adaptively remeshed two times will produce two results files, Jobname.RS01 and Jobname.RS02. For details on postprocessing adaptively meshed results, see Postprocessing after Adaptive Meshing (p. 118).

Note

A new results file will be created for each loop, even if the mesh does not change.

A new time history file having the extension Hlnn is also written each loop; however, these files may not be as useful for postprocessing as the RSnn files. In addition, LS-DYNA creates a set of adaptive meshing files with the root name "adapt". Therefore, you should not use "adapt" as your jobname when adaptive meshing is activated.
Chapter 4: Loading

After building the model, the next step is to apply loads to the structure in preparation for solution. In order to properly model your structure's behavior, it is necessary to apply loads with respect to a specified time interval. This chapter discusses the following topics related to loads:

- General loading options
  - The use of components or part IDs and array parameters
  - How to apply, delete, and list general loads [EDLOAD]
  - How to plot the explicit dynamics load curve [EDPL]
  - How to define a data curve [EDCURVE]
  - How to show and hide load symbols
- Constraints and initial conditions
  - How constraints [D, EDNROT] are used in ANSYS LS-DYNA
  - How to define sliding and cyclic symmetry planes [EDBOUND]
  - How to define other miscellaneous types of constraints [EDCNSTR]
  - How to define welds [EDWELD]
  - How to apply initial velocities [EDVEL, EDPVEL] to your model
- Coupling and constraint equations
  - Coupling degrees of freedom [CP]
  - Constraint equations between degrees of freedom [CE]
- Nonreflecting boundaries [EDNB]
- Temperature loading
- Dynamic relaxation

The following loading topics are available:
4.1. General Loading Options
4.2. Constraints and Initial Conditions
4.3. Coupling and Constraint Equations
4.4. Nonreflecting Boundaries
4.5. Temperature Loading
4.6. Dynamic Relaxation

4.1. General Loading Options

Unlike most implicit analyses, all loads in an explicit analysis must be time-dependent in nature. Hence, when using ANSYS LS-DYNA, many of the standard ANSYS commands are not valid. In particular, the F, SF, and BF family of commands are not applicable in ANSYS LS-DYNA because they can only be used to specify time-independent loads. Additionally, the D command can only be used to define constrained nodes. For
this reason, all loads in ANSYS LS-DYNA are applied using a pair of array parameters, one corresponding to the time and the other corresponding to the loading condition.

Note

Although nodal accelerations (AX, AY, AZ) and nodal velocities (VX, VY, VX) appear as degrees of freedom, they are not physical DOFs and cannot be constrained with the D command. To apply these nodal loads, use the EDLOAD command.

In ANSYS LS-DYNA, all loads are applied in one load step. This is much different from an implicit analysis where loads are often applied in multiple steps. In ANSYS LS-DYNA, for certain kinds of loads, you can also specify when a load is imposed on a body (birth time), and when the load is removed (death time) using the EDLOAD command. Please refer to Birth Time, Death Time, and CiD Support in the EDLOAD command to check the applicability of the birth/death time.

To apply a load to your model, you will need to follow these steps:

- Designate portions of the model that will receive the load as components (or parts, for rigid bodies)
- Define array parameters containing time intervals and load data values
- Specify load curves
- Define the load direction using the EDLCS command if the load is not acting in the global coordinate system
- Apply loads to the model

4.1.1. Components

With the exception of loads on rigid bodies, all loads are applied to components in an explicit analysis. Therefore, the first step in applying loads is to gather portions of your model into nodal or element groups called components. Each component should consist of those portions of the model that will receive the same loads and that are related to each other through such things as material properties, position in the model, expected behavior, etc.

For example, if you want to analyze a baseball being hit into a wall, you would probably define the nodes of the ball as one component, the nodes on the bat as another component, and the nodes on the wall as a third component.

You may define as many components as you wish, and then apply loads to each of them. The components must consist of either nodes or elements. (Components consisting of elements are only used when applying pressure loads.) To define a component, you must first select only those entities that you want to be included in the component. Then use the CM command or the following menu path in the GUI to define the component: Utility Menu> Select> Comp/Assembly> Create Component.

See Selecting and Components in the Basic Analysis Guide for more information on components.

For the case of rigid bodies, loads are applied to a specific part number rather than to components. This is due to the fact that rigid bodies already consist of a specific set of nodes and elements as defined by the command EDMP,RIGID,MAT.
4.1.2. Array Parameters

It is important to remember that in an explicit dynamic analysis the loading is applied to the structure over a specified time interval. This is done so that you may observe the model’s transient behavior for applied loads over the specific time period. Therefore, you will not only define what type of load you would like to apply to your model (FX, FY, FZ, ROTX, ROTY, ROTZ, etc.), but also the time history of the load for the time interval of interest.

Time intervals and their corresponding load values are grouped together and defined as array parameters. These array parameters should be defined in sets of two, with each entry of the first array specifying a time value, and each entry of the second array specifying a load value. The load entries in the second array correspond to the time values specified in the first array.

In the GUI, use this menu path to define an array parameter: Utility Menu > Parameters > Array Parameters > Define/Edit.

See Array Parameters in the ANSYS Parametric Design Language Guide for more information on how to define array parameters.

Note

Load values for intermediate time points are obtained by linear interpolation. However, load values outside of the specified time range are not extrapolated by the program. Therefore, you should ensure that the load time range is at least equal to the solution time. Otherwise, the results near the solution end time may be invalid due to premature load removal.

Once you define a set of array parameters that represent a time dependent load, you can apply the load by inputting the array parameters directly on the EDLOAD command. Or, you may choose to input the array parameters on the EDCURVE command to define a load curve. The corresponding load curve ID (LCID) can then be input on the EDLOAD command.

To illustrate the use of array parameters, consider the baseball example mentioned previously. Suppose that you wish to examine the distortion of the ball from the moment of impact with the bat to one second later. Assume that the displacement of the bat handle is known as a function of time and that the ball is initially traveling at 1600 in/sec (91 MPH) just prior to impact.

You will first need to define some nodal components that will be used to define the loading and contact surfaces. Create a component named ball that contains all of the nodes of the ball, for which an initial velocity (EDVEL) of 1600 in/sec will be applied (discussed later in this chapter). Then create a second component named ballsurf which contains only the nodes on the surface of the ball. This component will be used later in the contact specification. You also need to define a third component named batsurf, which contains the nodes on the surface of the bat. Contact algorithms are discussed in Chapter 6, Contact Surfaces (p. 55).

nselect,s,node,...       ! select all nodes comprising the ball
component,ball,node       ! define the component ball
nselect,s,node,...        ! select the nodes on the ball surface
component,ballsurf,node    ! define the component ballsurf
nselect,s,node,...        ! select the nodes on the bat surface
component,batsurf,node     ! define the component batsurf
nselect,all

Now select the nodes at the base of the bat handle (bathand) for which a displacement versus time loading curve will be applied.
Define an array named \textit{time} to store the time values. Remember to use time values that are consistent with all loads, dimensions, and material properties in your model. Next, define an array named \textit{xdisp} to contain the corresponding X displacements of the component \textit{bathand}. Likewise, define arrays \textit{ydisp} and \textit{zdisp} to contain the corresponding Y and Z displacements of \textit{bathand}.

\begin{verbatim}
nsel,a,node,... ! select the nodes at the bat handle base
cm,bathand,node ! define the component bathand
nsel,all
*dim,time,,4     ! dimension the array parameter time
*dim,xdisp,,4     ! dimension the array parameter xdisp
*dim,ydisp,,4     ! dimension the array parameter ydisp
*dim,zdisp,,4     ! dimension the array parameter zdisp
time(1)=0,0.25,0.5,0.75,1   ! times at specified displacements
xdisp(1)=0,-1,-2,-1,3      ! X displacement of the bat handle
ydisp(1)=0,1,2,3,4        ! Y displacement of the bat handle
zdisp(1)=0,3,6,8,9        ! Z displacement of the bat handle
\end{verbatim}

The example given is actually a simple version of a more complex phenomenon. In a more accurate simulation, additional displacement locations (and corresponding load curves) should be defined to better simulate the true motion of the bat handle. Further, the initial velocities of the nodes comprising the bat are all different. Finally, the ball is a composite of several different materials and material models.

\subsection*{4.1.3. Applying Loads}

Once you have defined the components and array parameters, you are ready to apply loads to your model \texttt{[EDLOAD command]}. In the GUI choose the following menu path: \texttt{Main Menu> Solution> Loading Options> Specify Loads}.

You may now choose to add loads (via the \texttt{ADD} label on \texttt{EDLOAD}) such as forces, moments, nodal accelerations, surface pressures, etc. Loads applied with the \texttt{EDLOAD} command are in the global Cartesian direction by default.

The dialog box in the GUI will give you a listing of all valid loads, as well as the components and parameters that have been previously defined. You simply choose the desired load label along with the combination of components (part numbers for rigid bodies) and array parameters (or previously defined load curve ID) you wish to use for that load. It is important to note, as listed below, that all load labels are not valid for all components or part numbers.

The following load labels are valid only for nodal components:

- Forces: FX, FY, FZ
- Moments: MX, MY, MZ
- Displacements: UX, UY, UZ
- Rotations: ROTX, ROTY, ROTZ
- Velocities: VX, VY, VZ
- Nodal Accelerations: AX, AY, AZ
- Body Accelerations: ACLX, ACLY, ACLZ
- Angular Velocities: OMGX, OMGY, OMGZ
- Temperature: TEMP

\textbf{Note}

Although V (X, Y, Z) and A (X, Y, Z) appear as DOFs, they are not actually physical DOFs. However, these quantities are computed as DOF solutions and stored for postprocessing.

The following labels are valid only for rigid bodies (part numbers):
Forces: RBFX, RBFY, RBFZ
Moments: RBMX, RBMY, RBMZ
Displacements: RBUX, RBUY, RBUZ
Rotations: RBRX, RBRY, RBRZ
Velocities: RBVX, RBVY, RBVZ
Angular Velocities: RBOX, RBOY, RBOZ

The following label is valid only for element components:

Pressure: PRESS

Returning to our baseball example, the necessary displacement versus time load curves are automatically created with the EDLOAD command from the time and x/y/zdisp array parameters.

```
edload,add,ux,,bathand,time,xdisp   ! X displacement of bathand
edload,add,uy,,bathand,time,ydisp   ! Y displacement of bathand
edload,add,uz,,bathand,time,zdisp   ! Z displacement of bathand
```

Alternatively, the very stiff bat could have been modeled as a rigid body to simplify the required input and to reduce the CPU time. For this case, the appropriate rigid body loads would be applied to valid part numbers (and not to nodal components).

```
edload,add,rbux,,2,time,xdisp    ! X displ. (if bat = part #2)
edload,add,rbuy,,2,time,ydisp    ! Y displ. (if bat = part #2)
edload,add,rbuz,,2,time,zdisp    ! Z displ. (if bat = part #2)
```

If the pressure that the bat imparts on the baseball was known as a function of time, the loading could be accomplished without defining any contact surfaces. Instead, the element component containing the elements on the surface of the ball would be loaded with the “PRESS” label of the EDLOAD command.

```
edload,add,press,1,cover,battime,batload
```

In this last scenario, the elements contained in the element component cover are loaded on face number one (face number is input in the KEY field) with the load curve produced by the battime and batload array parameters.

It is important to note that in the examples listed above, the pressure load uses the KEY field of the EDLOAD command to specify a load key. Load keys (1,2,3, etc.) are associated with surface loads and are listed under "surface loads" in the input data tables for each element type in the Element Reference. For most load types other than pressure loads, you can use the KEY field to specify a coordinate system identification number, CID, from the EDLCS command. The load will act in a direction you define on EDLCS, or in the global coordinate system direction if CID is not specified. See Defining Loads in a Local Coordinate System (p. 39) for more information.

**Note**

To avoid timing problems on some platforms, it is a good practice to always add a very small time value (such as $1.0 \times 10^{-6}$) to the value in the final item in the time array. For example, instead of the value 3.0, such an array might contain the following value for the last item:

```
timeint(1)=0,1,2,3.00001
```

The addition of this very small “padding” factor does not affect the accuracy of the results.

In addition to adding loads, you can also list and delete loads with the commands EDLOAD,LIST and EDLOAD,DELE. You can display load symbols on the element plot with the EDFPLOT command, and you can plot load curves with the EDPL command.
4.1.4. Data Curves

Data curves, which are defined with the **EDCURVE** command, have various uses in ANSYS LS-DYNA. They can be used to define material data curves (for example, stress-strain) and load data curves (force-deflection) associated with explicit dynamics material models. Data curves can also be used to define load curves that represent time dependent loads (force, displacement, velocity, etc.). These load curves can then be input on the **EDLOAD** command.

### 4.1.4.1. Using Data Curves with Material Models

Certain material models (for example, **TB**, **PLAW** or **TB,HONEY**) require specification of material property data which may be a function of effective strain rate, plastic strain, or volumetric strain. For such data, property curves need to be defined using the **EDCURVE** command prior to specifying the material behavior with the data table **[TBDATA]** command. Data curves are also used for defining force deflection behavior in rigid-body and drawbead contact.

Similar to placing loads on components, data curves are grouped into array parameters and then associated with a particular curve reference number (LCID) which can be used by a specified material model (PLAW, HONEY, etc.) or contact type (RNTR, ROTR) and drawbead. Use the following steps to specify data curves:

1. Define an array parameter that contains the abscissa values for the material or friction force behavior (e.g., effective plastic strain, effective strain rate, displacement, etc.).
2. Define a second array parameter that contains the ordinate values for the material behavior or friction force (e.g., initial yield stress, elastic modulus, force, etc.).
3. Specify the data curve that will be used to identify this data [EDCURVE]. During this step you will choose a data curve ID number that will be used to associate this data with a particular material behavior when setting up the data table **[TBDATA]**.

After the parameters have been specified, use one of the following menu paths in the GUI to define a data curve:

- **Main Menu> Preprocessor> Material Props> Curve Options**
- **Main Menu> Solution> Loading Options> Curve Options**

Defined data curves can be listed using **EDCURVE**,LIST. They can be plotted using **EDCURVE**,PLOT and can be deleted using **EDCURVE**,DELETE.

The following example shows how data curves can be used to define a piecewise linear plasticity curve (**TB,PLAW,8**) for steel:

```plaintext
! "3" was arbitrarily chosen as the material reference (MAT) number.
mp, ex,3,30.0e6 ! elastic (Young's) modulus (psi)
mp,dens,3,7.33e-4 ! mass density (lbf-sec^2/in^4)
mp,nuxy,3,0.30 ! Poisson's ratio (unitless)
!Note: First convert engineering stress versus engineering strain data
! into true stress versus true (hencky) strain data. Then subtract
! off the elastic true strain from the total true strain to find
! the plastic true strain, which is used with the total true stress
! in LS-DYNA *MAT_PIECEWISE_LINEAR_PLASTICITY material model #24.

!------------------------------------------------------------------------
! Stress-Strain Data used with Piecewise Linear Plasticity (Power Law 8):
!------------------------------------------------------------------------
! Stress/Eng.  Total  Elastic
! Strain/Eng.  Total  Total  Strain  Strain  Strain
! Point  (psi)  (in/in)  (psi)  (in/in)  (in/in)  (in/in)
!------------------------------------------------------------------------
! 1  0  0.0000  0  0.0000  0.0000  0.0000
```

Release 12.1 - © 2009 SASIP, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.
4.1.4.2. Using Data Curves for Loading

In addition to their use with certain material models, data curves (also known as load curves) can be used in the definition of time dependent loads. The procedure to define a load curve is the same as described above for material data curves, except that the first array parameter must contain the time values and the second array parameter must contain the corresponding load values. After you define a load curve with the EDCURVE command, the corresponding load curve reference number (LCID) can be input on the EDLOAD command.

The following 3-step example shows how load curves are used with the EDLOAD command.

```plaintext
! Step 1: Define Array Parameters
*dim, time,, 5
  time(1)=0, 0.025, .05, .075, .1
*dim, yforce,, 5
  yforce(1)=0, 100, 200, 300, 400
!
! Step 2: Define Load Curve and Corresponding LCID (#11)
edcurve, add, 1, time, yforce
!
! Step 3: Refer to LCID on the EDLOAD Command
edload, add, fy, , , comp,,,, 11, 1.0
```

As shown above, the LCID is referenced in the ninth field of the EDLOAD command. It is important to note that if you use an LCID with the EDLOAD command, array parameters should not also be specified to define the load. Using an LCID instead of array parameters on the EDLOAD command is particularly useful when a specific load curve is to be applied to multiple components or load labels.

4.1.5. Defining Loads in a Local Coordinate System

ANSYS LS-DYNA allows you to apply a load to a component or part in any coordinate direction for which a local coordinate system has been defined. This is accomplished using the KEY field of the EDLOAD command in conjunction with the local coordinate system (EDLCS command). Once a local coordinate system has been defined with the EDLCS command (EDLCS, ADD, CID, X1, Y1, Z1, X2, Y2, Z2, X3, Y3, Z3), the local coordinate system ID (CID) is used in the KEY field of the EDLOAD command to define the direction in which the load acts. For force and moment loads (Lab = FX, MX, etc. on EDLOAD), the load will be applied in the direction of the local coordinate system defined by EDLCS. For prescribed motion degrees of freedom (Lab = UX, ROTX, VX, AX, etc. on EDLOAD), the motion will act in the direction of a vector from point (X1, Y1, Z1) to point (X2, Y2, Z2) as input on EDLCS. If CID is not specified, the load will act in the global Cartesian co-
ordinate direction. Some load types do not support the use of local coordinate systems; see Birth Time, Death Time, and CID Support in the Notes section of the EDLOAD command for more information.

### 4.1.6. Specifying Birth and Death Times

For each load definition, you can specify a birth time and a death time by using the BTIME and DTIME fields of the EDLOAD command (EDLOAD,ADD,Lab,KEY,Cname,Par1,Par2,PHASE,LCID,SCALE,BTIME,DTIME). These options allow you to activate a load at any time during an analysis and then deactivate it at a later time. Such an option can be very useful in applications such as a multistage forming process where more than one load is applied in succession. Some load types do not support birth and death time; see Birth Time, Death Time, and CID Support in the Notes section of the EDLOAD command for more information.

### 4.2. Constraints and Initial Conditions

Before initiating the solution, you will need to apply constraints to your model. Additionally, you may wish to set initial velocities for moving objects.

#### 4.2.1. Constraints

Unlike the ANSYS implicit program, ANSYS LS-DYNA does distinguish between zero and nonzero constraints. Nonzero constraints are handled as loads (together with a load curve; see earlier discussion in this chapter). Only zero constraints can be applied with the D command, i.e., the value specified must always be 0 (zero). No other values are valid, as the D constraints are only used to fix certain portions of the model. You may use symmetry/antisymmetry boundary conditions to apply these “zero” constraints.

You can use the EDNROT command to apply zero constraints in terms of a rotated nodal coordinate system. You must first define the local coordinate system with the EDLCS command.

When modeling small symmetric sections of a geometry, you may need to define sliding or cyclic symmetry. You can use the EDBOUND command to define symmetry boundary planes for sliding or cyclic symmetry. You use nodal components to identify the boundaries and a direction vector to define a normal (sliding symmetry) or axis of rotation (cyclic symmetry).

Various other types of constraints can be modeled in ANSYS LS-DYNA using the EDCNSTR command. The available constraint types are extra node set (ENS), nodal rigid body (NRB), shell edge to solid (STS), and rivet (RIVET). In the GUI, you access these constraints by picking:

**Main Menu> Solution> Constraints> Apply> Additional Nodal**

The extra node set constraint type (EDCNSTR,ADD,ENS) allows the addition of nodes (via a nodal component) to an existing rigid body that was defined with the EDMP command. The nodal component that is added must not be attached to any other rigid body. The extra nodes that are added to a rigid body may be located anywhere in the model, and may have coordinates outside those of the original rigid body. The ENS option has many potential applications, including placing nodes where joints will be attached between rigid bodies, defining nodes where point loads will be applied, and defining a lumped mass at a specific location.

Unlike typical rigid bodies that are defined with the EDMP command, nodal rigid bodies defined with the EDCNSTR,ADD,NRB command are not associated with a part number, but are based on a node component. The NRB option can be advantageous when modeling rigid (welded) joints in a model. For a rigid joint, portions of different flexible components (having different MAT IDs) act together as a rigid body. It is difficult to define this type of rigid body with a unique MAT ID (and corresponding part number). However, the rigid joint can be easily defined using a nodal rigid body. Because nodal rigid bodies are not associated with a
part number, other options that use rigid bodies (such as loads applied with the **EDLOAD** command) cannot be used with a nodal rigid body.

The shell to solid edge option (**EDCNSTR**,ADD,STS) ties regions of solid elements to regions of shell elements. As shown in **Figure 4.1: Constrained Shell to Solid** (p. 41), a single shell node may be tied to up to nine solid nodes that define a ‘fiber’ vector. The solid element nodes that define the fiber remain linear (straight) throughout the analysis but can move relative to each other in the fiber direction. The shell node must be coincident with one of the solid element nodes along the fiber.

**Figure 4.1: Constrained Shell to Solid**

Similar to spotwelds defined with the **EDWELD** command, the RIVET option (**EDCNSTR**,ADD,RIVET) defines a massless rigid constraint between two nodes that are noncoincident. Unlike a spotweld, however, failure cannot be specified for a rivet. When a rivet is defined, the distance between the nodes is kept constant throughout any motion that occurs during a simulation. Nodes connected by a rivet cannot be part of any other constraints specified in the model.

### 4.2.2. Welds

In an explicit dynamic analysis, it is common to model components that are physically welded together. This is especially true in automotive applications where parts are often assembled using welds. In such cases, welded constraints can be modeled in ANSYS LS-DYNA using the **EDWELD** command. Two different types of welds can be modeled: massless spotwelds and generalized welds. Nodes connected by the **EDWELD** command cannot be constrained in any other way.

For a massless spotweld, you must specify two noncoincident nodes. You can define failure within the spotweld by inputting failure parameters on the **EDWELD** command. This failure is based on the following relationship:

\[
\left( \frac{f_n}{S_n} \right)^{\text{exp n}} + \left( \frac{f_s}{S_s} \right)^{\text{exp s}} \geq 1
\]
In the GUI, use the following menu path to define a massless spotweld: **Main Menu > Preprocessor > LS-DYNA Options > Spotweld > Massless Spotwld**

A generalized weld is typically used to model a long-welded section of two parts. For a generalized weld, you must specify a valid nodal component. Coincident nodes are permitted. However, if you use coincident nodes, you must specify a local coordinate system (EDLCS) which will be used for output data. Failure within the generalized weld is also defined with the failure parameters on the **EDWELD** command using the same relationship as shown in the equation above.

In the GUI, use the following menu path to define a generalized weld: **Main Menu > Preprocessor > LS-DYNA Options > Spotweld > Genrlzsd Spotwld**

### 4.2.3. Initial Velocity

In transient dynamic simulations, it is often necessary to specify initial conditions. In the ANSYS LS-DYNA program, initial velocities are defined using the **EDVEL** and **EDPVEL** commands. You can use these commands to apply both translational and rotational velocities to various entities. Use **EDVEL** to apply velocities to a nodal component or to a single node; use **EDPVEL** to apply velocities to a part or part assembly.

The **EDVEL** and **EDPVEL** commands provide two methods for specifying rotational velocities, **Option = VGEN** and **Option = VELO**. The VGEN method applies a rigid body rotation to an entity (node component, part, etc.) around a specified axis. The VELO method applies the rotational velocities directly to each node's rotation degree of freedom. Since only shell and beam elements have rotation degrees of freedom, the rotations input with the VELO method are only applicable to **SHELL163** and **BEAM161** elements. For both the VGEN and VELO methods, the translational velocities are always defined relative to the global Cartesian coordinate system.

**Note**

To model a rotating body, with or without translations, you should use **Option = VGEN** since this method applies a rigid body rotation.

Due to the LS-DYNA architecture, the two methods for defining initial velocities, **Option = VELO** and **Option = VGEN**, cannot be used in the same analysis.

To define an initial velocity, use the following procedure:

1. Define the entity to which you want to apply the initial velocity. This may be a single node, a nodal component [CM], a part [EDPART], or a part assembly [EDASMP].
2. Decide whether the VELO or VGEN option is appropriate for your application.
3. Specify the initial velocity [EDVEL or EDPVEL] by using one of the following menu paths in the GUI:

   **Main Menu > Solution > Initial Velocity > On Nodes > w/Nodal Rotate (VELO option)**
   **Main Menu > Solution > Initial Velocity > On Nodes > w/Axial Rotate (VGEN option)**
   **Main Menu > Solution > Initial Velocity > On Parts > w/Nodal Rotate (VELO option)**
   **Main Menu > Solution > Initial Velocity > On Parts > w/Axial Rotate (VGEN option)**

For either the VELO or the VGEN option, input the translational velocities relative to the global Cartesian coordinate system (VX, VY, and VZ fields on the **EDVEL** or **EDPVEL** command). For the VELO option, input the nodal rotations in terms of the global Cartesian coordinate system (OMEGAX, OMEGAY, and OMEGAZ fields). For the VGEN option, input the magnitude of the angular velocity (OMEGAX), the coordinate axis for...
rotation \((X, Y, C, \text{ and } ZC \text{ fields})\) and directional angles \((\text{ANGX}, \text{ ANGY}, \text{ and } \text{ANGZ} \text{ fields})\) relative to the global X, Y, and Z axes.

If you do not specify initial velocities using \texttt{EDVEL} or \texttt{EDPVEL}, all initial velocities in the model will be zero. Likewise, if you only specify the \textit{Cname} field on the \texttt{EDVEL} command (e.g., \texttt{EDVEL, VGEN, Cname}) or the \textit{PID} field on the \texttt{EDPVEL} command, (e.g., \texttt{EDPVEL, VGEN, PID}), zero initial velocities will be applied since the remaining fields on these two commands default to zero.

If you wish to change an initial velocity previously specified using \texttt{EDVEL}, just respecify a new velocity with the same component name or node number. This new value will overwrite the old data for the component or node. To list or delete initial velocities previously applied to node components or nodes, use the \texttt{EDVEL, LIST} and \texttt{EDVEL, DELE} commands (in the GUI, use \textit{Main Menu> Solution> Initial Velocity> On Nodes> List} and \textit{Main Menu> Solution> Initial Velocity> On Nodes> Delete}).

Similarly, if you wish to change an initial velocity previously specified using \texttt{EDPVEL}, just respecify a new velocity with the same part or part assembly ID number. This new value will overwrite the old data for the part or part assembly. To list or delete initial velocities previously applied to parts or part assemblies, use the \texttt{EDPVEL, LIST} and \texttt{EDPVEL, DELE} commands (in the GUI, use \textit{Main Menu> Solution> Initial Velocity> On Parts> List} and \textit{Main Menu> Solution> Initial Velocity> On Parts> Delete}).

For our baseball example, the initial velocity of the baseball would be specified as follows:

\begin{verbatim}
edvel,velo,ball,,,-1600.0 ! Component ball initially moving
\quad ! in -Z direction at 1600 in/sec
\end{verbatim}

Or, assuming the component \textit{ball} was part number 1 in the model, you could specify the velocity of the baseball as:

\begin{verbatim}
edpvel,velo,1,,,-1600.0 ! Part number 1 initially moving
\quad ! in -Z direction at 1600 in/sec
\end{verbatim}

4.3. Coupling and Constraint Equations

The \texttt{CP-family} of commands (\texttt{CP, CPDELE, CPINTF, CPLGEN, CPLIST, CPNGEN, CPSGEN}) can be used to define, modify, delete, list, and generate coupling between different degrees of freedom (DOFs) of a structure. Similarly, the \texttt{CE-family} of commands (\texttt{CE, CEDELE, CEINTF, CELIST, CERIG}) can be used to define, modify, delete, and list constraint equations between different DOFs of a structure. When used in the ANSYS LS-DYNA explicit dynamics program, the \texttt{CP} and \texttt{CE} commands can be used only with UX, UY, and UZ DOFs (rotational DOFs are not allowed).

Since the rotational DOFs (ROTX, ROTY, ROTZ) are not allowed, the \texttt{CP} family of commands should not be used in an explicit analysis to model rigid body behavior that involves rotations. If \texttt{CP} is used in this manner, it could lead to nonphysical responses.

Also, be aware that a coupled set containing nodes that are not coincident or nodes that are not along the line of the coupled degree of freedom does not produce a moment constraint. This means that if the structure rotates, the set of coupled nodes will rotate. Only the applied forces and the reaction forces will satisfy the moment equilibrium in the model. For each node in the coupled set, the result of satisfying the moment will depend on the distance from the node to the center of the coupled set, and the direction of displacement will depend on the resulting moment. This may lead to a nonphysical response in some cases.

See \textit{Coupling and Constraint Equations} in the \textit{Modeling and Meshing Guide} for details on defining coupling and constraint equations.
4.4. Nonreflecting Boundaries

When modeling geomechanical systems, an infinite domain is often required to represent the ground or other large solid body. For this type of analysis, you can use nonreflecting boundaries at the exterior of the model to limit the overall size of the model. You apply these boundaries to the surface of SOLID164 and SOLID168 elements that are being used to model an infinite domain. Nonreflecting boundaries will prevent artificial stress wave reflections generated at the boundary from reentering the model and contaminating the results.

When nonreflecting boundaries are included in the model, LS-DYNA computes an impedance matching function for all boundary segments based on an assumption of linear material behavior. Therefore, the finite element mesh should be constructed so that all significant nonlinear behavior is contained within the discrete model.

To define a nonreflecting boundary, select the nodes (NSEL command) that make up the desired boundary along the external surface of a SOLID164 or SOLID168 body, then define a component from these nodes (CM command). Use the EDNB command to apply the nonreflecting boundary to that nodal component, and to turn on the dilatational and shear activation flags. For example, the following command defines a nonreflecting boundary on the component “ground”:

EDNB,ADD,GROUND,1,1

To list and delete defined nonreflecting boundaries, use the commands EDNB LIST and EDNB DELE, respectively.

4.5. Temperature Loading

You may need to define temperature loading in an explicit dynamic analysis in order to accommodate temperature dependent materials, or to include the effects of thermally induced stresses. Temperature loading is applicable to the PLANE162, SHELL163, SOLID164, and SOLID168 element types. The ANSYS LS-DYNA program offers several types of temperature loading:

- Time-varying temperature applied to a nodal component (EDLOAD)
- Constant temperature applied to all nodes in the model (TUNIF / BFUNIF)
- Temperature results from an ANSYS thermal analysis applied as non-uniform temperature loads (that do not vary with time) in a subsequent explicit dynamic analysis (LDREAD; also requires a sequential solution)

The first method utilizes the EDLOAD,TEMP command and the general loading procedure to apply a time-varying temperature to a specific nodal component. You must define two array parameters to represent the load; the first parameter contains the time values and the second parameter contains the temperature values. You may choose to define a load curve with these parameters or to input them directly on the EDLOAD command. You can also scale the temperature values using the SCALE parameter on EDLOAD. See General Loading Options (p. 33) for a complete description of how to apply this type of load.

The second method allows you to apply a uniform constant temperature to all nodes in the model. This method may be used to model a structure subjected to steady-state thermal loading. You can use the TUNIF command or the BFUNIF,TEMP command to apply this type of temperature load.

The third method allows you to apply the temperatures calculated in an ANSYS thermal analysis as loads in an explicit dynamic analysis. This method is useful for modeling temperature-dependent phenomena such as forging. To use this method, you must perform an implicit-to-explicit sequential solution. In the explicit phase, you use the LDREAD command to read the temperature data from the thermal (implicit) analysis.
results file (Jobname.RTH) and apply them as loads to the nodes in the model. You can only transfer the temperatures from one specified time point in the thermal analysis. Refer to Chapter 15, Implicit-to-Explicit Sequential Solution (p. 135) of this guide for a detailed description of this loading procedure.

For all three methods of temperature loading, a reference temperature can be input via the TREF command. The thermal loading is defined as the difference between the applied temperature and the reference temperature. If the reference temperature is not specifically defined, it defaults to zero.

In order for any of these thermal loads to take effect, you must use the temperature dependent bilinear isotropic material model or the elastic viscoplastic thermal material model. The temperature dependent bilinear isotropic model can be used to represent a thermoelastic material (if needed) by omitting the yield strength and tangent modulus. Please refer to Chapter 7, Material Models (p. 67) of this guide for complete descriptions of these material models.

Note
It is important to note that the EDLOAD method of temperature loading cannot be used together with the LDREAD or TUNIF / BFUNIF methods of temperature loading. In addition, EDLOAD cannot be used to list or delete temperatures applied with LDREAD, TUNIF, or BFUNIF.

You can use LDREAD and TUNIF (or BFUNIF) together in the same explicit dynamic analysis. When LDREAD is issued, it applies temperature loads to the selected nodes, overwriting any temperature loads defined by TUNIF or BFUNIF. Any nodes that are unselected at the time LDREAD is issued will take on the temperature load defined by the TUNIF or BFUNIF command. To delete the temperature loads defined by LDREAD (at selected nodes or all nodes), use the BFDELE command. To list the temperature loads defined by LDREAD (at selected nodes or all nodes), use the BFLIST command.

4.6. Dynamic Relaxation

In order to conduct implicit-to-explicit sequential solutions (see Chapter 15, Implicit-to-Explicit Sequential Solution (p. 135)), dynamic relaxation capability has been added to the ANSYS LS-DYNA program (EDDRELAX command). True dynamic relaxation (EDDRELAX,DYNA) allows an explicit solver to conduct a static analysis by increasing the damping until the kinetic energy drops to zero. When an implicit solver is used to provide the preload (EDDRELAX,ANSYS), a slightly different approach is taken, in that the stress initialization is based on a prescribed geometry (i.e., the nodal displacement results from the implicit solution). In this latter case, the explicit solver only uses 101 time steps to apply the preload. In the former case, the solver will check the kinetic energy every 250 cycles (by default) until the kinetic energy from the applied preload is dissipated. ANSYS LS-DYNA supports both methods, which occur in pseudo time before the transient portion of the analysis begins at time zero. The EDLOAD command specifies the analysis type through the PHASE label.

EDLOAD, ADD, Lab, , Cname, Par1, Par2, PHASE

PHASE: 0 Load curve for transient analysis only (default) or for implicit-to-explicit sequential solutions
        1 Load curve for dynamic relaxation only
        2 Load curve for both transient and dynamic relaxation analyses

There are essentially five different types of analyses that can be conducted with ANSYS LS-DYNA concerning dynamic relaxation. They are discussed below:

1. Transient Dynamic Analysis Only (default - EDDRELAX,OFF): In this case, the PHASE parameter of the EDLOAD command is set to zero and dynamic relaxation is not used. This is the default setting.
2. Stress Initialization Only with No Transient Analysis (EDDRELAX, DYNA): This case basically uses the ANSYS LS-DYNA explicit solver to approximate the solution of a static analysis that is really best handled by an implicit solver such as ANSYS. The PHASE parameter on the EDLOAD command is set to one. The termination time (TIME command) must be set to zero to prevent unloading of the static load. True dynamic relaxation is used.

3. Stress Initialization with a No-Load Transient Analysis (EDDRELAX, DYNA): This case is the same as the previous case, except that the termination time is set to the desired value. At time zero, the structure is instantly unloaded and free to vibrate.

4. Stress Initialization with a Loaded Transient Analysis (EDDRELAX, DYNA): This is similar to the previous case, except that the load curve specified by the EDLOAD command (via PHASE = 2) will be used in both pseudo time for the dynamic relaxation to obtain the preload and in real time for the transient analysis. If a ramped load curve is used instead of a constant load curve, then the structure will unload at time zero and reload again. Therefore, if a ramped load curve is needed for the explicit solver to accurately converge on the static solution, it would be better to use two EDLOAD commands, the first (with PHASE = 1) to apply the preload using a ramped load curve and the second (with PHASE = 0) to continue the load with a load curve that is not ramped.

Note

If you issue EDLOAD multiple times for the same component (or Part ID) and the same load label, the values associated with the last EDLOAD command are used. Consequently, you cannot use multiple values of the PHASE parameter for a given component (or Part ID) and load label. To work around this limitation, you must use duplicate nodal components for the stress initialization and transient curves. (This workaround is applicable only if you use components to define the load.)

5. Implicit-to-Explicit Sequential Solutions (EDDRELAX, ANSYS): This case uses the displacement results from the ANSYS implicit solver to apply the preload for the ANSYS LS-DYNA explicit solver via stress initialization to a prescribed geometry. The PHASE parameter on the EDLOAD command must be set to zero. This scenario is explained in detail in Chapter 15, Implicit-to-Explicit Sequential Solution (p. 135) of this guide. Fields 2-6 of the EDDRELAX command are ignored for this case.
Chapter 5: Solution Features

The solution procedure for an explicit dynamic analysis is similar to an implicit analysis. However, the ANSYS LS-DYNA program offers several solution features that are unique to an explicit analysis.

The following ANSYS LS-DYNA solution topics are available:

5.1. Solution Process
5.2. LS-DYNA Termination Controls
5.3. LS-DYNA Parallel Processing Capabilities
5.4. Double Precision LS-DYNA
5.5. Solution Control and Monitoring
5.6. Plotting Small Elements
5.7. Editing the LS-DYNA Input File

5.1. Solution Process

After the model has been built (i.e., after completing element definitions, real constant and material property specifications, modeling, meshing, boundary/initial conditions, loading, and termination controls), you can start the solution process by issuing the SOLVE command (in the GUI, pick Main Menu> Solution> Solve).

At this point, several steps are performed by the ANSYS LS-DYNA program:

1. The header records, including geometric quantities (such as nodes and elements), are written on the two results files Jobname.RST and Jobname.HIS. (At this point, the ANSYS LS-DYNA database should contain all relevant information. Before issuing the SOLVE command, you should issue a SAVE command to write all model information to the file Jobname.DB.)

2. The input file for the LS-DYNA program, Jobname.K, is written using all information that has been entered so far.

3. Control is transferred from the ANSYS LS-DYNA program to the LS-DYNA program. The solutions produced by the LS-DYNA solver are written to the results files Jobname.RST and Jobname.HIS. If the EDOPT,ADD,BOTH option was specified before issuing the SOLVE command, results files for the LS-PREPOST postprocessor (files d3plot and d3thdt) will also be written.

When the solution is complete without errors or warnings, the ANSYS LS-DYNA GUI notifies the user that the solution is done, and control is transferred back to the ANSYS LS-DYNA program. The results can be viewed using the POST1 and POST26 processors of the ANSYS LS-DYNA program. If errors or warnings are produced, messages pop-up and are also displayed in the ANSYS Output Window. The messages state the number of errors and warnings that were produced, and make reference to the LS-DYNA message file, where details of the errors and warnings are written. These same details are also written to the LS-DYNA d3hsp file.

5.2. LS-DYNA Termination Controls

The point at which the LS-DYNA solution terminates will depend on the termination controls you specify when setting up the model. Several types of termination controls are available:
• Termination time - Use the TIME command to specify an end-time for the analysis. The calculation will stop when the accumulation of time steps reaches that end-time.

• CPU time limit - You can use the EDCPU command to specify a CPU time limit (in seconds). The calculation will stop when that time limit is reached.

• Termination criteria - You can use the EDTERM command to specify that the solution stop when a specific node or rigid body reaches a certain position, or when a specific node comes into contact with another surface. You may set up multiple criteria using this command. (See the EDTERM command for details.)

You should always specify an analysis end-time using the TIME command. The other termination controls are optional. The solution will terminate when any one of the specified termination criteria is met.

When you are performing a small restart (EDSTART,2) or a full restart (EDSTART,3) analysis, if the previous analysis terminated due to a satisfied termination criteria that was set by EDTERM, you must change or delete that criteria so that it will not cause the restart to terminate immediately.

5.3. LS-DYNA Parallel Processing Capabilities

For large models, you can use the shared memory parallel processing (SMP) or the massively parallel processing (MPP) capabilities of ANSYS LS-DYNA to shorten the elapsed time necessary to run an analysis. To use either of these features, you must purchase the appropriate number of ANSYS LS-DYNA Parallel licenses. The Parallel license incorporates both SMP and MPP capabilities. Please contact your ANSYS sales representative for more information on purchasing the appropriate licenses.

5.3.1. Shared Memory Parallel Processing

The shared memory parallel processing capabilities allow you to distribute model-solving power over multiple processors on the same machine. To use this feature you must have a machine with at least two processors, and you must have an ANSYS LS-DYNA Parallel license for each processor beyond the first one.

To use the SMP capabilities you need to issue the EDRUN command to specify an SMP run as well as the number of processors to use (EDRUN,SMP,,Ncpu).

When you are using shared memory parallel processing, the calculations may be executed in different order, depending on CPU availability and the workload on each CPU. Because of this, you may see slight differences in the results when you run the same job multiple times. To avoid these differences, you can specify that consistency be maintained (EDRUN,SMP,1). Maintaining consistency can result in an increase of up to 15% in CPU time.

If you are using all CPUs on a platform (ANSYS recommends always running one fewer CPU than is available), you should close all other applications on those CPUs. Any calculations or machine resources that compete with the ANSYS LS-DYNA application running in SMP mode will reduce the SMP performance significantly.

The SMP capability is expected to scale linearly when used with up to 5 processors. When used with more than 5 processors, any additional scalability is minimal; the larger the problem, the higher the potential for scalability.) If you are using more than 5 processors, we recommend using the MPP capability, which offers excellent scalability regardless of the number of processors.
5.3.2. Massively Parallel Processing

The massively parallel processing (MPP) capabilities of ANSYS LS-DYNA allow you to run the LS-DYNA solver over a cluster of machines or use multiple processors on a single machine. To use the LS-DYNA MPP feature, you must have an ANSYS LS-DYNA Parallel license for each processor beyond the first one.

Before running an analysis using LS-DYNA MPP, you must configure your system properly. You must have supported MPI software correctly installed, and the machines running LS-DYNA MPP must be properly configured. The configuration procedures for LS-DYNA MPP are the same as for Distributed ANSYS. For more information, see Prerequisites for Running Distributed ANSYS and Setting Up the Environment for Distributed ANSYS in the Distributed ANSYS Guide.

Not all platforms and MPI software listed in Prerequisites for Running Distributed ANSYS are supported by LS-DYNA MPP. For details, refer to the discussion of LS-DYNA MPP MPI support found in that section.

To run LS-DYNA MPP you can use the High Performance Computing Tab in the ANSYS Launcher or run the LS-DYNA executable via the command line using the correct arguments (see Starting Distributed ANSYS in the Distributed ANSYS Guide for more information). Running LS-DYNA MPP from the command line is similar to running Distributed ANSYS via the command line. The arguments that are common to both programs are:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-dis</td>
<td>Enables distributed processing (MPP) version of LS-DYNA</td>
</tr>
<tr>
<td>-np</td>
<td>Specifies the number of processors to use when running LS-DYNA MPP</td>
</tr>
<tr>
<td>-machines</td>
<td>Specifies the machines on which to run an LSDYNA MPP analysis</td>
</tr>
<tr>
<td>-mpifile</td>
<td>Specifies an existing MPI file</td>
</tr>
</tbody>
</table>

The -dis argument is required in order to run LS-DYNA MPP. Choose one of the three remaining arguments (-np, -machines, or -mpifile) to indicate how the analysis will be distributed over multiple processors or machines. Note that the -np and -machines arguments are mutually exclusive; -machines will override -np. See Starting Distributed ANSYS via Command Line in the Distributed ANSYS Guide for information on how to use these command line options.

Other command line options (such as I/O and other LS-DYNA specific arguments) are unique to LS-DYNA. Because you are using the command line, you must follow the procedure outlined in Editing the LS-DYNA Input File (p. 52). This involves using the EDWRITE command to create a Jobname.K input file, then using the lsdyna121 script with the appropriate arguments.

Other command line options (such as I/O and other LS-DYNA specific arguments) are unique to LS-DYNA. Because you are using the command line, you must follow the procedure outlined in Editing the LS-DYNA Input File (p. 52). This involves using the EDWRITE command to create a Jobname.K input file, then using the lsdyna121 script with the appropriate arguments.

The following is an example of a command line used to run LS-DYNA MPP. This example assumes you have ANSYS Mechanical with LS-DYNA at ANSYS Release 12.1, your input file is "file.k", and you are using two machines (one with four processors and one with two processors).

```
lsdyna121 -dis -machines machine1:4:machine2:2 i=file.k pr=ansysds
```
Chapter 5: Solution Features

The output files produced by LS-DYNA MPP are standard LS-DYNA output files and are not compatible with ANSYS. Therefore, you must use LS-DYNA postprocessors to view the results.

5.4. Double Precision LS-DYNA

For more accurate results, you can use the double precision capabilities of ANSYS LS-DYNA. This feature is especially useful in sequential explicit-to-implicit spring back types of analyses. To use this feature, you must have the double precision version of LS-DYNA available on your machine.

Note

The double precision version of LS-DYNA may be up to 20% slower than the single precision version. Results may also vary based on problem specifications.

Use the EDDBL (EDDBL,1) command to specify the double precision version.

You can check the numeric version status by issuing the EDDBL,STATUS command or the GUI dialog box for the command (accessed through Main Menu> Solution> Analysis Options> Double Precision).

5.5. Solution Control and Monitoring

The ANSYS LS-DYNA program allows certain sense switch controls that enable you to interrupt the solution process and to check the solution status. This section describes how to use these sense switch controls.

While the LS-DYNA execution is in progress, type CTRL-C in the console window of the ANSYS LS-DYNA program. This interrupts the LS-DYNA program execution and lets you enter program control commands (i.e., switches) in the console window of ANSYS LS-DYNA. (Unlike in ANSYS, you are not thrown out of the ANSYS LS-DYNA GUI by the CTRL-C interrupt, and the LS-DYNA solver will continue to run in the background.) A typical segment of output from the console window is shown below:

```
dt of cycle 1 is controlled by solid element 1

   time: 0.00000E+00
   time step: 3.30496E-05
   kinetic energy: 2.82787E+03
   internal energy: 1.00000E-20
   spring and damper energy: 1.00000E-20
   hourglass energy: 0.00000E+00
   system damping energy: 0.00000E+00
   sliding interface energy: 0.00000E+00
   external work: 0.00000E+00
   eroded kinetic energy: 0.00000E+00
   eroded internal energy: 0.00000E+00
   total energy: 2.82787E+03
   total energy / initial energy: 1.00000E+00
   energy ratio w/o eroded energy: 1.00000E+00
   global x velocity: 0.00000E+00
   global y velocity: -6.23477E+01
   global z velocity: 0.00000E+00
   cpu time per zone cycle: 6666660 nanoseconds
   average cpu time per zone cycle: 6666660 nanoseconds
   average clock time per zone cycle: 15167664 nanoseconds
   estimated total cpu time: 6 sec (0 hrs 0 mins)
   estimated cpu time to complete: 6 sec (0 hrs 0 mins)
   estimated total clock time: 13 sec (0 hrs 0 mins)
   estimated clock time to complete: 13 sec (0 hrs 0 mins)
   enter sense switch:
```

Release 12.1 - © 2009 SASIP, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.
At this point, you can enter one of the following four switches:

sw1  ANSYS LS-DYNA terminates. A restart file is written.
sw2  ANSYS LS-DYNA responds with time and cycle numbers and continues to run.
      This allows you to see how far the solution has actually progressed.
sw3  ANSYS LS-DYNA writes a restart file and continues to run.
sw4  ANSYS LS-DYNA writes a results data set and continues to run.

The first estimate of CPU time reported in the console window (before the issuance of CTRL-C) is usually inaccurate (see “estimated total cpu time” in the output example shown above). You can use CTRL-C to interrupt the execution of the LS-DYNA solver, then type the sense switch sw2 to get a better estimate of the execution time and cycle numbers. The ANSYS LS-DYNA program writes all important messages (errors, warnings, failed elements, contact problems, etc.) to the ANSYS LS-DYNA console window and to the LS-DYNA ASCII output file d3hsp.

On a UNIX system, the progress of the ANSYS LS-DYNA solution can also be checked by reading the last lines of certain ASCII output files. To do so, use another window to go to the directory where the ANSYS LS-DYNA program was started. Type the following command at the system prompt:

tail -m filename

where filename = d3hsp or glstat or matsum. For m (the number of lines to be viewed), a value of 30 should be sufficient.

The d3hsp file will always be created by ANSYS LS-DYNA. The files glstat and matsum will be created only if the following commands were issued in the PREP7 or SOLUTION processors of ANSYS LS-DYNA:

    EDOUT, GLSTAT
    EDOUT, MATSUM

(See Chapter 12, Postprocessing (p. 115) of this manual for more information on how output control commands are used.)

The files d3hsp, glstat and matsum are ASCII files. As the solution progresses, these files are written at requested time points. By looking at the current "time" value of the solution from the last 30 lines of any one of these files and comparing this "time" value against the specified "end time" value on the TIME command, you can determine the progress of the solution.

5.6. Plotting Small Elements

As outlined in Comparison of Implicit and Explicit Methods, ANSYS LS-DYNA automatically calculates the critical time step size of each element in a model based on its material properties and size. The overall time step for an entire model is then based on the smallest critical time step value of all the elements within the model.

You can plot explicit dynamic elements based on their time step sizes using the EDTO command. This allows you to monitor areas of the model that have small elements. There are three options available (OPTION field of EDTO) for plotting elements based on time step size.

- Option 1 - Plot the elements with the smallest time step sizes. Each element is shaded red or yellow based on its time step size. The number of elements plotted is based on the VALUE field (default = 100) of the EDTO command.
• Option 2 - Plot and list the elements with the smallest time step sizes. This option produces the same plot as Option 1, and also provides a listing of the smallest elements.

• Option 3 - This option produces a plot similar to Option 1, except that all selected elements are plotted. Elements beyond the number specified by \textit{VALUE1} are blue and translucent, with the degree of translucency based on the \textit{VALUE2} field of the \textbf{EDTP} command.

In the plot produced by \textbf{EDTP} the elements are shaded red (smallest), yellow (intermediate), or blue (largest) based on their time step sizes.

\begin{center}
\textbf{Note}
\end{center}

Care should be taken when using the \textbf{EDTP} command for large models because the time step size calculations may take a significant amount of CPU time.

\section*{5.7. Editing the LS-DYNA Input File}

Most of the major LS-DYNA capabilities are supported in the ANSYS LS-DYNA program and can be conveniently accessed through the graphical user interface of this program. It should be noted, however, that there are a number of other capabilities in the LS-DYNA program that cannot be directly accessed through the GUI of the ANSYS LS-DYNA program. Some examples are:

• several material properties, such as fabric, soil, etc.
• certain element types, such as seat belt
• application of constraints on rigid bodies in the local coordinate system

Although these capabilities cannot be directly accessed, users who are familiar with the LS-DYNA program can still use these features indirectly in the ANSYS LS-DYNA program. The procedure for including additional features is outlined below.

\begin{center}
\textbf{Note}
\end{center}

Only the LS-DYNA features that are described in the ANSYS LS-DYNA documentation and available through the ANSYS LS-DYNA interface are supported features. Although you can access other features as described below, features that you implement by editing the LS-DYNA input file are not supported.

After modeling has been completed, you can issue the \textbf{EDWRITE} command instead of issuing the \textbf{SOLVE} command. (In the GUI, pick \textbf{Main Menu> Solution> Write Jobname.k}.) \textbf{EDWRITE} will create the LS-DYNA input file \textit{Jobname.K}. (Note that if the analysis is a small restart, the input file will be named \textit{Jobname.R}, and if it is a full restart, the input file will be named \textit{Jobname\_nn.K}.)

The difference between the \textbf{EDWRITE} command and the \textbf{SOLVE} command is that step number 3 discussed in \textit{Solution Process} (p. 47) will not be executed. That is, \textit{Jobname.K} will be written (along with headers to the ANSYS results files), but the solution process of the LS-DYNA program will not be started. After issuing \textbf{EDWRITE}, use one of the following two methods to include additional functionality in the explicit dynamic analysis.

\textbf{Method A}

1. Exit the ANSYS LS-DYNA program.
2. Edit the LS-DYNA input file \texttt{Jobname.K} to incorporate the items that are not supported directly through the ANSYS LS-DYNA program.

3. Execute the LS-DYNA executable separately in the same directory where the two results files, \texttt{Jobname.RST} and \texttt{Jobname.HIS}, reside. The solution produced by the LS-DYNA executable will be appended to these results files.

4. After completion of the LS-DYNA execution, enter the ANSYS LS-DYNA program and view the results using the postprocessors (POST1 and POST26).

\textbf{Method B}

1. Using another window, go to the directory where the ANSYS LS-DYNA program was started. Do not exit the ANSYS LS-DYNA program in the original window.

2. Edit the LS-DYNA input file \texttt{Jobname.K} to incorporate the items that are not supported directly through the ANSYS LS-DYNA program.

3. Execute the LS-DYNA executable separately in the same directory (where the two results files, \texttt{Jobname.RST} and \texttt{Jobname.HIS}, reside). The solution produced by the LS-DYNA executable will be appended to these results files. (Remember, you have not exited the ANSYS LS-DYNA program in the original window.)

4. After completion of the LS-DYNA execution, go back to the original window and view the results using the ANSYS LS-DYNA postprocessors.

For both Method A and Method B, you execute the LS-DYNA program by running the \texttt{lsdynaxxx} script (where xxx is the ANSYS release number) with the correct arguments:

\begin{verbatim}
lsdynaxxx i = jobname.k pr = product name
\end{verbatim}

For example, if you have ANSYS Mechanical with LS-DYNA at ANSYS Release 12.1 and your input file is “crashtest.k,” you would type:

\begin{verbatim}
lsdyna121 i = crashtest.k pr = ansysds
\end{verbatim}

You may need additional command line arguments, depending on what type of analysis you are running. For example:

- For an implicit-to-explicit sequential solution (\texttt{REXPORT} and \texttt{EDDRELAX} commands; see \textit{Chapter 15, Implicit-to-Explicit Sequential Solution (p. 135)}), the argument "m=drelax" is required.

- For a restart analysis, (\texttt{EDSTART} command; see \textit{Chapter 13, Restarting (p. 123)}), the argument "r=d3dumpmn" is required, where \textit{nn} is the number of the dump file (01, 02, ... 99) to be used as a starting point.

- To run the double precision version of LS-DYNA, the ".-dp" argument is required.

- For any type of analysis, you can specify the memory to be used by including the command line argument "memory=n", where \textit{n} is the desired amount of memory in words.

\textbf{Note}

If the *MAT\_COMPOSITE\_MSC material models (*MAT_161 and *MAT162) are included in the LS-DYNA keyword input file, you will also need to obtain additional licensing from ANSYS, Inc. and set the corresponding environment variables, as discussed in the \textit{ANSYS, Inc. Installation Guide} for your platform.
When using these methods to modify the file Jobname.K, *nodes and elements must not be changed*. Also, note that the ANSYS LS-DYNA database is not updated with the changes you make in file Jobname.K. Thus, if an attempt is made to view the modified items in the PREP7, POST1 or POST26 processors after the solution is complete, the original version, and not the modified version, will be shown. (For these reasons, editing the LS-DYNA keyword input file, Jobname.K, is NOT a supported feature of ANSYS LS-DYNA.) In all cases, the complete set of results can be postprocessed in the LS-PREPOST postprocessor (assuming you requested LS-DYNA results files via the EDWRITE command).

### 5.7.1. Using a Preexisting File.K

In a typical ANSYS LS-DYNA analysis, the results file headers are written when you issue the SOLVE command (or when you issue EDWRITE). However, the LS-DYNA solver also has the ability to create the .RST and .HIS results file headers when only a file.K input file exists. This allows you to take a preexisting file.K, solve it using LS-DYNA, and then postprocess the results with the ANSYS LS-DYNA program.

To use this method, first make sure that the model defined in the .K file does not contain any features that ANSYS LS-DYNA does not support. Then edit the .K file so that the *DATABASE_FORMAT command is set to either 1 for ANSYS only results files, or 2 for ANSYS and LS-DYNA results files. Next, run the lsdynaxx script as described above.

You can then postprocess the results using POST1 and POST26. If the corresponding ANSYS database does not exist, element attributes (material properties, real constants, etc.) will not be available in the results files. However, you should still be able to postprocess the results using the ANSYS postprocessors. If you requested LS-DYNA results files, you will also be able to postprocess the results using LS-PREPOST.
Chapter 6: Contact Surfaces

Contact surfaces in ANSYS LS-DYNA allow you to represent a wide range of types of interaction between components in a model. This chapter provides guidelines for defining physically realistic contact in an explicit dynamic analysis.

It should be noted that contact is represented differently in an explicit dynamic analysis than it is in other types of ANSYS analyses. In other analyses, the contact is represented by actual “contact elements.” For explicit dynamics, there are no contact elements. You simply indicate the contact surfaces, the type of contact between them, and other parameters related to the contact type.

The following explicit dynamics contact topics are available:

6.1. Contact Definitions
6.2. Contact Options
6.3. Contact Search Methods
6.4. Special Considerations for Shells
6.5. Controlling Contact Depth
6.6. Contact Stiffness
6.7. 2-D Contact Option

6.1. Contact Definitions

Due to complicated large deformation dynamics which typically occur during an explicit dynamic analysis, determining contact between components in a model can be extremely difficult. For this reason, special features have been included in the ANSYS LS-DYNA program to make defining contact between surfaces as efficient as possible. All contact surfaces are defined in ANSYS LS-DYNA through the use of the EDCGEN command.

Use the following steps when using the EDCGEN command:

STEP 1: Determine the type of contact surface which best defines your physical model.
STEP 2: Identify contact entities.
STEP 3: Specify friction coefficient parameters.
STEP 4: Specify any additional input which is required for a given contact type.
STEP 5: Specify birth and death times for the contact definition.

STEP 1: Determining Contact Type

In order to adequately describe interaction between complex geometries during large deformation contact and dynamic impact, a large number of contact surface options have been incorporated into the ANSYS LS-DYNA product. These contact types, which include node-to-surface, surface-to-surface, single surface, single edge, eroding, tied, tiebreak, drawbead, and rigid contact options, are discussed in detail later in this chapter (see Contact Options (p. 59)). For most typical analyses, the following contact options are recommended.

• Automatic Single Surface (ASSC): Contact is established when a surface of one body contacts itself or the surface of another body. This type is easy to use because no contact or target surface definitions are required. It is efficient for self-contacting problems or large deformation problems where general areas of contact are not known beforehand.
• Automatic General (AG): This contact type is similar to Automatic Single Surface in that contact is established when a surface of one body contacts itself or the surface of another body. Automatic General contact is also easy to define since you do not need to specify contact and target surfaces. The main advantage of AG contact is that it is based on newer contact algorithms; consequently, its performance may be better than ASSC contact in some applications.

• Node-to-Surface (NTS): Contact is established when a contacting node penetrates a target surface. This type is commonly used for general contact between two surfaces and is most efficient when a smaller surface comes into contact with a larger surface, such as a thin rod impacting a flat plate.

• Surface-to-Surface: (STS) Contact is established when a surface of one body penetrates the surface of another body. This type is commonly used for arbitrary bodies that have large contact areas and is very efficient for bodies that experience large amounts of relative sliding with friction, such as a block sliding on a plane.

**STEP 2: Defining Contact Entities**

With the exception of the single surface (ASSC, SS, and ESS), automatic general (AG), and single edge contact (SE), all of the contact options require you to identify contact and target surfaces for which contact can occur. You can do this by using nodal components, PART ID definitions, and/or part assembly ID definitions. When using contact or target components, use select logic and the **CM** command to group the nodes (only nodal components are valid). Then, as shown in the sample input listing below, use the **EDCGEN** command to specify contact between the desired components such as the ball and bat surfaces of the example problem discussed in *Chapter 4, Loading* (p. 33):

```plaintext
NSEL,S,NODE,.... ! Select a set of nodes on ball surface
CM,BALLSURF,NODE ! Place selected nodes into component
    ! BALLSURF
NSEL,S,NODE,.... ! Select a set of nodes on bat surface
CM,BATSURF,NODE ! Place selected nodes into component
    ! BATSURF
EDCGEN,NTS,BALLSURF,BATSURF,.25,.23 ! Generate node-to-surface contact
    ! between components
    ! BALLSURF and BATSURF
```

Alternatively, contact surfaces can be specified using part numbers or part assembly numbers that are currently defined within the finite element model. Part assembly numbers are defined with the **EDASMP** command.

The input line below demonstrates how the **EDCGEN** command is used to define contact between two different parts / part assemblies in a model:

```plaintext
EDCGEN,STS,1,2,.25,.23 ! Generate surface-to-surface contact
    ! between PARTS 1 and 2
```

Additionally, contact can be defined between contact and target surfaces using a combination of PART / part assembly and component definitions, as illustrated below:

```plaintext
EDCGEN,NTS, N1,2,.3,.28 ! Generate node-to-surface contact
    ! between component N1 and PART 2
EDCGEN,ESTS,1,4,.15,.15 ! Generate eroding surface-to-surface contact
    ! between PART 1 and part assembly 4
EDCGEN,STS,1,1,.1,.1 ! Generate surface-to-surface contact
    ! between PART 1 and itself
```

You can also use the **EDCGEN** command for defining contact between part assemblies as shown in the following sample input:

```plaintext
EDCGEN,STS,5,6,.3,.28 ! Generate surface-to-surface contact
    ! between part assemblies 5 and 6
```
For the specialized case of single surface contact (ASSC, AG, ESS, and SS), no contact or target surface definitions are required. As explained later in this section, single surface contact is the most general type of contact definition as all external surfaces within a model can be in contact at any point during an analysis. The program will ignore any contact and target surfaces defined for single surface contact and will issue a warning message upon execution of the EDCGEN command. A typical command for single surface contact is presented below:

```
EDCGEN,ASSC,,,34,.34 ! Generate automatic single surface contact for the entire model
```

Note--When defining contact entities in an explicit analysis, no initial penetrations are allowed. Therefore, use great care when defining contact components.

**STEP 3: Specifying Friction Coefficient Parameters**

The frictional coefficient used for contact is determined from the static friction coefficient (FS), the dynamic friction coefficient (FD), and the exponential decay coefficient (DC). (FS, FD, and DC can be input on the EDCGEN command.) The frictional coefficient is assumed to be dependent on the relative velocity, \( V_{rel} \), of the surfaces in contact:

\[
\mu_c = FD + (FS - FD)e^{-DC \cdot V_{rel}}
\]

The coefficient for viscous friction, VC (also input on EDCGEN), can be used to limit the friction force to a maximum. A limiting force is computed:

\[
F_{lim} = VC \cdot A_{cont}
\]

where \( A_{cont} \) is the area of the segment contacted by the node in contact. The suggested value for VC is to use the yield stress in shear:

\[
VC = \frac{\sigma_0}{\sqrt{3}}
\]

where \( \sigma_0 \) is the yield stress of the contacted material.

In order to avoid undesirable oscillation in contact, e.g., for sheet forming simulation, a contact damping perpendicular to the contacting surfaces is applied. The contact damping coefficient is calculated as follows:

\[
\xi = \frac{VDC}{100} \cdot \xi_{crit}
\]

VDC is the viscous damping coefficient (input on EDCGEN command as percent of critical damping; e.g., VDC = 20 indicates 20%). \( \xi_{crit} \) is determined in the following fashion by ANSYS LS-DYNA:

\[
\xi_{crit} = 2m\omega
\]

where
\[ m = \min(m_{\text{slave}}, m_{\text{master}}) \]

\[ \omega = \sqrt{\frac{k \cdot m_{\text{slave}} + m_{\text{master}}}{m_{\text{slave}} \cdot m_{\text{master}}}} \quad ; \text{k is interface stiffness} \]

**STEP 4: Specifying Additional Input**

For eroding, rigid, tiebreak, and drawbead contact, additional information (fields \( V1-V4 \) on the **EDCGEN** command) may also be required. This information varies with contact type, as outlined below.

Eroding surface contacts (ENTS, ESS, and ESTS) are used when surfaces of solid elements fail and contact needs to be redefined with the remaining internal elements. For eroding contact, \( V1-V3 \) are defined as follows: the boundary condition symmetry option key (\( V1 \)) determines whether a symmetry condition will be retained along a surface where elements fail; the interior erosion option key (\( V2 \)) specifies whether erosion can subsequently occur along internal surfaces when the exterior surface fails; the adjacent material key (\( V3 \)) determines whether solid element faces are included for erosion along free boundaries.

Rigid body contact (RNTR and ROTR) is normally used for multi-body dynamics. For rigid body contact, a user defined force deflection curve is used to resist penetration rather than a linear stiffness. Therefore, for rigid body contact, you must specify a data curve ID (\( V1 \)), a force calculation method type (\( V2 \)) applicable to the specified rigid contact option, and an unloading stiffness value (\( V3 \)).

Tiebreak contact (TSTS and TNTS) is used to define contact failure when surfaces are ‘glued’ together. For both types of tiebreak contact, tensile (\( V1 \)) and shear (\( V2 \)) failure stresses are used to calculate the failure criterion and are required for input. For tiebreak nodes-to-surface contact, exponents for the normal force (\( V3 \)) and the shear force (\( V4 \)) can also be specified to define the failure criterion.

Drawbead contact (DRAWBEAD) is typically used in specialized cases to represent drawbeads, which help restrain a blank during drawing operations. In drawbead contact, you must input a load curve ID (\( V1 \)), which gives the bending component of the restraining force as a function of drawbead displacement, and a drawbead depth (\( V3 \)). You may optionally include a curve ID (\( V2 \)), which gives the normal restraining force as a function of drawbead displacement and the number of equally spaced integration points (\( V4 \)) along the drawbead.

**STEP 5: Specifying Birth and Death Times**

For each contact definition, you can specify a birth time and a death time by using fields **BTIME** and **DTIME** on the **EDCGEN** command. This allows you to activate contact at any time during the transient analysis, then deactivate it at a later time.

**6.1.1. Listing, Plotting and Deleting Contact Entities**

Once you define contact using the **EDCGEN** command, you can list, plot, or delete contact definitions. Use the **EDCLIST** command to list all currently defined contact entities. A sample listing is shown below. Each contact definition is given a contact entity reference number that can be used when plotting contact definitions.

```
CURRENT EXPLICIT DYNAMIC CONTACT DEFINITIONS
1 General Surface to Surface Contact Defined Between:
   Nodal contact Entity  N1
   And Nodal Target Entity  N2
   FS = 0.10000  FD = 0.080000  DC = 0.000000  VC = 0.000000  VDC = 0.0000
2 Automatic Single Surface Contact Defined For:
```
You can use the `EDPC` command to select and plot contact entities. The plot will consist of nodes or elements, depending on the method that was used to define the contact surfaces (that is, components or parts). You can specify a minimum contact entity number, maximum contact entity number, and contact entity number increment by using the `MIN`, `MAX`, and `INC` fields of the `EDPC` command. Hence, by issuing the command `EDPC,1,2,1` for the above listing, the defined entities for the STS and ASSC contact would be selected and plotted. It is important to note that for single surface contact definitions, all external surfaces in the model will be selected and plotted.

**Note**

`EDPC` selects the nodes and/or elements of the specified contact entities. Therefore, after plotting the contact entities, you must reselect all nodes and elements required for subsequent operations (such as `SOLVE`). Use the `NSEL,ALL` and `ESEL,ALL` commands (or other appropriate forms of these commands).

If contact has been incorrectly specified, you can delete it using the `EDDC` command. To delete a specific contact definition, issue the command `EDDC,DELE,Ctype,Cont,Targ`, where `Ctype` is the contact type and `Cont` and `Targ` are the contact and target parts or components for which contact has been defined. To delete all current contact definitions, issue the command `EDDC,DELE,ALL`.

You cannot delete contact specifications in a small restart analysis (`EDSTART,2`). However, you can use the `EDDC` command to deactivate (`EDDC,DACT,Ctype,Cont,Targ`) or reactivate (`EDDC,RACT,Ctype,Cont,Targ`) contact in a small restart. This capability is useful when you know at what stages in an analysis certain types of contact will occur. To use this feature you must define all contact specifications (`EDCGEN` command) in the new analysis, and you must perform at least one small restart. Once a contact specification is defined, you can deactivate it in a restart when it is not needed, and reactivate it in a later restart when it is needed. By considering contact only when it is needed, you can achieve significant CPU time savings. (Remember that if you delete contact in a new analysis using `EDDC,DELE`, its definition is removed from the database and you cannot reactivate it in a later restart.)

The `EDDC` command is not supported in an explicit dynamic full restart analysis (`EDSTART,3`). Thus, you cannot delete, deactivate, or reactivate contact specifications in a full restart that were defined in a previous analysis.

### 6.2. Contact Options

In order to adequately characterize the complex interaction between surfaces in an explicit dynamic analysis, twenty-four different contact types have been incorporated in the ANSYS LS-DYNA program (see the table below). With such a large number of contact options, it is extremely important for a user to have an understanding of each contact type so that the proper contact option can be used to realistically model a physical phenomenon. Therefore, in this section, each of the contact options available in the ANSYS LS-DYNA program will be discussed.

#### Table 6-1 Contact Types

<table>
<thead>
<tr>
<th></th>
<th>Single surface</th>
<th>Nodes to surface</th>
<th>Surface to surface</th>
</tr>
</thead>
<tbody>
<tr>
<td>General (Normal)</td>
<td>SS</td>
<td>NTS</td>
<td>STS, OSTS</td>
</tr>
<tr>
<td>Automatic</td>
<td>ASSC, AG, ASS2D</td>
<td>ANTS</td>
<td>ASTS</td>
</tr>
<tr>
<td>Rigid</td>
<td></td>
<td>RNTR</td>
<td>ROTR</td>
</tr>
</tbody>
</table>
### 6.2.1. Definition of Contact Types

As shown in the columns of Table 6-1, you can choose from three basic contact types in the ANSYS LS-DYNA program: single surface contact, nodes-to-surface contact, and surface-to-surface contact.

1. **Single Surface Contact (SS, ASSC, AG, ASS2D, ESS, SE)**

   Single surface contact is established when a surface of one body contacts itself or the surface of another body. In single surface contact, the ANSYS LS-DYNA program automatically determines which surfaces within a model may come into contact. Therefore, single surface contact is the simplest type to define because no contact or target surface definitions are required. When it is defined, single surface contact allows all external surfaces within a model to come into contact. This option can be very powerful for self-contact or large deformation problems when general areas of contact are not known beforehand. Unlike implicit modeling, where over-defining contact will significantly increase computation time, using single surface contact in an explicit analysis will cause only minor increases in CPU time. Most impact and crash-dynamic applications will require single surface contact to be defined. Since automatic general (AG) contact is very robust and includes shell edge (SE) contact as well as improved beam contact, it is recommended as the first choice for self contact and large deformation problems when the contact conditions are not easy to predict.

2. **Node-to-Surface Contact (NTS, ANTS, RNTR, TDNS, TNTS, ENTS, DRAWBEAD, FNTS)**

   Node-to-surface contact is a contact type which is established when a contacting node penetrates a target surface. This type of contact is commonly used for general contact between two surfaces. Use the same rules as in ANSYS implicit to determine which surfaces are target or contact:
   - The flat or concave surface is the target. The convex surface is the contact surface.
   - The coarser mesh is the target surface. The finer mesh is the contact surface.

   In the case of Drawbead contact, the bead is always the nodal contact surface and the blank is always the target surface.

3. **Surface-to-Surface Contact (STS, OSTS, ASTS, ROTR, TDSS, TSTS, ESTS, FSTS, FOSS, TSES)**

   Surface-to-surface contact is established when a surface of one body penetrates the surface of another body. Surface-to-surface contact is the most general type of contact as it is commonly used for bodies that have arbitrary shapes with relatively large contact areas. This type of contact is most efficient for bodies that experience large amounts of relative sliding, such as a block sliding on a plane or a sphere sliding within a groove.

### 6.2.2. Definition of Contact Options

For each of the three contact types, there are often several contact options available. In the ANSYS LS-DYNA program, the following options may be used:

<table>
<thead>
<tr>
<th></th>
<th>Single surface</th>
<th>Nodes to surface</th>
<th>Surface to surface</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tied</strong></td>
<td></td>
<td>TDNS</td>
<td>TDSS, TSES</td>
</tr>
<tr>
<td><strong>Tied with failure</strong></td>
<td></td>
<td>TNTS</td>
<td>TSTS</td>
</tr>
<tr>
<td><strong>Eroding</strong></td>
<td>ESS</td>
<td>ENTS</td>
<td>ESTS</td>
</tr>
<tr>
<td><strong>Edge</strong></td>
<td>SE</td>
<td>DRAWBEAD</td>
<td></td>
</tr>
<tr>
<td><strong>Forming</strong></td>
<td></td>
<td>FNTS</td>
<td>FSTS, FOSS</td>
</tr>
</tbody>
</table>

---

**Release 12.1 - © 2009 SASIP, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.**
6.2.2. Definition of Contact Options

1. General (Normal) Contact (SS, NTS, STS, OSTS)

   Although the general contact options use the simplest contact algorithms, they still are used for a wide range of applications. In fact, the NTS and STS options are two of the three recommended options for ANSYS LS-DYNA. The primary advantage of using the general contact algorithms is that they are extremely fast and robust. The only concern when working with the general contact options deals with contact surface orientation. The contact surface orientation defines which side of a surface is solid and which side is 'air'. When using solid elements, ANSYS LS-DYNA automatically sets the contact orientation correctly for the general contact options. For shell elements, however, you must set the contact surface orientation when using general contact. To do this, set the ORIE field of the EDCONTACT command to 2 to activate automatic reorientation of the contact surfaces. It is important to note, the reorientation will occur only if there is no initial penetration of the shell surface.

2. Automatic Contact (ASSC, AG, ASS2D, ANTS, ASTS)

   Along with the general contact family, the automatic contact options are the most commonly used contact algorithms. The main difference between the automatic and general options is that the contact surface orientation for shell elements is automatically determined by the automatic contact algorithms. In automatic contact, checks are made for contact on both sides of shell elements. Therefore, the contact search depth is always limited. If this is considered to be the reason for breaking through a contact surface, try using general contact with an infinite or a large search depth. See Controlling Contact Depth (p. 64) later in this chapter.

3. Eroding Contact (ESS, ENTS, ESTS)

   The eroding contact options are needed when the elements forming one or both exterior surfaces experience material failure during contact. Contact is allowed to continue with the remaining interior elements. Eroding contact should be used with solid elements in penetration problems and other applications which experience surface failure. When using eroding contact, a symmetry plane option (V1), an interior/exterior node erosion option (V2), and an adjacent material treatment option (V3) must be specified on the EDCGEN command.

4. Rigid Contact (RNTR, ROTR)

   The rigid body contact options are similar to the general contact options NTS and OSTS except that, rather than using a linear stiffness to resist penetration, a user-defined force-deflection curve is used. These contacts are typically used for multi-body dynamics where all of the bodies are rigid. Rigid body contacts are beneficial because they can allow inclusion of energy absorption without the need for modeling deformable elements. However, rigid body contact (RNTR, ROTR) cannot be used with deformable bodies. The contact of a rigid body to a deformable body must be defined with general, automatic, or eroding options. For the rigid options, the data curve ID (V1), force calculation method (V2), and unloading option (V3) must be specified on the EDCGEN command.

5. Tied Contact (TDNS, TDSS, TSES)

   The tied contact options actually 'glue' the contact nodes (surfaces) to the target surfaces. The contact and target surfaces must be initially coplanar so that during initialization, an isoparametric position of the contact node (surface) within the target segment is calculated. Thereafter, upon application of loads or initial velocities, the contact nodes (surfaces) are forced to maintain their isoparametric position within the target surface. The effect of tied contact is that the target surfaces can deform and the slave nodes are forced to follow that deformation. When defining tied contact, the body with the coarser mesh should always be defined as the target surface. Only translational degrees of freedom (UX, UY, and UZ) are affected by tied contact.

6. Tiebreak (Tied with Failure) Contact (TNTS, TSTS)
Tiebreak contact is identical to tied contact except that the contact nodes (surfaces) are tied to the target surfaces only until a failure criterion is reached. This is done by 'pinning' the contact nodes (surfaces) to the target using a penalty stiffness; after the failure criterion is exceeded, the contact nodes (surfaces) are allowed to slide relative to, or separate from, the target surface. The tiebreak contact options are typically used to represent spot-welded or bolted connections. The main difference between TNTS and TSTS is that the TSTS failure is based on a failure stress and the TNTS is based on a failure force. When using TSTS, the normal (V1) and shear (V2) failure stresses must be specified using the EDCGEN command. For the TNTS option, the normal (V1) and shear (V2) failure forces and normal (V3) and shear (V4) force exponents must be supplied using the EDCGEN command. Failure of the connection will occur when:

\[
\left( \frac{|f_n|}{f_{n_{\text{fail}}}} \right)^{m_1} + \left( \frac{|f_s|}{f_{s_{\text{fail}}}} \right)^{m_2} \geq 1
\]

7. Edge Contact (SE)

The single edge contact option should be used when contact occurs orthogonal to the shell surface normal direction. This contact option does not require contact or target surface definitions. Single edge contact is often used in sheet metal applications which have their surface normals orthogonal to the impact direction.

8. Drawbead Contact (DRAWBEAD)

Drawbead Contact is typically used in metal forming operations in which special care must be taken to restrain the blank. During drawing and stamping simulations, it is common for the blank to lose contact with the forming surfaces. Drawbead contact, which allows implementation of bending and frictional restraining forces, helps to ensure that the blank will remain in contact the entire length of the drawbead depth.
9. Forming Contact (FNTS, FSTS, FOSS)

The forming contact options are primarily used in metal forming applications. For these contact types, the tools and dies are typically defined as the target (master), while the workpiece is defined as the contact (slave) surface. Mesh connectivity is not required for the forming contact option, but the orientation of the tooling meshes must be in the same direction. The forming contact options are based on the Automatic contact types and are therefore very robust in metal forming applications.

6.3. Contact Search Methods

There are two different contact search algorithms used by ANSYS LS-DYNA for determining which target surface is being contacted by which contact surface. These methods are outlined below:

6.3.1. Mesh Connectivity Tracking

In mesh connectivity tracking, the contact search algorithm uses shared nodes of neighboring element segments to identify possible sources of contact. Therefore, when a target segment is no longer in contact with a contact surface node, the neighboring segments are checked. The mesh connectivity method is beneficial because it is very fast, but has the disadvantage of requiring that the mesh must be continuous for the contact algorithm to work correctly. Therefore, you should specify different contact sets for distinct regions. The mesh connectivity method is used by the NTS, OSTS, TSTS, TNTS, and TDNS contact options. However, by setting the $SHTK$ field of the EDCONTACT command to be positive, contact options NTS, OSTS, and TDNS will use the bucket sort method.

6.3.2. Bucket Sort Method

The bucket sort algorithm, which is used by all the contact options other than those specified above, divides the target surface region into cubes (buckets). Contacting nodes can contact any segment of the target surface in the same bucket or adjacent buckets. The bucket sort method is extremely robust, but can be somewhat slower than the mesh connectivity tracking method if the target surface contains a large number of elements.

6.3.3. Limiting the Contact Search Domain

You can limit the overall region for which a contact search will be executed by defining a contact box domain. When you specify a contact box, a contact search will only be performed within the volume specified by
the coordinates of the box. An advantage of this feature is that it minimizes CPU time when you know the
potential contact area of two bodies beforehand. The contact box option is only valid when defining contact
with parts or part assemblies. The contact box is defined using the EDBX command.

EDBX, Option, BOXID, XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX

Option ADD, DELETE, LIST
BOXID User defined List ID number
XMIN Minimum x coordinate
XMAX Maximum x coordinate
YMIN Minimum y coordinate
YMAX Maximum y coordinate
ZMIN Minimum z coordinate
ZMAX Maximum z coordinate

Once defined, a BOXID can be used in the BOXID1 and BOXID2 fields of the EDCGEN command. BOXID1
corresponds to the contact box and BOXID2 corresponds to the target contact box.

EDCGEN, NTS, Cont, Targ, ... DTIME, BOXID1, BOXID2

6.4. Special Considerations for Shells

Modeling rigid bodies with shell elements requires special consideration. For the automatic nodes-to-surface
(ANTS), automatic single surface (ASSC), automatic surface-to-surface (ASTS), and single surface (SS) contact
definitions, shell thickness is taken into account for the contact formulation for both the determination of
the contact surface as well as the search depth. Therefore, it is necessary to ensure that realistic thicknesses
are specified for the rigid body shells. A thickness that is too small may result in the loss of contact, and a
value that is too large may result in degradation of speed in the contact sorting algorithm. The shell thickness
contact option field SHTK of the EDCONTACT command is ignored for the above contact types.

Setting the SHTK field of the EDCONTACT command to 1 or 2 has several affects on the NTS, STS, and OSTS
contact options. The first is that the contact depth is directly computed (see the equations in Controlling
Contact Depth (p. 64)) based on element thickness for shells and edge length for solids and cannot be con-
trolled by the user. Second, as explained above, the contact search algorithm becomes Bucket Sort, so mesh
connectivity is not necessary for the contact to work correctly.

6.5. Controlling Contact Depth

For the contact options STS, NTS, OSTS, TNTS, and TSTS, you must be sure that spurious contact is not
defined between components in a model. For these contact types, a contact depth of 1 x 10^10 (nearly infinite)
is assumed by the ANSYS LS-DYNA program. Therefore, any time a contact node passes behind a target
surface (or vice-versa), contact is defined and a force proportional to the contact depth is generated. In an
explicit dynamic analysis, it is common for unwanted spurious contact to be defined because of the geometry
of parts which may be in relative motion. In cases where the contact is not genuine, and the contact depth
is relatively large, the contact forces can become large and cause the model to become unstable. For this
reason, the ANSYS LS-DYNA program allows you to specify a maximum contact depth, beyond which the
contact penetrations are considered spurious and will be ignored. To control the contact penetration distance,
set the PENCHK field of the EDCONTACT command to a value of 1 or 2.
The `PENCHK` field discussed above controls penetration checking for all contact definitions in the model of types STS, NTS, OSTS, TNTS, and TSTS. You can change the penetration checking of each individual contact definition of these same types by using the `EDSP` command.

For all of the contact options other than STS, NTS, OSTS, TNTS, and TSTS, the contact penetration depth is automatically limited by the element thickness and cannot be adjusted by the user. The expressions for contact depth in shell and solid elements are listed below:

**shell elements:**

\[
\text{contact depth} = \min[\text{shell thickness}, 0.4 \times \min \text{ side length}, 0.5 \times \sqrt{\text{area}}]
\]

**solid elements:**

\[
\text{contact depth} = \min\left[\frac{\text{volume}}{\text{area}}, 0.5 \times \sqrt{\text{area}}\right]
\]

### 6.6. Contact Stiffness

#### 6.6.1. Choice of Penalty Factor

A stiffness relationship between two bodies must be established for contact to occur. Without a contact stiffness, bodies will pass through one another. The relationship is generated through an 'elastic spring' that is put between the two bodies, where the contact force is equal to the product of the contact stiffness (k) and the penetration (δ). The amount of penetration (δ), or incompatibility, between the two bodies is therefore dependent on the stiffness k. Ideally, there should be no penetration, but this implies that \( k = \infty \), which will lead to numerical instabilities. The value of k that is used depends on the relative stiffness of the bodies in contact. In the ANSYS LS-DYNA program, the contact stiffness is determined by the following relationships:

\[
k = \frac{\text{fs} \times \text{Area}^2 \times K}{\text{Volume}} \quad \text{for segments on solid elements}
\]

\[
k = \frac{\text{fs} \times \text{Area} \times K}{\text{Minimum Diagonal}} \quad \text{for segments on shell elements}
\]

- Area = area of contact segment
- K = bulk modulus of contacted element
- fs = penalty factor (0.1 by default)

In almost all cases, the contact stiffness parameter automatically determined by the ANSYS LS-DYNA program will provide excellent results. The contact stiffness, however, can be changed for all contact surfaces by entering a new value for the penalty factor, fs, with the `SFSI` field of the `EDCONTACT` command. In practice, increasing the value of `SFSI` above 0.1 will typically cause instabilities to occur.

#### 6.6.2. Symmetry Stiffness

If the contact stresses of contact and target surfaces are badly mismatched due to large differences in material properties or element size, then instabilities or unrealistic behavior may arise. This can be avoided by
adjusting the penalty stiffness option, \textsc{PENO} of the \textsc{EDCONTACT} command. For example, by setting \textsc{PENO} to 1, both the contact and target stiffnesses are taken into account and the least stresses of the surfaces are used.

If there are very large differences in the contact stiffnesses, the program scales them to closer values, overwriting user input for the scale factor.

Together with the related mass, a closed contact element is a spring-mass system for which the current stability criterion yields a critical time step in the explicit analysis. The program reports the least one of the time steps with a message. If the actual time step size used is larger than the one listed in this message, then you should scale the penalty value of the offending surface with the \textsc{EDCONTACT} command or reduce the actual time step size with the \textsc{EDCTS} command. If the difference between the two time steps is small, this approach may not be necessary.

In addition to using the \textsc{SFI} and \textsc{PENO} options of the \textsc{EDCONTACT} command to control penalty stiffness between contact surfaces, individual contact (slave) and target (master) stiffness values can be adjusted using the \textsc{EDCMORE} command. Unlike the \textsc{EDCONTACT} command, which is applied to all contact definitions within a model, the \textsc{EDCMORE} command allows the specification of additional contact parameters for a single contact surface specification. The \textsc{Val1} and \textsc{Val2} fields of the \textsc{EDCMORE} command can be used to adjust the slave and master penalty stiffnesses in cases where the default contact stiffness determined by the ANSYS LS-DYNA program is found to be inadequate.

### 6.7. 2-D Contact Option

In order to define contact between \textsc{PLANE162} elements, a specialized 2-D contact option exists in the ANSYS LS-DYNA program. The option, \textsc{ASS2D}, is a single surface contact option (similar to \textsc{ASSC}). Similar to the contact used for the 3-D elements, the \textsc{ASS2D} option is specified using the \textsc{EDCGEN} command:

\texttt{EDCGEN, ASS2D, Cont., FS, FD, DC, ... BTIME, DTIME}

For 2-D contact, only \textit{part assemblies} can be used to define the contact component. The only other fields used with the \textsc{ASS2D} option are \textsc{FS}, \textsc{FD}, \textsc{DC}, \textsc{BTIME}, and \textsc{DTIME}. All of the other fields of the \textsc{EDCGEN} command will be ignored for 2-D contact.
Chapter 7: Material Models

ANSYS LS-DYNA includes over 40 material models that can be used to represent a wide range of material behavior. The available materials are listed below. The models and the procedure for using them are outlined in detail in the remainder of this chapter. For additional information on each material model, refer to Appendix B (p. 171) or to Chapter 16 of the LS-DYNA Theoretical Manual. (The LS-DYNA material number corresponding to each material model is shown in parentheses below, where applicable.)

Linear Elastic Models

- Isotropic (#1)
- Orthotropic (#2)
- Anisotropic (#2)
- Elastic Fluid (#1)

Nonlinear Elastic Models

- Blatz-Ko Rubber (#7)
- Mooney-Rivlin Rubber (#27)
- Viscoelastic (#6)

Nonlinear Inelastic Models

- Bilinear Isotropic (#3)
- Temperature Dependent Bilinear Isotropic (#4)
- Transversely Anisotropic Elastic Plastic (#37)
- Transversely Anisotropic FLD (#39)
- Bilinear Kinematic (#3)
- Plastic Kinematic (#3)
- 3-Parameter Barlat (#36)
- Barlat Anisotropic Plasticity (#33)
- Rate Sensitive Power Law Plasticity (#64)
- Strain Rate Dependent Plasticity (#19)
- Piecewise Linear Plasticity (rate dependent) (#24)
- Modified Piecewise Linear Plasticity (rate dependent) (#123)
- Composite Damage (#22)
- Concrete Damage (#72)
- Power Law Plasticity (#18)
• Elastic Viscoelastic Thermal Model (#106)

Pressure Dependent Plasticity Models

• Elastic-Plastic Hydrodynamic (# 10)
• Geological Cap (# 25)

Foam Models

• Closed Cell Foam (# 53)
• Viscous Foam (# 62)
• Low Density Foam (# 57)
• Crushable Foam (# 63)
• Honeycomb (# 26)

Models Requiring Equations of State

• Bamman Plasticity (# 51)
• Johnson-Cook Plasticity (# 15)
• Null (# 9)
• Zerilli-Armstrong (# 65)
• Steinberg (# 11)

Discrete Element Models

• Linear Elastic Spring
• General Nonlinear Spring
• Nonlinear Elastic Spring
• Elastoplastic Spring
• Inelastic Tension or Compression-Only Spring
• Maxwell Viscosity Spring
• Linear Viscosity Damper
• Nonlinear Viscosity Damper
• Cable (# 71)

Rigid Model

• Rigid (# 20)

7.1. Defining Explicit Dynamics Material Models

You define material models using the ANSYS commands MP, MPTEMP, MPDATA, TB, TBTEMP, and TBDATA, and the ANSYS LS-DYNA command EDMP. The next section, Explicit Dynamics Material Model Descriptions, describes how you define the individual material properties of each material model via commands.
Defining material models via the GUI is much more direct than doing so via commands:

1. Choose menu path **Main Menu> Preprocessor> Material Props> Material Models**. The **Define Material Model Behavior** dialog box appears.

**Note**

If you do not first define an ANSYS LS-DYNA element type, you will not have access to the ANSYS LS-DYNA material models.

2. In the **Material Models Available** window on the right, double-click on LS-DYNA, then on one of the material model categories: Linear, Nonlinear, Equation of State, Discrete Element Properties, or Rigid Material.

3. Double-click on a material subcategory. For example, under the Nonlinear category, the subcategories are: Elastic, Inelastic, and Foam Material Models.

4. Continue to double-click on any further material subcategories until a data input dialog box appears. The options that appear in this dialog box will be all of the individual material property options that are valid for the material model you have selected.

5. Enter the desired values, and click on **OK**. The model that you defined is listed by model type and number in the **Material Models Defined** window on the left.

You can later edit the values by double-clicking on the material model listed on the left side in the **Material Models Defined** window until the associated data input dialog box appears. There, you can edit the data and click on **OK** again.

You can copy the contents of an existing material model by choosing **Edit> Copy** and specifying the new model number. The copied material model is listed on the left side in the **Material Models Defined** window with the new number. Its contents are identical to the model whose contents you copied.

You can delete a material model by clicking on the model number to highlight it, then choosing **Edit> Delete**.

For further information on defining materials using the GUI, see **Material Model Interface** in the **Basic Analysis Guide**. Also, see **Using Tree Structures** in the **Operations Guide** for information specifically on the tree structure layout of the material model interface.

If you define, edit, copy, or delete material models via the GUI, ANSYS will automatically issue the correct commands and write them to the log file.

### 7.2. Explicit Dynamics Material Model Descriptions

This section describes each material model in detail. Wherever “load curve ID” is mentioned as required input, you actually input a material data curve ID. Material data curves are defined using the **EDCURVE** command as outlined in **Chapter 4, Loading (p. 33)**. All applicable properties for a material model appear in that model's dialog box when you are working interactively. The corresponding commands are provided here for when you are working in batch or command mode. Be sure to define all properties that are listed for the material model; do not define extraneous properties.
7.2.1. Linear Elastic Models

7.2.1.1. Isotropic Elastic Model

Isotropic material model. Use the MP command to input the required values:

\[
\begin{align*}
\text{MP, DENS} & \quad \text{density} \\
\text{MP, EX} & \quad \text{elastic modulus} \\
\text{MP, NUXY} & \quad \text{Poisson’s ratio}
\end{align*}
\]

For an example input listing, see Isotropic Elastic Example: High Carbon Steel (p. 173).

7.2.1.2. Orthotropic Elastic Model

Orthotropic material model. Use the MP command to input the required values.

\[
\begin{align*}
\text{MP, DENS} & \quad \text{density} \\
\text{MP, EX} & \quad \text{elastic moduli (also EY, EZ); one value required} \\
\text{MP, NUXY} & \quad \text{Minor Poisson’s ratio (also NUZ, NUX); one value required or MP, PRXY - Major Poisson’s ratio (also PRYZ, PRXZ); one value required} \\
\text{MP, GXY} & \quad \text{shear moduli (also GYZ and GXZ); one value required}
\end{align*}
\]

If only one value is specified for a given material property family (i.e., EX), the other values will be automatically defined (e.g., EY = EZ = EX). To specify a material coordinate system, use the EDLCS and EDMP, ORTHO commands. If no material coordinate system is specified, material properties are locally orthotropic with material axes defined by nodes I, J, and L of the element (see the figure below). For a multi-layer composite laminate, use the TB, COMP command instead and specify the lamina (ply) properties as SHELL163 real constants. See Composite Damage Model (p. 83) for more information.

For an example input listing, see Orthotropic Elastic Example: Aluminum Oxide (p. 174).

7.2.1.3. Anisotropic Elastic Model

This material description requires the full elasticity matrix. Because of symmetry, only 21 constants are required. This material is only valid for SOLID164 and SOLID168 elements.
Input density with MP. The constants are input in upper triangular form by means of the TB,ANEL command. To specify a material orientation axis, use the EDLCS and EDMP,ORTHO commands. If no material coordinate system is specified, material properties are locally orthotropic with material axes defined by nodes I, J, and L of the element (see figure of element coordinate system above).

\[
C = \begin{bmatrix}
C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\
C_{22} & C_{23} & C_{24} & C_{25} & C_{26} &
C_{33} & C_{34} & C_{35} & C_{36} &
C_{44} & C_{45} & C_{46} &
C_{55} & C_{56} &
C_{66}
\end{bmatrix}
\]

Symm

When you list out the data table information [TBLIST] for this material type, the constants will be listed in lower triangular form [D] instead of upper triangular form [C]. This discrepancy is not a computational error; the material data is correctly transferred to the LS-DYNA program.

For an example input listing, see Anisotropic Elastic Example: Cadmium (p. 174).

### 7.2.1.4. Elastic Fluid Model

Use this option to model containers filled with fluid that undergo dynamic impact loading. You input density using the MP command (DENS). Elastic fluid is then defined as a material type using the EDMP command:

\[
\begin{align*}
MP,\text{DENS} &
\text{EDMP,FLUID,}MAT,VAL1
\end{align*}
\]

The fluid model requires that the bulk modulus be specified. You can input the bulk modulus in the VAL1 field of the above command. As an alternative to inputting the bulk modulus directly using EDMP, you can input the elastic modulus (EX) and Poisson’s ratio (NUXY) using the MP command, and the program will calculate the bulk modulus as shown below.

\[
K = \frac{E}{3(1 - 2v)}
\]

If VAL1 (on EDMP), EX, and NUXY are all specified, VAL1 will be used as the bulk modulus.
7.2.2. Nonlinear Elastic Models

7.2.2.1. Blatz-Ko Rubber Elastic Model

The hyperelastic continuum rubber model defined by Blatz and Ko. This model uses the second Piola-Kirchhoff stress:

\[
S_{ij} = G \left( \frac{1}{V} C_{ij} - V \left( \frac{1}{1 - 2\nu} \right) \delta_{ij} \right)
\]

\( G \) is the shear modulus, \( V \) is the relative volume, \( \nu \) is the Poisson’s ratio, \( C_{ij} \) is the right Cauchy-Green strain tensor, and \( \delta_{ij} \) is the Kronecker delta. Use the MP command to input density (DENS) and shear modulus (GXY).

For an example input listing, see Blatz-Ko Example: Rubber (p. 174).

7.2.2.2. Mooney-Rivlin Rubber Elastic Model

Incompressible rubber material model; nearly identical to the 2-parameter existing ANSYS Mooney-Rivlin model. The strain energy density function is defined in terms of input parameters \( C_{10}, C_{01} \) and \( \nu \):

\[
W = C_{10} (I_1 - 3) + C_{01}(I_2 - 3) + C \left( \frac{1}{I_c} - 1 \right) + D(I_3 - 1)^2
\]

\[
C = \frac{C_{10}}{2} + C_{01}
\]

\[
D = \frac{C_{10}(5\nu - 2) + C_{01}(11\nu - 5)}{2(1 - 2\nu)}
\]

\( I_1, I_2, \) and \( I_3 \) are invariants of the right Cauchy-Green Tensor.

Input Poisson’s ratio (\( \nu \)) and density with the MP command. (For Poisson’s ratio, a value greater than .49 is recommended; smaller values may not work.) Input the Mooney-Rivlin constants with the TB and TBDATA commands. Data at only one temperature is permitted and must be specified in locations 1 and 2 for the data table:

\[
\text{TBDATA, 1, } C_{10}
\]

\[
\text{TBDATA, 2, } C_{01}
\]

As an alternative to inputting \( C_{10} \) and \( C_{01} \) directly, you can set these constants to zero and supply tabulated uniaxial data via a load curve. The program will calculate the constants based on the experimental data input in locations 3-6 of the TBDATA command. To use this input method, you must set the TBOPT field of the TB command to 2:

\[
\text{TB,MOONEY,,,2}
\]

\[
\text{TBDATA, 1, } C_{10} \text{ (set to zero to use experimental data)}
\]
**7.2.2. Nonlinear Elastic Models**

**TBDATA, 2, C₀₁** (set to zero to use experimental data)
**TBDATA, 3, C₃** (specimen gauge length L₀)
**TBDATA, 4, C₄** (specimen gauge width)
**TBDATA, 5, C₅** (specimen thickness)
**TBDATA, 6, C₆** (load curve ID)

The load curve definition that provides the uniaxial data should give the change in gauge length ∆L versus the corresponding force. In compression, both the force and the change in gauge length must be specified as negative values. In tension, the force and change in gauge length should be input as positive values. The principal stretch ratio in the uniaxial direction, λ₁, is then given by:

$$\lambda_1 = \frac{L_0 + \Delta L}{L_0}$$

where L₀ is the initial length and L is the actual length.

Alternatively, you can input the stress versus strain curve by setting the gauge length, thickness, and width equal to 1.0 and defining the engineering strain in place of the change in gauge length and the nominal (engineering) stress in place of the force.

The least squares fit to the experimental data is performed during the initialization phase of the solution in ANSYS LS-DYNA.

For an example input listing, see *Mooney-Rivlin Example: Rubber* (p. 174).

### 7.2.2.3. Viscoelastic Model

Linear viscoelastic material model introduced by Herrmann and Peterson. The model assumes the deviatoric behavior:

$$\sigma_{ij} = 2\int_0^1 \phi(t - \tau) \left[ \frac{\partial \varepsilon_{ij}^e(\tau)}{\partial \tau} \right] d\tau$$

where the shear relaxation modulus is given by:

$$\phi(t) = G_\infty + (G_0 - G_\infty) e^{-\beta t}$$

In the model, elastic bulk behavior is assumed when calculating the incrementally integrated pressure from the volume, V: p = K ln V. The parameters Gₐ∞, G₀, K (Bulk modulus) and β are required to define the linear viscoelastic material model. Input these values with TB, EVISC and locations 46, 47, 48, and 61 of the TBDATA command:

**TB, EVISC**
**TBDATA, 46, G₀**
**TBDATA, 47, G_∞**
**TBDATA, 48, K**
**TBDATA, 61, 1/β**
Chapter 7: Material Models

Note

For this material option, you must specify density (DENS) with the MP command.

For an example input listing, see Viscoelastic Example: Glass (p. 174).

7.2.3. Nonlinear Inelastic Models

7.2.3.1. Bilinear Isotropic Model

Classical bilinear isotropic hardening model (strain rate independent) that uses two slopes (elastic and plastic) to represent the stress-strain behavior of a material. Specify the stress-strain behavior at only one temperature. (A temperature dependent model is also available; see Temperature Dependent Bilinear Isotropic Model.) Input elastic modulus (EX), Poisson’s ratio (NUXY), and density (DENS) with the MP command. The program calculates the bulk modulus (K) using the EX and NUXY values that you input. Input the yield strength and tangent slope with TB,BISO and locations 1 and 2 of the TBDATA command:

\[ \text{TB,BISO} \]

TBDATA, 1, \( \sigma_Y \) (yield stress)

TBDATA, 2, \( E_{\text{tan}} \) (tangent modulus)

For an example input listing, see Bilinear Isotropic Plasticity Example: Nickel Alloy (p. 175).

7.2.3.2. Temperature Dependent Bilinear Isotropic Model

Classical bilinear isotropic hardening model that is strain rate independent and uses two slopes (elastic and plastic) to represent the stress-strain behavior of a material. The stress-strain behavior can be specified at up to six different temperatures. If the stress-strain behavior is specified at only one temperature, the Bilinear Isotropic (which is both strain rate and temperature independent) material model is assumed. You can use this model to represent a thermo-elastic material by inputting a large value for yield strength.

Input density (DENS) with the MP command. (The density is temperature independent.) Input elastic modulus (EX), Poisson’s ratio (NUXY), and thermal expansion coefficient (ALPX) with the MPTEMP and MPDATA commands. (These properties are temperature dependent.) Input the yield strength and tangent slope with the TB,BISO,,NTEMP command, TBTEMP commands, and locations 1 and 2 of the TBDATA command. The yield strength and tangent modulus must be defined relative to the same temperature data that are used on the MPTEMP command.

\[ \text{MP,DENS} \]

MPTEMP, 1, TEMP_1, TEMP_2, ..., TEMP_n-temp

MPDATA, EX_1, EX_1, EX_2, ..., EX_n-temp

MPDATA, NUXY_1, NUXY_1, NUXY_2, ..., NUXY_n-temp

MPDATA, ALPX_1, ALPX_1, ALPX_2, ..., ALPX_n-temp

TB, BISO,,NTEMP (NTEMP can be 2 to 6)

TBTEMP, TEMP_1 (first temperature point)

TBDATA, 1, \( \sigma_Y \) (yield stress)

TBDATA, 2, \( E_{\text{tan}} \) (tangent modulus)

TBTEMP, TEMP_2 (second temperature point)

TBDATA, 1, \( \sigma_Y \) (yield stress)

TBDATA, 2, \( E_{\text{tan}} \) (tangent modulus)

... (continue the pattern NTEMP times)

TBTEMP, TEMP_n-temp (last temperature point)
7.2.3.3. Transversely Anisotropic Hardening Model

Fully iterative anisotropic plasticity model available for shell and 2-D elements only. In this model the yield function given by Hill\[3\] (p. 195) is reduced to the following for the case of plane stress:

\[
F(\sigma) = \sqrt{\sigma_{11}^2 + \sigma_{22}^2 - \frac{2R}{R+1} \sigma_{11} \sigma_{22} + 2 \frac{2R+1}{R+1} \sigma_{12}^2}
\]

where \( R \) is the anisotropic hardening parameter which is the ratio of the in-plane plastic strain rate, \( \dot{\varepsilon}^P_{22} \), to the out of plane plastic strain rate, \( \dot{\varepsilon}^P_{33} \):

\[
R = \frac{\dot{\varepsilon}^P_{22}}{\dot{\varepsilon}^P_{33}}
\]

The stress-strain behavior can be specified at only one temperature. Input the elastic modulus (EX), density (DENS), and Poisson’s ratio (NUXY) with the MP command. Input the yield stress, tangent modulus, anisotropic hardening parameter, and a load curve ID for effective yield stress versus effective plastic strain with TB,PLAW,,,,7 and locations 1 - 4 of the TBDATA command:

TB, PLAW,,,,7
TBDATA, 1, \( \sigma_y \) (yield stress)
TBDATA, 2, \( E_{\text{tan}} \) (tangent modulus)
TBDATA, 3, \( R \) (anisotropic hardening parameter)
TBDATA, 4, LCID (load curve ID for yield stress vs. plastic strain)

For an example input listing, see Transversely Anisotropic Elastic Plastic Example: 1010 Steel (p. 175).

7.2.3.4. Transversely Anisotropic FLD Hardening Model

This material model is used for simulating the sheet metal forming of anisotropic materials. Only transversely anisotropic materials can be considered. For this model, the dependence of the flow stress with the effective plastic strain can be modeled using a defined load curve (using EDCURVE). In addition, you can define a forming limit diagram (also using EDCURVE). This diagram will be used by the ANSYS LS-DYNA program to compute the maximum strain ratio that the material can experience.

Note

For this material model, you must supply data over a temperature range wide enough to cover the actual temperatures that will occur in the analysis. Otherwise, the analysis will terminate.
This plasticity model is only available for shell and 2-D elements. The model directly follows the plasticity theory introduced in the Transversely Anisotropic Elastic Plastic model described earlier in this section. You can refer to that model for the theoretical basis.

To utilize the Transversely Anisotropic FLD model, you must input the density (DENS), elastic modulus (EX), and Poisson’s ratio (NUXY) with the MP command. As shown below, the additional input parameters are specified using TB,PLAW,,,,10 and locations 1-5 of the TBDATA command.

\[
\begin{align*}
\text{TB,PLAW,,,,10} \\
\text{TBDATA,1,} & \sigma_y \text{ (yield stress)} \\
\text{TBDATA,2,} & E_{\tan} \text{ (tangent modulus)} \\
\text{TBDATA,3,} & R \text{ (anisotropic hardening parameter)} \\
\text{TBDATA,4,} & \text{LCID1 (load curve for defining effective stress vs. plastic strain)} \\
\text{TBDATA,5,} & \text{LCID2 (load curve for defining FLD)}
\end{align*}
\]

For an example input listing, see Transversely Anisotropic FLD Example: Stainless Steel (p. 175).

### 7.2.3.5. Bilinear Kinematic Model

(strain rate independent)

Classical bilinear kinematic hardening model that uses two slopes (elastic and plastic) to represent the stress-strain behavior of a material. The stress-strain behavior can be specified at only one temperature. Input elastic modulus (EX), density (DENS), and Poisson’s ratio (NUXY) with the MP command. Input the yield strength and tangent slope with TB,BKIN and locations 1 and 2 of the TBDATA command:

\[
\begin{align*}
\text{TB,BKIN} \\
\text{TBDATA,1,} & \sigma_y \text{ (yield stress)} \\
\text{TBDATA,2,} & E_{\tan} \text{ (tangent modulus)}
\end{align*}
\]

For an example input listing, see Bilinear Kinematic Plasticity Example: Titanium Alloy (p. 176).

### 7.2.3.6. Plastic Kinematic Model

Isotropic, kinematic, or a combination of isotropic and kinematic hardening models with strain rate dependency and failure. Isotropic and kinematic contributions may be varied by adjusting the hardening parameter \( \beta \) between 0 (kinematic hardening only) and 1 (isotropic hardening only). Strain rate is accounted for using the Cowper-Symonds model which scales the yield stress by the strain rate dependent factor as shown below:
\[ \sigma_Y = \left[ 1 + \left( \frac{\dot{\varepsilon}}{C} \right)^{\frac{1}{P}} \right] (\sigma_0 + \beta E_p \varepsilon_{p \text{eff}}) \]

where \( \sigma_0 \) is the initial yield stress, \( \dot{\varepsilon} \) is the strain rate, \( C \) and \( P \) are the Cowper-Symonds strain rate parameters, \( \varepsilon_{p \text{eff}} \) is the effective plastic strain, and \( E_p \) is the plastic hardening modulus which is given by

\[ E_p = \frac{E_{\tan} E}{E - E_{\tan}} \]

The stress-strain behavior can only be specified at one temperature. Input the elastic modulus (EX), density (DENS), and Poisson's ratio (NUXY) with the MP command. Input the yield stress, tangent slope, hardening parameter, strain rate parameters \( C \) and \( P \), and the failure strain with TB,PLAW,,,,1 and locations 1 - 6 of the TBDATA command:

- TB,PLAW,,,,1
- TBDATA, 1, \( \sigma_Y \) (yield stress)
- TBDATA, 2, \( E_{\tan} \) (tangent modulus)
- TBDATA, 3, \( \beta \) (hardening parameter)
- TBDATA, 4, \( C \) (strain rate parameter)
- TBDATA, 5, \( P \) (strain rate parameter)
- TBDATA, 6, \( \varepsilon_f \) (failure strain)

For an example input listing, see Plastic Kinematic Example: 1018 Steel (p. 176).

### 7.2.3.7. 3-Parameter Barlat Model

Anisotropic plasticity model developed by Barlat and Lian[1,] (p. 195) used for modeling aluminum sheets under plane stress conditions. Both exponential and linear hardening rules are available. The anisotropic yield criterion for plane stress is defined as:

\[ 2(\sigma_Y)^m = a|K_1 + K_2|^m + a|K_1 - K_2|^m + c|2K_2|^m \]

where \( \sigma_Y \) is the yield stress, \( a \) and \( c \) are anisotropic material constants, \( m \) is Barlat exponent, and \( K_1 \) and \( K_2 \) are defined by:

\[ K_1 = \frac{\sigma_{xx} + h\sigma_{yy}}{2} \]

\[ K_2 = \sqrt{\left( \frac{\sigma_{xx} - h\sigma_{yy}}{2} \right)^2 + p^2 \varepsilon_{xy}^2} \]

where \( h \) and \( p \) are additional anisotropic material constants. For the exponential hardening option, the material yield strength is given by:

\[ \sigma_Y = k (\varepsilon_0 + \varepsilon_p)n \]
where \( k \) is the strength coefficient, \( \varepsilon_0 \) is the initial strain at yield, \( \varepsilon^p \) is the plastic strain, and \( n \) is the hardening coefficient. All of the anisotropic material constants, excluding \( p \) which is determined implicitly, are determined from Barlat and Lian width to thickness strain ratio (\( R \)) values as shown:

\[
a = 2 - 2 \sqrt{\frac{R_{00}}{1 + R_{00}} - \frac{R_{90}}{1 + R_{90}}}
\]

\[
c = 2 - a
\]

\[
h = \sqrt{\frac{R_{00} - \frac{1}{1 + R_{90}}}{\frac{R_{00}}{1 + R_{90}} - \frac{R_{90}}{R_{90}}}}
\]

The width to thickness strain ratio for any angle \( \Phi \) can be calculated from:

\[
R_\Phi = \frac{2m\sigma^m}{\frac{\partial \phi}{\partial \sigma_{xx}} + \frac{\partial \phi}{\partial \sigma_{yy}}} \sigma_\phi - 1
\]

Above, \( \sigma_\phi \) is the uniaxial tension in the \( \Phi \) direction. The stress-strain behavior can be specified at only one temperature. Input the elastic modulus (\( E_{XX} \)), density (\( DENS \)), and Poisson's ratio (\( NUXY \)) with the \texttt{MP} command. Input the hardening rule type, \( HR \) (equal to 1 for linear or 2 for exponential), tangent modulus (for \( HR = 1 \)) or strength coefficient (for \( HR = 2 \)), yield stress (for \( HR = 1 \)) or hardening coefficient (for \( HR = 2 \)), the Barlat exponent, \( m \), the width to thickness strain ratio values, \( R_{00} \), \( R_{45} \), and \( R_{90} \), and the orthotropic material axes with \texttt{TB,PLAW,,3} and locations 1 - 8 of the \texttt{TBDATA} command:

\begin{verbatim}
TB,PLAW,,3
TBDATA,1,HR (hardening rule type)
TBDATA,2,E_{tan} or k (tangent modulus or strength coefficient)
TBDATA,3,\sigma_Y or n (yield stress or hardening coefficient)
TBDATA,4,m (Barlat exponent)
TBDATA,5,R_{00}
TBDATA,6,R_{45}
TBDATA,7,R_{90}
TBDATA,8,CSID (defines orthotropic material axes)
\end{verbatim}

The last input item, \( CSID \), has two valid values: 0 (default) and 2. If \( CSID = 0 \), the local coordinate system will be defined by nodes I, J, and L of the element:
If CSID = 2, material axes are determined by a local coordinate system specified with the EDLCS command (see the command description for the details of how the axes are oriented). Before specifying the material property, you must define the local coordinate system using EDLCS, and then issue EDMP,ORTHO,VAL1, where VAL1 is the coordinate system ID from EDLCS.

For an example input listing, see 3 Parameter Barlat Example: Aluminum 5182 (p. 176).

### 7.2.3.8. Barlat Anisotropic Plasticity Model

Anisotropic plasticity model developed by Barlat, Lege, and Brem[2] (p. 195) used for modeling material behavior in forming processes. The anisotropic yield function $\Phi$ is defined as:

$$\Phi = |S_1 - S_2|^m + |S_2 - S_3|^m + |S_3 - S_1|^m$$

where $m$ is the flow potential exponent and $S_i$ are the principal values of the symmetric matrix $S_{ij}$.

$$S_{xx} = \frac{1}{3}[c(\sigma_{xx} - \sigma_{yy}) - b(\sigma_{zz} - \sigma_{xx})]$$
$$S_{yy} = \frac{1}{3}[a(\sigma_{yy} - \sigma_{zz}) - c(\sigma_{xx} - \sigma_{yy})]$$
$$S_{zz} = \frac{1}{3}[b(\sigma_{zz} - \sigma_{xx}) - a(\sigma_{yy} - \sigma_{zz})]$$
$$S_{yz} = f_{\sigma_{yz}}$$
$$S_{zx} = g_{\sigma_{zx}}$$
$$S_{xy} = f_{\sigma_{xy}}$$

where $a, b, c, f, g,$ and $h$ represent the anisotropic material constants. When $a=b=c=f=g=h=1$, isotropic material behavior is modeled and the yield surface reduces to the Tresca surface for $m = 1$ and the von Mises surface for $m = 2$ or $4$. For this material option, the yield strength is given by:

$$\sigma_y = k(\epsilon^p + \epsilon_0)^n$$

where $k$ is the strength coefficient, $\epsilon^p$ is the plastic strain, $\epsilon_0$ is the initial strain at yield, and $n$ is the hardening coefficient. The stress-strain behavior can be specified at only one temperature.

Input the elastic modulus (EX), density (DENS), and Poisson's ratio (NUXY) with the MP command. Input the strength coefficient, the initial strain at yield, the hardening coefficient, the flow potential exponent, and the Barlat anisotropic constants a-h with TB,PLAW,,,,6 and locations 1 - 10 of the TBDATA command.

TB,PLAW,,,,6
TBDATA 1, k (strength coefficient)
TBDATA 2, 0 (initial strain)
TBDATA 3, n (hardening coefficient)
TBDATA 4, m (flow potential (Barlat) exponent)
TBDATA 5, a
TBDATA 6, b
TBDATA 7, c
TBDATA 8, f
TBDATA 9, g
TBDATA 10, h
For an example input listing, see Barlat Anisotropic Plasticity Example: 2008-T4 Aluminum (p. 177).

### 7.2.3.9. Rate Sensitive Power Law Plasticity Model

Strain rate dependent plasticity model typically used for superplastic forming analyses. The material model follows a Ramburgh-Osgood constitutive relationship of the form:

\[ \sigma_{yy} = k \varepsilon_m \varepsilon_n \]

where \( \varepsilon \) is the strain, \( \dot{\varepsilon} \) is the strain rate, \( k \) is the material constant, \( m \) is the hardening coefficient, and \( n \) is the strain rate sensitivity coefficient. The stress-strain behavior can be specified at only one temperature. Input the elastic modulus (EX), density (DENS), and Poisson's ratio (NUXY) with the **MP** command. Input the material constant, hardening coefficient, strain rate sensitivity coefficient, and initial strain rate with **TB,PLAW,,,,4** and locations 1 - 4 of the **TBDATA** command:

\[ \begin{align*}
\text{TBDATA, 1, } k & \text{ (material constant)} \\
\text{TBDATA, 2, } m & \text{ (hardening coefficient)} \\
\text{TBDATA, 3, } n & \text{ (strain rate sensitivity coefficient)} \\
\text{TBDATA, 4, } \dot{\varepsilon}_0 & \text{ (initial strain rate)}
\end{align*} \]

For an example input listing, see Rate Sensitive Powerlaw Plasticity Example: A356 Aluminum (p. 177).

### 7.2.3.10. Strain Rate Dependent Plasticity Model

Strain rate dependent isotropic plasticity model used mainly for metal and plastic forming analyses.

In this model, a load curve is used to describe the initial yield strength, \( \sigma_0 \), as a function of effective strain rate. The yield stress for this material model is defined as:

\[ \sigma_Y = \sigma_0 \dot{\varepsilon} + E_h \dot{\varepsilon}_p \]

where \( \sigma_0 \) is the initial yield strength, \( \dot{\varepsilon}' \) is the effective strain rate, \( \dot{\varepsilon}_p \) is the effective plastic strain, and \( E_h \) is given by:

\[ E_h = \frac{E E_{\tan}}{E - E_{\tan}} \]

The stress-strain behavior can be specified at only one temperature. Input the elastic modulus (EX), density (DENS), and Poisson's ratio (NUXY) using the **MP** command. Input the load curve ID for defining the initial yield stress vs. effective strain rate, the tangent modulus, the load curve ID for defining the elastic modulus vs. effective strain rate (optional), the load curve ID for defining the tangent modulus vs. effective strain rate (optional), and the load curve ID for defining the von Mises stress at failure vs. effective strain rate using **TB,PLAW,,,,5** and locations 1 - 5 of the **TBDATA** command. For shell elements, you have the option of specifying Mn Time in location 6, instead of LCID4 in location 5, to define failure of the material. Mn Time is the minimum time step size for automatic element deletion.

\[ \begin{align*}
\text{TB,PLAW,,,,5} \\
\text{TBDATA, 1, } \text{LCID1 (load curve ID for defining the initial yield stress vs. effective strain rate)}
\end{align*} \]


**7.2.3. Nonlinear Inelastic Models**

**7.2.3.11. Piecewise Linear Plasticity Model**

This model provides a multilinear elastic-plastic material option that allows stress vs. strain curve input and strain rate dependency. This is a very commonly used plasticity law, especially for steel. Failure based on plastic strain can also be modeled with this material.

The piecewise linear plasticity model provides three different methods to account for the strain rate. The first method utilizes the Cowper-Symonds model, which scales the yield stress as shown:

\[
\sigma_y(\dot{\varepsilon}_{\text{eff}}, \ddot{\varepsilon}_{\text{eff}}) = \sigma_y(\dot{\varepsilon}_{\text{eff}}) \left[ 1 + \left( \frac{\dot{\varepsilon}_{\text{eff}}}{C} \right)^P \right]
\]

where \( \dot{\varepsilon} \) is the effective plastic strain rate, \( C \) and \( P \) are strain rate parameters, and \( \sigma_y(\dot{\varepsilon}_{\text{eff}}) \) is the yield stress without considering strain rate. This yield stress quantity can be input as either \( \sigma_y \) and \( E_{\text{tan}} \) (TBDATA entries 1 and 2) or as a total true stress vs. effective plastic strain curve, LCID1 (TBDATA entry 6).

The second method uses the LCID2 curve (TBDATA entry 7) to scale the yield stress with respect to strain rate. If LCID2 is input, it is used instead of the Cowper-Symonds scaling term defined by the strain rate parameters \( C \) and \( P \).

The third method used to account for the strain rate dependency requires that you define separate stress vs. effective plastic strain curves at specific strain rates. Using TBDATA entries 8 - 27, you may input stress vs. effective plastic strain behavior for up to ten different strain rates (Rate1 and LCID3, Rate2 and LCID4, etc.). It is important that you input all strain rate definitions in ascending order. Note that if Rate1 and LCID3 are specified, TBDATA entries 4-7 (C, P, LCID1, and LCID2) are automatically ignored.

Input the elastic modulus (EX), density (DENS), and Poisson’s ratio (NUXY) with the **MP** command. Input the yield stress, the tangent modulus, the effective plastic strain at failure, and the remaining parameters (listed below) with **TB,PLAW,,,,8** and locations 1 - 27 of the **TBDATA** command:

- **TB, PLAW,,,,8**
- **TBDATA,1, \sigma_y** (yield stress)
- **TBDATA,2, E_{\text{tan}}** (tangent modulus)
- **TBDATA,3, f** (effective plastic strain at failure)
- **TBDATA,4, C** (strain rate parameter)
- **TBDATA,5, P** (strain rate parameter)
- **TBDATA,6, LCID1** (load curve defining total true stress vs. effective plastic strain)
- **TBDATA,7, LCID2** (load curve that scales LCID1 with respect to strain rate)
- **TBDATA,8, Rate1** (strain rate value 1)
- **TBDATA,9, Rate2** (strain rate value 2)
- **TBDATA,10, LCID3** (stress vs. effective plastic strain curve for Rate1)
- **TBDATA,11, LCID4** (stress vs. effective plastic strain curve for Rate2)
### 7.2.3.12. Modified Piecewise Linear Plasticity Model

This model provides a multilinear elastic-plastic material option that allows stress vs. strain curve input and strain rate dependency. This model is similar to piecewise linear plasticity (TB,PLAW,,,,8), but has an enhanced failure criteria. Failure is based on effective plastic strain, plastic thinning, or the major principal in-plane strain component (see TBDATA entries 8 - 10: EPSTHIN, EPSMAJ, and NUMINT).

The modified piecewise linear plasticity model uses the same three methods to account for the strain rate as the piecewise linear plasticity model (see Piecewise Linear Plasticity Model (p. 81) for details). To account for strain rate, do one of the following:

- Input the strain rate parameters C and P (TBDATA entries 4 and 5) to use the Cowper-Symonds model.
- Input the LCID2 curve (TBDATA entry 7) to scale the yield stress with respect to strain.
- Define separate stress vs. strain curves for different strain rates (TBDATA entries 11 - 30).

Input the elastic modulus (EX), density (DENS), and Poisson’s ratio (NUXY) with the MP command. Input the remaining parameters with TB,PLAW,,,,11 and locations 1 - 30 of the TBDATA command:

TB, PLAW,,,,11
TBDATA,1, σ_Y (yield stress)
TBDATA,2, E_{tan} (tangent modulus)
TBDATA,3, f (effective plastic strain at failure)
TBDATA,4, C (strain rate parameter)
TBDATA,5, P (strain rate parameter)
TBDATA,6, LCID1 (load curve defining total true stress vs. effective plastic strain)
TBDATA,7, LCID2 (load curve that scales LCID1 with respect to strain rate)
TBDATA,8, EPSTHIN (thinning plastic strain at failure)
7.2.3. Nonlinear Inelastic Models

If load curve LCID1 (total true stress vs. effective plastic strain) is input, the yield stress \( \sigma_Y \) and tangent modulus \( E_{tan} \) values will be overwritten. If LCID2 (the scaling curve) is input, strain rate parameters C and P will be overwritten. If C and P, LCID2, and Rate1 (first strain rate value) are all set to zero, strain rate dependency is ignored.

For an example input listing, see Modified Piecewise Linear Plasticity Example: PVC (p. 178).

### 7.2.3.13. Composite Damage Model

Material model developed by Chang & Chang\[4\] (p. 195) for the failure of composite materials. The following five parameters are used in the model:

- \( S_1 \) = longitudinal tensile strength
- \( S_2 \) = transverse tensile strength
- \( S_{12} \) = shear strength
- \( C_2 \) = transverse compressive strength
- \( \alpha \) = nonlinear shear stress parameter

All parameters are determined experimentally. Input the elastic moduli (\( E_X, E_Y, E_Z \)), shear moduli (\( G_{XY}, G_{YZ}, G_{XZ} \)), density (DENS), and Poisson’s ratios (NUXY, NUYZ, NUXZ) with the MP command. Input the bulk modulus at compressive failure, the shear strength, the longitudinal tensile strength, transverse tensile strength, transverse compressive strength, and nonlinear shear stress parameter with TB,COMP and locations 1 - 6 of the TBDATA command:

| TBDATA, 1, KFAIL (bulk modulus of material in compressive failure) |
| TBDATA, 2, S12 (shear strength) |
| TBDATA, 3, S1 (longitudinal tensile strength) |
| TBDATA, 4, S2 (transverse tensile strength) |
| TBDATA, 5, C2 (transverse compressive strength) |
Note

Please refer to the *LS-DYNA Theoretical Manual* for more details about LS-DYNA Material model #22 (Composite Damage). This material model is required for multi-layer composite laminates, even if the failure capabilities are not used. The lamina (ply) properties are specified as SHELL163 real constants.

### 7.2.3.14. Concrete Damage Model

The Concrete Damage model is used to analyze buried steel reinforced concrete subject to impact loadings. This model requires constants for the concrete and reinforced materials, and also requires tabulated equation of state constants. (Refer to *Equation of State Models* (p. 93) for a detailed description of the tabulated equation of state.) Input the density (DENS) and Poisson's ratio (NUXY) with the MP command. Input all values shown below with the TB,CONCR,,,,2 command and locations 1 – 78 of the TBDATA command:

- **TB,CONCR,,,,2**
- **TBDATA,** 1, $\sigma_f$ (maximum principal stress for failure)
- **TBDATA,** 2, $A_0$ (cohesion constant)
- **TBDATA,** 3, $A_1$ (pressure hardening coefficient)
- **TBDATA,** 4, $A_2$ (pressure hardening coefficient)
- **TBDATA,** 5, $A_0Y$ (cohesion for yield)
- **TBDATA,** 6, $A_1Y$ (pressure hardening coefficient for yield limit)
- **TBDATA,** 7, $A_2Y$ (pressure hardening coefficient for yield limit)
- **TBDATA,** 8, $A_1F$ (pressure hardening coefficient for failed material)
- **TBDATA,** 9, $A_2F$ (pressure hardening coefficient for failed material)
- **TBDATA,** 10, $B_1$ (damage scale factor)
- **TBDATA,** 11, $B_2$ (damage scale factor for uniaxial tension)
- **TBDATA,** 12, $B_3$ (damage scale factor for triaxial tension)
- **TBDATA,** 13, PER (percent reinforcement)
- **TBDATA,** 14, $E_r$ (elastic moduli of reinforcement)
- **TBDATA,** 15, $NUXY_r$ (Poisson's ratio of reinforcement)
- **TBDATA,** 16, $\sigma_y$ (initial yield stress)
- **TBDATA,** 17, $E_{tan}$ (tangent modulus)
- **TBDATA,** 18, LCP (load curve ID giving rate sensitivity for principal material)
- **TBDATA,** 19, LCR (load curve ID giving rate sensitivity for reinforcement)
- **TBDATA,** 20-32, $\lambda_{1-13}$ (tabulated damage functions 1-13)
- **TBDATA,** 33-45, $\eta_{1-13}$ (tabulated scale factors 1-13)
- **TBDATA,** 46, GAMA (temperature constant)
- **TBDATA,** 47, $E_0$ (initial internal energy)
- **TBDATA,** 48, $V_0$ (initial relative volume)
- **TBDATA,** 49-58, $v_1-v_{10}$ (volumetric strain data values 1-10; natural log of the relative volume)
- **TBDATA,** 59-68, $C_1-C_{10}$ (volumetric pressure values at $v_i$)
- **TBDATA,** 69-78, $T_1-T_{10}$ (temperature values at $v_i$)

### 7.2.3.15. Power Law Plasticity Model

Strain rate dependent plasticity model typically used for metal and plastic forming analyses. Elastoplastic behavior with isotropic hardening is provided with this model. The material model has a Power Law constitutive relationship which includes the Cowper-Symonds multiplier to account for strain rate:
\[ \sigma_Y = \left[ 1 + \left( \frac{\dot{\varepsilon}}{C} \right)^n \right] k \left[ \varepsilon_e + \varepsilon_{eff}^p \right]^n \]

where \( \dot{\varepsilon} \) is the strain rate, \( C \) and \( P \) are the Cowper-Symonds strain rate parameters, \( \varepsilon_e \) is the elastic strain to yield, \( \varepsilon_{eff}^p \) is the effective plastic strain, \( k \) is the strength coefficient, and \( n \) is the hardening coefficient. The stress-strain behavior can be specified at only one temperature. Input the elastic modulus (EX), density (DENs), and Poisson’s ratio (NUXY) with the MP command. Input the strength coefficient, hardening coefficient, and strain rate parameters \( C \) and \( P \) with TB,PLAW,,,2 and locations 1 - 4 of the TBDATA command:

TB, PLAW,,,2
TBDATA, 1, k (strength coefficient)
TBDATA, 2, n (hardening coefficient)
TBDATA, 3, C (strain rate parameter)
TBDATA, 4, P (strain rate parameter)

For an example input listing, see *Powerlaw Plasticity Example: Aluminum 1100* (p. 179).

### 7.2.3.16. Elastic Viscoplastic Thermal Model

This material allows an elastic viscoplastic material definition with thermal effects. The uniaxial stress-stress-strain curve has the form:

\[
\sigma(\varepsilon_{eff}^p, \varepsilon_{eff}) = \sigma_0 + Q_{r1} (1 - \exp(-C_{r1} \varepsilon_{eff}^p)) + Q_{r2} (1 - \exp(-C_{r2} \varepsilon_{eff}^p)) \\
+ Q_{x1} (1 - \exp(-C_{x1} \varepsilon_{eff}^p)) + Q_{x2} (1 - \exp(-C_{x2} \varepsilon_{eff}^p)) \\
+ V_k \dot{\varepsilon}_{eff}^p V_m
\]

where \( \sigma_0 \) is the initial yield stress, \( \varepsilon_{eff}^p \) is the effective plastic strain, and the other parameters are as described in the TBDATA input shown below.

Strain rate is accounted for using the Cowper and Symonds model (where \( C \) and \( P \) are the Cowper-Symonds strain rate parameters), which scales the yield stress with the factor:

\[
1 + \left( \frac{\dot{\varepsilon}_{eff}^p}{C} \right)^n
\]

To convert from these constants, set the viscoelastic constants, \( V_k \) and \( V_m \), to the following values:

\[
V_k = \sigma \left( \frac{1}{C} \right)^n \\
V_m = \frac{1}{P}
\]
You must input density (DENS) using the **MP** command. You must also input the elastic modulus (EX), Poisson's ratio (NUXY), and the coefficient of thermal expansion (ALPX) by either using the **MP** command or by inputting load curves in locations 13, 14, and 18 of the **TBDATA** command. You can use the load curve input to represent each quantity as a function of temperature. Input the remaining parameters (described below) with the **TB,PLAW,,,12** command and locations 1 - 18 of the **TBDATA** command.

\[
\text{TBDATA,12, LCID1 (load curve ID for total true stress vs. effective plastic strain (if specified, TBDATA 2-9 are ignored))}
\]

\[
\text{TBDATA,13, LCID2 (load curve ID for elastic modulus vs. temperature)}
\]

\[
\text{TBDATA,14, LCID3 (load curve ID for Poisson's ratio vs. temperature)}
\]

\[
\text{TBDATA,15, LCID4 (load curve ID for initial yield stress vs. temperature; if input, } \sigma_0 \text{ is ignored)}
\]

\[
\text{TBDATA,16, LCID5 (load curve ID for scaling factor on isotropic hardening parameters (QR1 and QR2) vs. temperature)}
\]

\[
\text{TBDATA,17, LCID6 (load curve ID for scaling factor on kinematic hardening parameters (QX1 and QX2) vs. temperature)}
\]

\[
\text{TBDATA,18, LCID7 (load curve ID for thermal coefficient of expansion vs. temperature)}
\]

---

**Note**

If the total true stress vs. effective plastic strain is input (LCID1 in location 12 of **TBDATA**), then **TBDATA** locations 2-9 are ignored. Also, if LCID2, LCID3, or LCID7 is specified, then the corresponding **MP** command is ignored. (For example, LCID2 overrides the command **MP,EX**.)

For an example input listing, see *Elastic Viscoplastic Thermal Example* (p. 179).

### 7.2.4. Pressure Dependent Plasticity Models

#### 7.2.4.1. Elastic-Plastic Hydrodynamic Model

This model is used for modeling materials undergoing large amounts of strain, where the plastic behavior can either be defined from a set of data points or from the yield stress and tangent modulus. If the effective true plastic strain and effective true stress data values are not specified, the yield strength will be calculated as shown below (based on isotropic hardening):

\[
\sigma_y = \sigma_0 + E_h \bar{\varepsilon}^p
\]

You can calculate the plastic hardening modulus, \( E_h \), in terms of the Young's modulus and tangent modulus:
If the effective true plastic strain and stress values are specified, the stress-strain behavior is defined by the data points along the effective true stress vs. true plastic strain curve. Up to 16 data points may be specified. Linear extrapolation will be used if the strain values exceed the maximum input values; therefore, you should input values to cover the full range of strains expected in the analysis. Input the density (DENS), elastic modulus (EX), and shear modulus (GXY) with the MP command. Input the parameters described below with TB,PLAW,,,9 and locations 1-45 of the TBDATA command:

- TB,PLAW,,,9
- TBDATA,1, \( \sigma_0 \) (initial yield stress)
- TBDATA,2, \( E_h \) (hardening modulus)
- TBDATA,3,PC (pressure cutoff value)
- TBDATA,4, \( f \) (failure strain)
- TBDATA,5-20, \( 1 \)-16 (effective strain data curve values)
- TBDATA,21-36, \( \sigma_1 \)-\( \sigma_{16} \) (effective stress data curve values)
- TBDATA,37,Co (linear polynomial equation of state constant)
- TBDATA,38,C1 (linear polynomial equation of state constant)
- TBDATA,39,C2 (linear polynomial equation of state constant)
- TBDATA,40,C3 (linear polynomial equation of state constant)
- TBDATA,41,C4 (linear polynomial equation of state constant)
- TBDATA,42,C5 (linear polynomial equation of state constant)
- TBDATA,43,C6 (linear polynomial equation of state constant)
- TBDATA,44,Eo (initial internal energy)
- TBDATA,45,Vo (initial relative volume)

Note that constants 37 through 45 specified on the TBDATA command are the same as the linear polynomial equation of state model. Please refer to *Equation of State Models* (p. 93) later in this chapter for more details.

### 7.2.4.2. Geological Cap Model

This model is an inviscid, two invariant geologic cap model which can be used for geomechanical problems or for materials such as concrete. In this model, the two invariant cap theory is extended to include nonlinear kinematic hardening. A discussion of the extended cap model and its parameters is given below.

The cap model is formulated in terms of the invariants of the stress tensor. The square root of the second invariant of the deviatoric stress tensor, \( \sqrt{J_{2D}} \), is found from the deviatoric stresses (\( s \)) as

\[
\sqrt{J_{2D}} = \sqrt{\frac{1}{2} S_{ij} S_{ij}}
\]

and is the objective scalar measure of the distortional or shearing stress. The first invariant of the stress, \( J_1 \), is the trace of the stress tensor.
The cap model consists of three surfaces in pressure $\sqrt{J_{2D}} - J_1$ space, as shown in Figure 7.1: Surface of the Two-invariant Cap Model (p. 88). Surface $f_1$ is the failure envelope, $f_2$ is the cap surface, and $f_3$ is the tension cutoff. The functional form of $f_1$ is

$$f_1 = \sqrt{J_{2D}} - \min(F_e(J_1), T_{\text{mises}})$$

where $F_e$ is given by

$$F_e(J_1) \equiv \alpha - \gamma \exp(-\beta J_1) + \theta J_1$$

and

$$T_{\text{mises}} \equiv |X(\kappa) - L(\kappa)|$$

This failure envelope surface is fixed in $\sqrt{J_{2D}} - J_1$ space, and therefore does not harden unless kinematic hardening is present. Next, there is a cap surface $f_2$ in the figure, with $f_2$ given by

$$f_2 = \sqrt{J_{2D}} - F_c(J_1, \kappa)$$

where $F_c$ is defined by

$$F_c(J_1, \kappa) \equiv \frac{1}{R} \sqrt{[X(\kappa) - L(\kappa)]^2 - [J_1 - L(\kappa)]^2}$$

$X(\kappa)$ is the intersection of the cap surface with the $J_1$ axis

$$X(\kappa) = \kappa + RF_e(\kappa)$$

and $L(\kappa)$ is defined by
The hardening parameter $\kappa$ is related to the plastic volume change $\varepsilon^p_v$ through the hardening law

$$\varepsilon^p_v = W \{1 - \exp[-D(X(\kappa) - X_0)]\}$$

Geometrically, $\kappa$ is seen in the figure as the $J_1$ coordinate of the intersection of the cap surface and the failure surface. Finally, there is the tension cutoff surface, denoted $f_3$ in the figure. The function $f_3$ is given by

$$f_3 \equiv T - J_1$$

where $T$ is the input material parameter which specifies the maximum hydrostatic tension sustainable by the material. The elastic domain in $\sqrt{J_2D} - J_1$ space is then bounded by the failure envelope surface above, the tension cutoff surface on the left, and the cap surface on the right.

Input the density (DENS) and shear modulus (GXY) with the MP command. Input the parameters described below with the TB,GCAP command and locations 1–13 of the TBDATA command.

- $\text{TBDATA,1,K}$ (bulk modulus)
- $\text{TBDATA,2,}\alpha$ (failure envelope parameter)
- $\text{TBDATA,3,}\theta$ (failure envelope linear coefficient)
- $\text{TBDATA,4,}\gamma$ (failure envelope exponential coefficient)
- $\text{TBDATA,5,}\beta$ (failure envelope exponent)
- $\text{TBDATA,6,R}$ (cap surface axis ratio)
- $\text{TBDATA,7,D}$ (hardening law exponent)
- $\text{TBDATA,8,W}$ (hardening law coefficient)
- $\text{TBDATA,9,X}_0$ (hardening law exponent)
- $\text{TBDATA,10,C}$ (kinematic hardening coefficient)
- $\text{TBDATA,11,N}$ (kinematic hardening parameter)
- $\text{TBDATA,12,Ftype}$ (formulation flag: use 1 for soil or concrete, 2 for rock)
- $\text{TBDATA,13,Toff}$ (tension cutoff value; $Toff < 0$, positive in compression)

Please refer to Material Model 25 in the LS-DYNA Theoretical Manual for more information about this material model.

For an example input listing, see Geological Cap Example: SRI Dynamic Concrete (p. 180).

### 7.2.5. Foam Models

#### 7.2.5.1. Closed Cell Foam Model

Rigid, closed cell, low density polyurethane foam material model often used for modeling impact limiters in automobile designs. This material model is very similar to the honeycomb model in that the components of the stress tensor are uncoupled until volumetric compaction is achieved. Unlike honeycomb, however,
the closed cell foam model is isotropic in nature and includes the effects of confined air pressure. The material model defines stress to be:

\[ \sigma_{ij} = \sigma_{ij}^{sk} + \delta_{ij} \left[ p_o \gamma / (1 + \gamma - \phi) \right] \]

where \( \sigma_{ij}^{sk} \) is the skeletal stress, \( p_o \) is the initial foam pressure, \( \phi \) is the ratio of foam to polymer density, \( \delta_{ij} \) is the Kronecker delta, and \( \gamma \) is the volumetric strain which is defined by:

\[ \gamma = V - 1 + \gamma_0 \]

where \( V \) is the relative volume and \( \gamma_0 \) is the initial volumetric strain. The yield condition is applied to the principal trial stresses and is defined by:

\[ \sigma_Y = a + b(1 + c\gamma) \]

where \( a, b, \) and \( c \) are user defined input constants. The stress-strain behavior can be specified at only one temperature. Input the elastic modulus (EX) and density (DENS) with the MP command. Poisson's ratio for this model is assumed to be zero. Input the yield stress constants \( a, b, \) and \( c \), the initial foam pressure, the ratio of foam to polymer density, and the initial volumetric strain with TB,FOAM,,,,1 and locations 1 - 6 of the TBDATA command:

TB, FOAM,,,,1
TBDATA, 1, a
TBDATA, 2, b
TBDATA, 3, c
TBDATA, 4, \( p_o \) (initial foam pressure)
TBDATA, 5, \( \phi \) (ratio of foam to polymer density)
TBDATA, 6, \( \gamma_0 \) (initial volumetric strain)

7.2.5.2. Viscous Foam Model

Energy absorbing foam material used in crash-simulation models. The model consists of a nonlinear elastic stiffness in parallel with a viscous damper. The elastic stiffness is intended to limit total crush while the viscosity absorbs energy. The elastic stiffness, \( E' \), and initial viscous coefficient, \( V' \), are both nonlinear functions of the relative volume, \( V \):

\[ E' = E_1 V^{-n_1} \]

\[ V' = V_2 |1 - V|^{n_2} \]

where \( E_1 \) is the initial elastic stiffness, \( V_2 \) is the initial viscous coefficient, and \( n_1 \) and \( n_2 \) are the Power Laws for the elastic stiffness and viscous coefficient, respectively. The stress-strain behavior can be specified at only one temperature. Input the initial elastic stiffness (EX), Poisson's ratio (NUXY), and density (DENS) with the MP command. Input the Power Law for the elastic stiffness, the initial viscous coefficient, the elastic stiffness for viscosity (required to prevent time step problems), and the pearly for viscosity with TB,FOAM,,,,3 and locations 1 - 4 of the TBDATA command:

TB, FOAM,,,,3
TBDATA, 1, \( n_1 \) (Power Law for the elastic stiffness)
7.2.5.3. Low Density Foam Model

Highly compressible (urethane) foam material model often used for padded materials such as seat cushions. In compression, the model assumes hysteresis unloading behavior with possible energy dissipation. In tension the material model behaves linearly until tearing occurs. For uniaxial loading, the model assumes that there is no coupling in transverse directions. By using input shape factor controls (a hysteresis unloading factor (HU), a decay constant ($\beta$), and a shape factor for unloading) the observed unloading behavior of foams can be closely approximated. The stress-strain behavior can be specified at only one temperature. Input the elastic modulus (EX) and density (DENS) with the MP command. Input the load curve ID for nominal stress vs. strain, the tension cutoff (tearing) stress, the hysteresis unloading factor, the decay constant, the viscous coefficient, the shape factor for unloading, the failure option when the cutoff stress is reached, and the bulk viscosity action flag with TB,FOAM,,,,2 and locations 1 - 8 of the TBDATA command:

TB, FOAM,,,,2
TBDATA, 1, LCID (load curve ID for stress-strain behavior)
TBDATA, 2, TC (tension cutoff stress, default=1E20)
TBDATA, 3, HU (hysteretic unloading factor: between 1.0 - no energy dissipation (default) and 0.0 - full energy dissipation)
TBDATA, 4, $\beta$ (decay constant)
TBDATA, 5, DAMP (viscous coefficient, values between 0.05 and 0.5 are recommended)
TBDATA, 6, SHAPE (shape unloading factor, default = 1)
TBDATA, 7, FAIL (failure option when cutoff stress is reached: 0.0 - tensile stress remains at cutoff value, 1.0 - tensile stress reset to zero)
TBDATA, 8, BVFLAG (bulk viscosity action flag: 0.0 - no bulk viscosity (recommended) and 1.0 - bulk viscosity active)

7.2.5.4. Crushable Foam Model

Material model used for crushable foams in side impact and other applications where cyclic behavior is unimportant. The foam model is strain rate dependent and crushes one-dimensionally with a Poisson's ratio that is essentially zero. In the formulation, the elastic modulus is considered constant and the stress is updated assuming elastic behavior:

$$\sigma_{ij}^{n+1} = \sigma_{ij}^{n} + E \epsilon_{ij}^{n+1/2} \Delta t^{n+1/2}$$

where $\dot{\epsilon}_{ij}$ is the strain rate, E is the elastic modulus, and t is time. The model includes a tensile stress cutoff value which defines failure under tensile loads. For stresses below the tensile cutoff value, the model predicts similar response between tensile and compressive loading. It is important to have a nonzero value for the cutoff stress to prevent deterioration of the material under small tensile loads. Input the elastic modulus (EX), density (DENS), and Poisson's ratio (NUXY) with the MP command. Input the load curve ID for defining stress vs. volumetric strain, the tension cutoff value, and the viscous damping coefficient with TB,FOAM,,,,4 and locations 1 - 3 of the TBDATA command:

TB, FOAM,,,,4
TBDATA, 1, LCID (load curve ID for defining yield stress vs. volumetric strain)
TBDATA, 2, TC (tension cutoff value)
TBDATA, 3, DAMP (viscous damping coefficient, values between 0.05 and 0.5 are recommended)
7.2.5.5. Honeycomb Foam Model

Orthotropic material model used for honeycomb materials. The behavior of the model before compaction is orthotropic where the components of the stress tensor are uncoupled. The elastic moduli are considered to vary linearly with relative volume as shown below:

\[
\begin{align*}
E_{aa} &= E_{aau} + \beta (E - E_{aau}) \\
E_{bb} &= E_{bbu} + \beta (E - E_{bbu}) \\
E_{cc} &= E_{ccu} + \beta (E - E_{ccu}) \\
G_{ab} &= G_{abu} + \beta (G - G_{abu}) \\
G_{bc} &= G_{bcu} + \beta (G - G_{bcu}) \\
G_{ca} &= G_{cau} + \beta (G - G_{cau})
\end{align*}
\]

where

\[
G = \frac{E}{2(1 + \nu)}
\]

is the elastic shear modulus for fully compacted honeycomb material.

and

\[
\beta = \max \left[ \min \left( \frac{1 - V}{1 - V_f}, 1 \right), 0 \right]
\]

\[V = \text{the relative volume (defined as ratio of current to original volume)}\]
\[V_f = \text{relative volume of the fully compacted honeycomb}\]

To specify a material coordinate system, use the EDLCS and EDMP,ORTHO commands. If no material coordinate system is specified, material properties are locally orthotropic with material axes defined by nodes I, J, and L of the element (see Orthotropic Elastic Model for a graphic depiction).

Load curves can be used to input the magnitude of the average stress as the relative volume changes. Each curve must have the same abscissa values. Curves can be defined either as a function of relative volume (V) or volumetric strain (1 - V). Input the elastic modulus (EX), density (DENS), and Poisson's ratio (NUXY) with the MP command. Input all values shown below with TB,HONEY and locations 1 - 17 of the TBDATA command:

TB, HONEY
TBDATA, 1, \sigma_Y (yield stress for fully compacted honeycomb)
TBDATA, 2, \nu (relative volume of fully compacted honeycomb)
TBDATA, 3, \mu (material viscosity coefficient, default = 0.05)
TBDATA, 4, E_{aau} (Elastic modulus in aa direction for uncompressed configuration)
TBDATA, 5, E_{bbu} (Elastic modulus in bb direction for uncompressed configuration)
TBDATA, 6, E_{ccu} (Elastic modulus in cc direction for uncompressed configuration)
TBDATA, 7, G_{abu} (Shear modulus in ab direction for uncompressed configuration)
TBDATA, 8, G_{bcu} (Shear modulus in bc direction for uncompressed configuration)
TBDATA, 9, G_{cau} (Shear modulus in cc direction for uncompressed configuration)
TBDATA, 10, LCA (Load curve ID for aa direction stress vs. relative volume or volumetric strain)
TBDATA, 11, LCB (Load curve ID for bb direction stress vs. relative volume or volumetric strain)
TBDATA, 12, LCC (Load curve ID for cc direction stress vs. relative volume or volumetric strain)
TBDATA, 13, LCS (Load curve ID for shear yield vs. relative volume or volumetric strain)
7.2.6. Equation of State Models

There are 3 different equation of state models available in ANSYS LS-DYNA:

- Linear polynomial
- Gruneisen
- Tabulated

These equation of state models are used in conjunction with certain material models, such as the Johnson-Cook Plasticity and Zerilli-Armstrong models. You identify the equation of state to be used by inputting the appropriate number in the `EOSOPT` field of the `TB` command.

7.2.6.1. Linear Polynomial Equation of State

(EOSOPT = 1)

This equation of state is linear in internal energy. The pressure is given by:

\[ P = C_0 + C_1 \mu + C_2 \mu^2 + C_3 \mu^3 + (C_4 + C_5 \mu + C_6 \mu^2)E \]

where terms \( C_2 \mu^2 \) and \( C_6 \mu^2 \) are set to zero if \( \mu \leq 0 \), and \( \rho / \rho_0 \) is the ratio of current density to initial density.

Input the required constants using the `TBDATA` command. The starting location (LOC) for the constants depends on which material model you are using. (See the description for the specific material model to find out what starting location value should be used.)

- `TBDATA, LOC+C_0`
- `TBDATA, LOC+1+C_1`
- `TBDATA, LOC+2+C_2`
- `TBDATA, LOC+3+C_3`
- `TBDATA, LOC+4+C_4`
- `TBDATA, LOC+5+C_5`
- `TBDATA, LOC+6+C_6`
- `TBDATA, LOC+7+E_0` (initial internal energy)
- `TBDATA, LOC+8+V_0` (initial relative volume)

7.2.6.2. Gruneisen Equation of State

(EOSOPT = 2)

This equation of state defines the pressure volume relationship in one of two ways, depending on whether the material is compressed or expanded.
The Gruneisen equation of state with cubic shock velocity-particle velocity defines pressure for compressed materials as:

\[
p = p_0 C_a \left[ 1 + \left( 1 - \frac{\gamma_0}{2} \right) \mu - \frac{a^2}{2} \mu^2 \right] + \left( \gamma_0 + a \mu \right) E
\]

and for expanded materials as:

\[
p = p_0 C_a + (\gamma_0 + a \mu) E
\]

where \( C \) is the intercept of the vs-vp curve; \( S_1, S_2, \) and \( S_3 \) are the coefficients of the slope of the vs-vp curve, \( \gamma_0 \) is the Gruneisen gamma, \( a \) is the first order volume correction to \( \gamma_0 \), and

\[
\mu = \frac{\rho}{\rho_0} - 1
\]

Input the required constants using the TBDATA command. The starting location (LOC) for the constants depends on which material model you are using. (See the description for the specific material model to find out what starting location value should be used.)

- \( \text{TBDATA}, \text{LOC}, C \)
- \( \text{TBDATA}, \text{LOC}+1, S_1 \)
- \( \text{TBDATA}, \text{LOC}+2, S_2 \)
- \( \text{TBDATA}, \text{LOC}+3, S_3 \)
- \( \text{TBDATA}, \text{LOC}+4, \gamma_0 \)
- \( \text{TBDATA}, \text{LOC}+5, A \)
- \( \text{TBDATA}, \text{LOC}+6, E_0 \) (initial internal energy)
- \( \text{TBDATA}, \text{LOC}+7, V_0 \) (initial relative volume)

**7.2.6.3. Tabulated Equation of State**

(EOSOPT = 3)

The tabulated equation of state model is linear in internal energy. Pressure is defined by:

\[
P = C_i (\epsilon_{vi}) + \gamma T_i (\epsilon_{vi}) E
\]

Input the required constants using the TBDATA command. The starting location (LOC) for the constants depends on which material model you are using. (See the description for the specific material model to find out what starting location value should be used.)

- \( \text{TBDATA}, \text{LOC}, \text{GAMA} \) (temperature constant)
- \( \text{TBDATA}, \text{LOC}+1, E_0 \) (initial internal energy)
- \( \text{TBDATA}, \text{LOC}+2, V_0 \) (initial relative volume)
- \( \text{TBDATA}, \text{LOC}+3 \text{ to } \text{LOC}+12, \epsilon_{vi}^{1-10} \) (volumetric strain data values 1-10, natural log of the relative volume)
- \( \text{TBDATA}, \text{LOC}+13 \text{ to } \text{LOC}+22, C_1-C_{10} \) (volumetric pressure values at \( \epsilon_{vi} \))
- \( \text{TBDATA}, \text{LOC}+23 \text{ to } \text{LOC}+32, T_1-T_{10} \) (temperature values at \( \epsilon_{vi} \))
7.2.6.4. Bamman Plasticity Model

This is a fairly complex material model used in metal forming processes with strain rate and temperature dependent plasticity. The Bamman model does not require an additional equation of state model (EOSOPT is not used) because internal equation of state variables are specified in fields 21 through 26 of the TBDATA command (as shown below).

Input the density (DENS), elastic modulus (EX) and Poisson’s ratio (NUXY) with the MP command. Input the parameters described below with TB,EOS and locations 1-26 of the TBDATA command:

\[
\begin{align*}
&\text{TB,EOS,,4} \\
&\text{TBDATA,1,T} \\&\text{TBDATA,2,HC} \text{ (heat generation coefficient)} \\
&\text{TBDATA,3-20,C}\_1-C\_18 \text{ (constants for flow stress)} \\
&\text{TBDATA,21-26,A}\_1-A\_6 \text{ (internal equation of state variables)}
\end{align*}
\]

For a complete discussion of the kinematics associated with this model, as well as a more detailed description of the required input constants, refer to the description of Material Model 51 in Chapter 16 of the LS-DYNA Theoretical Manual.

7.2.6.5. Johnson-Cook Plasticity Model

This model, also called the viscoplastic model, is a strain-rate and adiabatic (heat conduction is neglected) temperature-dependent plasticity model. This model is suitable for problems where strain rates vary over a large range, and temperature changes due to plastic dissipation cause material softening. The model may be used for shell and solid elements. For solid elements, you need an equation of state (discussed later in this section).

Johnson and Cook express the flow stress as

\[
\sigma_y = (A + B \varepsilon_p^n)\left(1 + c \ln \dot{\varepsilon}^*\right)\left(1 - T^*^m\right)
\]

where:

- \(A, B, C, n,\) and \(m\) = material constant
- \(\varepsilon_p\) = effective plastic strain
- \(\dot{\varepsilon}^* = \frac{\dot{\varepsilon}_p}{\dot{\varepsilon}_0}\)
- \(T^* = \text{homologous temperature} = \frac{T - T_{room}}{T_{melt} - T_{room}}\)

The strain at fracture is given by:

\[
\varepsilon_f = [D_1 + D_2 \exp D_3 \sigma^* \ln \dot{\varepsilon}^* + (1 + D_4 \ln \dot{\varepsilon}^*) (1 + D_5 T^*)]
\]

Where \(\sigma^*\) is the ratio of pressure divided by effective stress:
Fracture occurs when the damage parameter

\[ D = \sum \frac{\Delta \varepsilon^p}{\varepsilon^t} \]

reaches the value of 1.

Input the Young's modulus (EX), density (DENS), and Poisson's ratio (NUXY) with the MP command. Input the parameters described in the above equations with TB,EOS and locations 1 - 15 of the TBDATA command:

\[
\begin{align*}
TB,EOS,,1, & \text{EOSOPT} \\
TBDATA,1,A & \\
TBDATA,2,B & \\
TBDATA,3,n & \\
TBDATA,4,C & \\
TBDATA,5,m & \\
TBDATA,6,T_{\text{melt}} & \\
TBDATA,7,T_{\text{room}} & \\
TBDATA,8, \dot{\varepsilon}^* \text{ (effective plastic strain rate)} & \\
TBDATA,9,CP \text{ (specific heat)} & \\
TBDATA,10, \text{pressure cutoff} & \\
TBDATA,11,D_1 & \\
TBDATA,12,D_2 & \\
TBDATA,13,D_3 & \\
TBDATA,14,D_4 & \\
TBDATA,15,D_5 & \\
\end{align*}
\]

When using the Johnson-Cook Plasticity Model, you can define one of three types of equations of state: Linear Polynomial (\text{EOSOPT} = 1), Gruneisen (\text{EOSOPT} = 2), and Tabulated (\text{EOSOPT} = 3). Each has its own set of required constants that are input starting with position (LOC) 16 of the TBDATA command. See the descriptions of the three equation of state models in Equation of State Models (p. 93).

For an example input listing using the Linear Polynomial EOS, see Johnson-Cook Linear Polynomial EOS Example: 1006 Steel (p. 181).

For an example input listing using the Gruneisen EOS, see Johnson-Cook Gruneisen EOS Example: OFHC Copper (p. 181).

### 7.2.6.6. Null Material Model

This material allows equations of state to be considered without computing deviatoric stresses. Optionally, you can define a viscosity. Erosion in tension and compression is possible. The Young's modulus and Poisson's ratio are used only for setting the contact interface stiffness, so you should use reasonable values.

Input the Young's modulus (EX), density (DENS), and Poisson's ratio (NUXY) with the MP command. Input the parameters listed below with TB,EOS and locations 1 - 4 of the TBDATA command:

\[
\begin{align*}
TB, & \text{ EOS,,,2, EOSOPT} \\
\end{align*}
\]
7.2.6. Equation of State Models

**TBDATA**, 1, Pressure Cutoff ($\leq 0.0$)
**TBDATA**, 2, Viscosity Coefficient (optional)
**TBDATA**, 3, Relative volume for erosion in tension ($V/V_o$)
**TBDATA**, 4, Relative volume for erosion in compression ($V/V_o$)

When using the Null Model, you can define one of three types of equations of state: Linear Polynomial ($\text{EOSOPT} = 1$), Gruneisen ($\text{EOSOPT} = 2$), and Tabulated ($\text{EOSOPT} = 3$). Each has its own set of required constants that are input starting with position (LOC) 16 of the **TBDATA** command. See the descriptions of the three equation of state models in *Equation of State Models* (p. 93).

For an example input listing using the Linear Polynomial EOS, see *Null Material Linear Polynomial EOS Example: Brass* (p. 182).

For an example input listing using the Gruneisen EOS, see *Null Material Gruneisen EOS Example: Aluminium* (p. 182).

### 7.2.6.7. Zerilli-Armstrong Model

This model is used in metal forming processes and high speed impact applications where the stress depends on strain, strain rate, and temperature. The Zerilli-Armstrong model expresses the yield stress as follows.

For FCC Metals:

$$\sigma_y = C_1 + \left[ C_2 (\varepsilon^p)^{0.5} \left( e^{(-C_3 + C_4 \ln(\dot{\varepsilon}^*))} T \right) + C_5 \right] \left( \frac{\mu(T)}{\mu(293)} \right)$$

and for BCC Metals:

$$\sigma_y = C_1 + C_2 e^{(-C_3 + C_4 \ln(\dot{\varepsilon}^*))} T + \left[ C_5 (\varepsilon^p)^n + C_6 \right] \left( \frac{\mu(T)}{\mu(293)} \right)$$

where:

$$\left( \frac{\mu(T)}{\mu(293)} \right) = B_1 + B_2 T + B_3 T^2$$

and the relationship between heat capacity and temperature is given by:

$$C_p = G_1 + G_2 T + G_3 T^2 + G_4 T^3$$

An equation of state must also be defined with the Zerilli-Armstrong model.

Input the density (DENS), elastic modulus (EX) and Poisson's ratio (NUXY) with the **MP** command. Input the parameters described below with **TB,EOS** and locations 1-19 of the **TBDATA** command:

**TB,EOS,,,3, EOSOPT**

**TBDATA**, 1, $\dot{\varepsilon}_0$ (initial strain)
**TBDATA**, 2, $n$ (strain exponent for BCC metals)
**TBDATA**, 3, $T_{\text{room}}$ (room temperature)
**TBDATA**, 4, PC (pressure cutoff value)
When using the Zerilli-Armstrong Model, you can define one of three types of equations of state: Linear Polynomial (\(EOSOPT = 1\)), Gruneisen (\(EOSOPT = 2\)), and Tabulated (\(EOSOPT = 3\)). Each has its own set of required constants that are input starting with position (LOC) 20 of the TBDATA command. See the descriptions of the three equation of state models in Equation of State Models (p. 93).

### 7.2.6.8. Steinberg Model

This material model is used for modeling high strain rate effects in solid elements with failure. The yield strength is a function of both temperature and pressure. The model must be defined with an equation of state.

The Steinberg material model was originally developed for treating plasticity at very high strain rates \(10^5\) s\(^{-1}\) where the influence of strain rate variation on the material's yield stress begins to saturate out. In this model, the shear modulus (\(G\)) and the yield strength (\(\sigma_y\)) both increase with pressure but decrease with temperature. Both \(G\) and \(\sigma_y\) approach zero as the melting temperature of the material is reached.

In this model, the shear modulus before the material melts is defined by:

\[
G = G_0 \left[ 1 + bpV^{1/3} - h \left( \frac{E_i - E_c}{3R'} - 300 \right) \right] e^{-fE_i/E_m - E_i}
\]

where \(G_0\), \(b\), \(h\), and \(f\) are experimentally determined material constants, \(p\) is the pressure, \(V\) is the relative volume, and \(E_c\) is the cold compression energy:

\[
E_c(x) = \int_0^x pdx - \frac{900 R' \exp(ax)}{(1-x)^2(Y_o - a - 1/2)}
\]

\(x = 1 - V\)

and \(E_m\) is the melting energy:

\[
E_m(x) = E_c(x) + 3R'T_m(x)
\]
which is in terms of the melting temperature $T_m(x)$:

$$T_m(x) = \frac{T_{mo} \exp(2ax)}{\sqrt{2(y_o - a - 1/3)}}$$

and the melting temperature at $\rho = \rho_o$, $T_{mo}$.

In the above equations, $R'$ is defined by

$$R' = \frac{R\rho}{A}$$

where $R$ is the gas constant and $A$ is the atomic weight. If $R'$ is not defined, LS-DYNA computes it with $R$ in the cm-gram-microsecond system of units.

The yield strength is given by:

$$\sigma_y = \sigma'_o \left[ 1 + b'pV^{1/3} - h \left( \frac{E_i - E_c}{3R'} - 300 \right) \right] e^{-fE_i/E_m - E_i}$$

if $E_m$ exceeds $E_i$. Here, $\sigma'_o$ is given by:

$$\sigma_y = \sigma'_o \left[ 1 + \beta (\varepsilon_i + \bar{\varepsilon}p) \right]^n$$

where $\varepsilon_i$ is the initial plastic strain. Whenever $\sigma'_o$ exceeds $\sigma_{mv}$, $\sigma'_o$ is set equal to $\sigma_{mv}$. After the material melts, $\sigma_y$ and $G$ are set to one half their initial values.

If the coefficients $EC_0$-$EC_9$ are not defined in the input, LS-DYNA will fit the cold compression energy to a ten term polynomial expansion either as a function of $\mu$ or $\eta$, depending on the input variable FLAG:

$$E_c(\eta^i) = \sum_{i=0}^{9} EC_i \eta^i$$

$$E_c(\mu^i) = \sum_{i=0}^{9} EC_i \mu^i$$

where $EC_i$ is the $i$th coefficient, and:

$$\eta = \frac{\rho}{\rho_o}$$
\[ \mu = \frac{\rho}{\rho_0} - 1 \]

A linear least squares method is used to perform the fit.

A choice of three spall models is offered to represent material splitting, cracking, and failure under tensile loads. The pressure limit model, SPALL = 1, limits the hydrostatic tension to the specified value, \( p_{cut} \). If pressures more tensile than this limit are calculated, the pressure is reset to \( p_{cut} \). This option is not strictly a spall model, since the deviatoric stresses are unaffected by the pressure reaching the tensile cutoff, and the pressure cutoff value, \( p_{cut} \), remains unchanged throughout the analysis. The maximum principal stress spall model, SPALL = 2, detects spall if the maximum principal stress, \( \sigma_{max} \), exceeds the limiting value \(-p_{cut}\). Note that the negative sign is required because \( p_{cut} \) is measured positive in compression, while \( \sigma_{max} \) is positive in tension. Once spall is detected with this model, the deviatoric stresses are reset to zero, and no hydrostatic tension (\( p < 0 \)) is permitted. If tensile pressures are calculated, they are reset to 0 in the spalled material. Thus, the spalled material behaves as a rubble or incohesive material. The hydrostatic tension spall model, SPALL = 3, detects spall if the pressure becomes more tensile than the specified limit, \( p_{cut} \). Once spall is detected the deviatoric stresses are reset to zero, and nonzero values of pressure are required to be compressive (positive). If hydrostatic tension (\( p < 0 \)) is subsequently calculated, the pressure is reset to 0 for that element.

Input the shear modulus (GXY) and density (DENS) with the MP command. Input the parameters described in the above equations with TB,EOS and locations 1–28 of the TBDATA command.

\[
\begin{align*}
\text{TB,EOS,} & \quad 5,\text{EOSOPT} \\
\text{TBDATA,} & \quad 1,\sigma_o \quad \text{(initial yield stress)} \\
\text{TBDATA,} & \quad 2,\beta \quad \text{(hardening coefficient)} \\
\text{TBDATA,} & \quad 3,n \quad \text{(hardening parameter)} \\
\text{TBDATA,} & \quad 4,i \quad \text{(initial plastic strain)} \\
\text{TBDATA,} & \quad 5,\sigma_m \quad \text{(maximum yield stress)} \\
\text{TBDATA,} & \quad 6,b \quad \text{(shear modulus parameter)} \\
\text{TBDATA,} & \quad 7,b' \quad \text{(Steinberg yield strength parameter)} \\
\text{TBDATA,} & \quad 8,h \quad \text{(Steinberg yield strength parameter)} \\
\text{TBDATA,} & \quad 9,f \quad \text{(Steinberg exponential coefficient)} \\
\text{TBDATA,} & \quad 10,A \quad \text{(atomic weight)} \\
\text{TBDATA,} & \quad 11,T_{mo} \quad \text{(absolute melting temperature)} \\
\text{TBDATA,} & \quad 12,\gamma_o \quad \text{(melting temperature parameter)} \\
\text{TBDATA,} & \quad 13,a \quad \text{(melting temperature parameter)} \\
\text{TBDATA,} & \quad 14,PC \quad \text{(pressure cutoff, } p_{cut} \text{)} \\
\text{TBDATA,} & \quad 15,\text{SPALL} \quad \text{(Spall type. Set SPALL = 1 to use the pressure limit spall model; set SPALL = 2 to use the maximum principal stress spall model; set SPALL = 3 to use the hydrostatic tension spall model.)} \\
\text{TBDATA,} & \quad 16,\text{FLAG} \quad \text{(cold compression energy flag)} \\
\text{TBDATA,} & \quad 17,\text{MMN} \quad \text{(minimum bulk modulus)} \\
\text{TBDATA,} & \quad 18,\text{MMX} \quad \text{(maximum bulk modulus)} \\
\text{TBDATA,} & \quad 19-28,\text{EC}_0-\text{EC}_9 \quad \text{(cold compression energy constants)}
\end{align*}
\]

When using the Steinberg Model, you can define one of three types of equations of state: Linear Polynomial (EOSOPT = 1), Gruneisen (EOSOPT = 2), and Tabulated (EOSOPT = 3). Each has its own set of required constants that are input starting with position (LOC) 29 of the TBDATA command. See the descriptions of the three equation of state models in Equation of State Models (p. 93).

For an example input listing using the Gruneisen EOS, see Steinberg Gruneisen EOS Example: Stainless Steel (p. 183).
7.2.7. Discrete Element Models

7.2.7.1. Linear Elastic Spring Model

This model provides a translational or rotational elastic spring located between two nodes. Input the spring elastic stiffness with the `TB,DISCRETE,,,0` and location 1 of the `TBDATA` command:

```
TB,DISCRETE,,,0
TBDATA,1,KE (Elastic stiffness (force/displacement) or (moment/rotation))
```

7.2.7.2. General Nonlinear Spring Model

This model provides general nonlinear translational or rotational spring with arbitrary loading and unloading definitions located between two nodes. Hardening and softening can be defined. Input the model parameters with the `TB,DISCRETE,,,5` and locations 1 to 5 of the `TBDATA` command:

```
TB,DISCRETE,,,5
TBDATA,1,LCDL (Load curve ID describing force vs. displacement or moment vs. rotation relationship for loading)
TBDATA,2,LCDU (Load curve ID describing force vs. displacement or moment vs. rotation relationship for unloading)
TBDATA,3,BETA (Hardening parameter)
TBDATA,4,TYI (Initial yield force in tension (= 0: Tensile and compressive yield with strain softening; ≠ 0: Kinematic hardening without strain softening; = 1: Isotropic hardening without strain softening.))
TBDATA,5,CYI (Initial yield force in compression (< 0))
```

7.2.7.3. Nonlinear Elastic Spring Model

This model provides nonlinear elastic translational or rotational spring with arbitrary force/displacement or moment/rotation dependency. Input the model parameters with the `TB,DISCRETE,,,3` and locations 1 and 2 of the `TBDATA` command:

```
TB,DISCRETE,,,3
TBDATA,1,LCID (Load curve ID (force/displacement or moment/rotation))
TBDATA,2,LCR (Optional load curve ID describing scale factor on force or moment as a function of relative velocity or rotational velocity, respectively.)
```

7.2.7.4. Elastoplastic Spring Model

This model provides elastoplastic translational or rotational spring with isotropic hardening located between two nodes. Input the spring elastic stiffness, tangent stiffness and yield force with the `TB,DISCRETE,,,2` and locations 1 to 3 of the `TBDATA` command:

```
TB,DISCRETE,,,2
TBDATA,1,KP (Elastic stiffness (force/displacement) or (moment/rotation))
TBDATA,2,KT (Tangent Stiffness (force/displacement or moment/rotation))
TBDATA,3,FY (Yield (force) or (moment))
```

7.2.7.5. Inelastic Tension- or Compression-Only Spring Model

This model provides an inelastic tension or compression only, translational or rotational spring located between two nodes. Optionally, the user can define unloading stiffness instead of the maximum loading stiffness. Input the model parameters with the `TB,DISCRETE,,,7` and locations 1 to 3 of the `TBDATA` command:
7.2.7.6. Maxwell Viscosity Spring Model

This model provides a three parameter Maxwell viscoelastic translational or rotational spring located between two nodes. Optionally, a cutoff time with a remaining force/moment can be defined. Input the model parameters with the \texttt{TB,DISCRETE,,,,6} and locations 1 to 6 of the \texttt{TBDATA} command:

\begin{itemize}
  \item \texttt{TB,DISCRETE,,,,6}
  \item \texttt{TBDATA,1,Ko} (Short time stiffness)
  \item \texttt{TBDATA,2,KI} (Long time stiffness)
  \item \texttt{TBDATA,3,BETA} (Decay parameter)
  \item \texttt{TBDATA,4,TC} (Cutoff time, after which a constant force/moment is transmitted)
  \item \texttt{TBDATA,5,FC} (Force/moment after cutoff time)
  \item \texttt{TBDATA,6,COPT} (Time implementation option)
\end{itemize}

7.2.7.7. Linear Viscosity Damper Model

This model provides linear translational or rotational damper located between two nodes. Input the damping constant with the \texttt{TB,DISCRETE,,,,1} and location 1 of the \texttt{TBDATA} command:

\begin{itemize}
  \item \texttt{TB,DISCRETE,,,,1}
  \item \texttt{TBDATA,1,DC} (Damping constant (force/displacement rate) or (moment/rotation rate))
\end{itemize}

7.2.7.8. Nonlinear Viscosity Damper Model

This model provides a nonlinear damping spring with arbitrary force/velocity or moment/rotational velocity dependency located between two nodes. The load curve must define the response in the negative and positive quadrants and pass through point (0, 0). Input load curve ID with the \texttt{TB,DISCRETE,,,,4} and location 1 of the \texttt{TBDATA} command:

\begin{itemize}
  \item \texttt{TB,DISCRETE,,,,4}
  \item \texttt{TBDATA,1,LCID} (Load curve ID describing force vs. rate-of-displacement relationship or moment vs. rate-of-rotation relationship. The load curve must define the response in the negative and positive quadrants and pass through point (0, 0).)
\end{itemize}

7.2.7.9. Cable Model

Use this model to realistically model elastic cables. No force will develop in compression. The force, \(F\), generated by the cable is nonzero if and only if the cable is in tension. The force is given by:

\[ F = K \times \max(\Delta L, 0) \]

where \(\Delta L\) is the change in length

\[ \Delta L = \text{current length} - (\text{initial length} - \text{offset}) \]
and the stiffness is defined as:

\[ K = \frac{E \times \text{area}}{(\text{initial length} - \text{offset})} \]

The area and offset are defined by means of the real constants for LINK167. For a slack cable, the offset should be a negative length. For an initial tensile force, the offset should be positive. If a load curve is specified, Young's modulus is ignored and the load curve is used instead. The points on the load curve are defined as engineering stress versus engineering strain, i.e., the change in length over the initial length.

Use the MP and EDMP commands to input the required values:

\[
\begin{align*}
\text{MP}, & \text{DENS} \\
\text{MP}, & \text{EX} \\
\text{EDMP}, & \text{CABLE}, \text{MAT}, \text{Load Curve ID}
\end{align*}
\]

For an example input listing, see *Cable Material Example: Steel* (p. 183).

### 7.2.8. Other Models

#### 7.2.8.1. Rigid Model

Rigid bodies are defined for a material type using the command EDMP. For example, to define a rigid body for material 2, issue: EDMP,RIGID,2. All elements defined with the specified material number are then considered to be part of that rigid body. This material number, along with the element type and real constant type numbers of the elements, will define the PART ID of the rigid body. This PART ID can then be used to define loads and constraints to the rigid body (as described in *Chapter 4, Loading* (p. 33)). Elements within a rigid body do not have to be linked by mesh connectivity. Therefore, to represent more than one independent rigid body in a model, multiple rigid material types must be specified. Two independent rigid bodies, however, cannot share a common node.

Along with the EDMP command, you must use the MP command to specify Young’s modulus (Ex), Poisson’s ratio (NUXY), and density (DENS) for the rigid body’s material type. Realistic material property values are required so that the stiffness of the contact surfaces can be calculated by the program. For this reason, never use unrealistically high values of Young’s modulus or density for a rigid body in an explicit dynamic analysis. A rigid body cannot be stiffened because it is perfectly rigid.

Because the motion of the mass center of a rigid body is transferred to its nodes, constraints should not be applied to rigid bodies using the D command. The constraints and initial velocities of one node of a rigid body will be transferred to the center of mass of the body. However, if more than one node is constrained, it is hard to determine which constraints will be used. To properly apply constraints to a rigid body, the translational (VAL1) and rotational (VAL2) constraint parameter fields of the EDMP command are used as shown below:

\[
\begin{align*}
\text{VAL1} & \text{- Translational constraint parameter (relative to global Cartesian coordinates)} \\
0 & \text{No constraints (default)} \\
1 & \text{Constrained X displacement} \\
2 & \text{Constrained Y displacement} \\
3 & \text{Constrained Z displacement} \\
4 & \text{Constrained X and Y displacements}
\end{align*}
\]
5  Constrained Y and Z displacements
6  Constrained Z and X displacements
7  Constrained X, Y, and Z displacements

**VAL2** - Rotational constraint parameter (relative to global Cartesian coordinates)

<table>
<thead>
<tr>
<th>VAL2</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No constraints (default)</td>
</tr>
<tr>
<td>1</td>
<td>Constrained X rotation</td>
</tr>
<tr>
<td>2</td>
<td>Constrained Y rotation</td>
</tr>
<tr>
<td>3</td>
<td>Constrained Z rotation</td>
</tr>
<tr>
<td>4</td>
<td>Constrained X and Y rotations</td>
</tr>
<tr>
<td>5</td>
<td>Constrained Y and Z rotations</td>
</tr>
<tr>
<td>6</td>
<td>Constrained Z and X rotations</td>
</tr>
<tr>
<td>7</td>
<td>Constrained X, Y, and Z rotations</td>
</tr>
</tbody>
</table>

For example, the command **EDMP,RIGID,2,7,7** would constrain the rigid body elements of material 2 in all degrees of freedom.

After defining a rigid body, you can assign inertia properties, mass, and an initial velocity vector to that body using the **EDIPART** command. If you do not specifically define the inertia properties for a rigid body, the program will calculate them based on the finite element model.

For an example input listing, see **Rigid Material Example: Steel (p. 183)**.
Chapter 8: Rigid Bodies

Using rigid bodies to define stiff parts in a finite element model can greatly reduce the computation time required to perform an explicit analysis. When rigid bodies are defined, all of the degrees of freedom of the nodes in the body are coupled to the body's center of mass. Hence, a rigid body has only six degrees of freedom regardless of the number of nodes that define it. By default, the mass, center of mass, and inertia properties of each rigid body are calculated from the body's volume and density of elements. The forces and moments acting on the body are summed from the nodal forces and moments at each time step. The motion of the body is then calculated, and displacements are transferred to the nodes.

The following rigid body topics are available:

8.1. Defining Rigid Bodies
8.2. Specifying Inertia Properties
8.3. Loading
8.4. Switching Parts from Deformable to Rigid
8.5. Nodal Rigid Bodies

8.1. Defining Rigid Bodies

In ANSYS LS-DYNA, a rigid body that remains rigid for the duration of the analysis is designated as a material model. Use the EDMP command to specify this type of rigid body. For information on defining rigid bodies as a material model, see Chapter 7, Material Models (p. 67).

In addition, two rigid bodies can be merged together so that they act as a single rigid body using the EDCRB command. Unlike the definition of rigid bodies which are based on material numbers, rigid body constraints are based on PART ID and a constraint equation number. Therefore, to place a rigid body constraint between two bodies, the command EDCRB,ADD,NEQN,PARTM,PARTS should be issued, where NEQN is the constraint equation reference number, PARTM is the master rigid body PART ID number and PARTS is the slave rigid body PART ID number. Care should be taken not to issue more than one EDCRB command with the same NEQN value, as only the last NEQN referenced will be used. When using the EDCRB command, the second rigid body noted becomes absorbed into the first, so any subsequent references to the second body are meaningless.

8.2. Specifying Inertia Properties

By default, the program calculates the inertia properties of each rigid body. However, you may find it useful to assign a specific center of gravity, mass, initial velocity (in a local or global system of coordinates) and a specific inertia tensor to a rigid body instead of relying on the values calculated from the finite element model during the solution. You can use the command EDIPART to assign any of these characteristics to the rigid body. The command format is

EDIPART, PART, Option, Cvect, TM, IRCs, Ivect, Vvect, CID

where:

PART is the part ID for which the inertia is defined;
Option is the option to be performed (ADD, DELETE, or LIST);
Cvect is a vector containing coordinates of the center of the mass of the part;
**8.3. Loading**

As described in *Chapter 4, Loading (p. 33)*, displacements and velocities are applied to rigid bodies using the **EDLOAD** command. It is important to note that all rigid body displacements and velocities are applied to a PART ID rather than to a component of nodes. For example, a typical **EDLOAD** command may look like this:

```
EDLOAD,ADD,RBUX, ,2,PAR1,PAR2
```

This command defines a UX displacement load on the rigid body identified as PART number 2. The PART number is input in the `Cname` field of the **EDLOAD** command. (Note that for other load types, a component name is input in this field. However, for the rigid body load labels, a PART number must be input instead of a component name.) The PART number must correspond to a rigid body that has been defined with the **EDMP** command.

**8.4. Switching Parts from Deformable to Rigid**

In some dynamic applications, long duration, large rigid body motions arise that are prohibitively expensive to simulate if the majority of elements in the model are deformable. One such example would be an automobile rollover where the time duration of the rollover would dominate the CPU cost relative to the impact...
that occurs much later. To improve efficiency for this class of applications, ANSYS LS-DYNA offers the capability to switch a subset of materials from a deformable state to a rigid state, and then back to deformable. By switching the deformable parts to rigid during the rigid body motion stage, you can achieve significant savings in computation time.

Deformable/rigid switching is inherently related to performing a restart. You need to stop the analysis, define the part switch, then restart the analysis again. Although you would not typically use part switching in the original analysis, you must set a flag in the new analysis input to let LS-DYNA know that all materials in the model have the potential to become rigid bodies sometime during the calculation. Issuing the `EDRD` or `EDRI` command in the original analysis will set this flag.

**Note**

Parts that are defined as rigid using the command `EDMP,RIGID` are permanently rigid and cannot be changed to deformable.

To switch a deformable part to a rigid part, issue

`EDRD,D2R,PART,MRB`

where `PART` is the part number, and `MRB` indicates a master rigid part to which the switched part should be merged. (In the GUI, go to Main Menu> Solution> Rigid-Deformable> Switch.) If you do not wish to merge the part to another rigid body, simply leave `MRB` blank. Once a deformable part is switched to rigid using the above command, you can switch it back to deformable in a subsequent restart by issuing

`EDRD,R2D,PART`

In a new analysis, if no deformable materials are going to be switched to rigid but switching may occur in a subsequent restart analysis, then issue `EDRD,D2R` with no additional arguments. (In the GUI, go to Main Menu> Solution> Rigid-Deformable> Switch and select “Initialize” as the action.)

When you switch a deformable part to rigid, you can define the inertia properties for the rigid part by issuing the `EDRI` command (Main Menu> Solution> Rigid-Deformable> Inertia Property). If you do not define the inertia properties, they will be calculated by the program.

After a deformable-to-rigid switch, some constraints defined for the deformable part will become invalid. To avoid instabilities in the calculations, you should use the `EDRC` command (Main Menu> Solution> Rigid-Deformable> Controls) to change the state of these constraint definitions. Please refer to the `EDRC` command description for more details.

For rigid body switching to work properly, the choice of element formulation is critical. In the current LS-DYNA implementation, Hughes-Liu shell and beam elements cannot be used with part switching because of the strain and stress update algorithms used in these elements. The element formulations that cannot be used include KEYOPT(1) = 1, 6, 7, and 11 for `SHELL163` and KEYOPT(1) = 0 and 1 for `BEAM161`.

A special consideration for `SOLID164` elements is that, upon switching from a rigid state to deformable, the element stresses are zeroed out to eliminate spurious behavior.

**8.5. Nodal Rigid Bodies**

Unlike typical rigid bodies that are defined with the `EDMP` command, nodal rigid bodies are not associated with a part number. A nodal rigid body is defined by the command `EDCNSTR,ADD,NRB,COMP1`, where `COMP1` is a node component. Nodal rigid bodies are typically used for modeling rigid (welded) joints in...
which different flexible components (having different MAT IDs) act together as a rigid body. Because nodal rigid bodies are not associated with a part number, other options that use rigid bodies discussed earlier in this chapter (such as loads applied with the EDLOAD command ) cannot be used with a nodal rigid body. See Constraints (p. 40) and the EDCNSTR command for more information on using nodal rigid bodies.
Chapter 9: Hourglassing

Despite being robust for large deformations and saving extensive amounts of computer time, the one-point (reduced) integration solid and shell elements used in ANSYS LS-DYNA are prone to zero-energy modes. These modes, commonly referred to as hourglassing modes, are oscillatory in nature and tend to have periods that are much shorter than those of the overall structural response (i.e., they result in mathematical states that are not physically possible). They typically have no stiffness and give a zigzag appearance to a mesh (see Figure 9.1: Hourglass Deformations (p. 109)) known as hourglass deformations. The occurrence of hourglass deformations in an analysis can invalidate the results and should always be minimized.

Figure 9.1: Hourglass Deformations

Undeformed mesh and deformed mesh with hourglassing effect

Hourglassing can affect brick and quadrilateral shell and 2-D elements, but not triangular shell, triangular 2-D, or beam elements.

Good modeling practices normally prevent hourglassing from becoming significant. The general principles are to use a uniform mesh and to avoid concentrated loads on a single point. Since one excited element transfers the hourglassing mode to its neighbors, all point loads should be spread over an area of several neighboring nodes. In general, refining the overall mesh will almost always significantly reduce the effects of hourglassing.

ANSYS LS-DYNA offers a number of internal hourglass controls. The idea behind these methods is (1) to add stiffness which resists hourglass modes but not rigid body motions and linear deformation fields, or (2) to damp velocities in the direction of hourglass modes.

One method for controlling hourglassing modes is to adjust the model's bulk viscosity. Hourglass deformations are resisted by a structure's bulk viscosity, which is automatically calculated by the program. It is possible to increase the bulk viscosity of the model by adjusting the linear ($LVCO$) and quadratic ($QVCO$) coefficients of the $EDBVIS$ command. It is generally not recommended, however, to dramatically change the default values of the $EDBVIS$ command because of adverse effects they will have on the global modes of the structure.

Another more generally applicable solution to hourglassing problems is to use the fully integrated formulations of $SHELL163$ and $SOLID164$. Fully integrated elements will never experience hourglassing modes. However, these options are more costly (in CPU time) than other element formulations, and they may lead to unreal-
Istically stiff results (locking) for problems involving incompressible behavior, metal plasticity, and bending. The locking is remedied in SHELL163 by using an assumed strain field.

Hourglassing deformations can also be resisted by adding elastic stiffness to a model. Hourglassing can be a problem with small displacement situations, particularly when dynamic relaxation is used. In these cases, it is often beneficial to add elastic stiffness to the model instead of using bulk viscosity methods. This can be done by increasing the hourglassing coefficient ($HGCO$) of the EDHGLS command. However, use care when increasing these coefficients because they may over-stiffen the model’s response in large deformation problems and cause instabilities when $HGCO$ exceeds 0.15.

The hourglass control methods discussed so far have been for the entire model. The final method of hourglass control is to locally increase a portion of the model’s stiffness by using the command EDMP,HGLS. For this command, the material number, the hourglass control type (viscous or stiffness), the hourglass coefficient, and bulk viscosity coefficients must be specified. (The defaults for hourglass coefficient and bulk viscosities should be sufficient.) Using this method, hourglass control is specified for a given material and not for the entire model. This allows resisting hourglass deformations in high risk areas of the model without dramatically changing the stiffness characteristics of the entire model.

When performing an explicit dynamics analysis with reduced integration elements, it is always important to determine whether hourglassing effects have significantly degraded the results. As a general guideline, the hourglassing energy should not exceed 10% of the internal energy. The hourglass energy can be compared to the internal energy by reviewing the ASCII files GLSTAT and MATSUM and can be plotted in POST26 (see Chapter 12, Postprocessing (p. 115)). To make sure that hourglass energy results are reported in these files, the $HGEN$ field of the EDENERGY command must be set to 1.
Chapter 10: Mass Scaling

As discussed in Appendix A and shown in the figure below, the minimum time step size for explicit time integration depends on the minimum element length, \( l_{\text{min}} \), and the sonic speed, \( c \). (The example depicted represents a 2-D continuum.) Note that for a given set of material properties, the minimum time step size \( \Delta t_{\text{min}} \) is controlled by the smallest element dimension in the model (\( l_2 \) in this case). Also, for a given mesh, the minimum time step size is dependent upon sonic speed, which is a function of material properties (density, elastic modulus, and Poisson’s ratio). You can use the \textbf{EDTP} command to check the minimum time step size in your model.

\[
\Delta t_{\text{min}} = \frac{l_{\text{min}}}{c} = \frac{l_2}{c}
\]

\[
c = \sqrt{\frac{E}{(1 - \nu^2) \cdot \rho}}
\]

\( \nu = \) Poisson’s ratio
\( \rho = \) specific mass density
\( E = \) Young’s Modulus

In the ANSYS LS-DYNA program, you can control the minimum time step size by including mass scaling in the analysis. It may be necessary to use mass scaling if the program calculated time step is too small. When mass scaling is requested, element density is adjusted to achieve a user specified time step size:

\[
\left( \frac{\Delta t_{\text{specified}}}{l_i} \right)^2 = \left( 1 - \nu^2 \right) \cdot \frac{\rho_i}{E}
\]

\[
\rho_i = \frac{(\Delta t_{\text{specified}})^2 \cdot E}{l_i^2 \cdot (1 - \nu^2)} \quad \text{in element } i
\]

The \textbf{EDCTS} command is used to specify mass scaling in ANSYS LS-DYNA. Use this command to apply mass scaling to the program in one of two ways, depending on the value of the specified time step size \( DTMS \):

- If \( DTMS \) is positive, the same time step size is used for all elements, and mass scaling is applied to all elements.
• If $DTMS$ is negative, mass scaling is applied only to elements whose calculated time step size is smaller than $DTMS$.

Of the two methods listed above, the second is typically more efficient and therefore recommended. Although proper use of mass scaling will add a small amount of mass to the model and slightly change a structure's center of mass, the CPU reduction it achieves far outweighs the minor errors introduced. For example, when using mass scaling, it is common to achieve a 50% decrease in CPU time while adding only .001% mass to the model. Care must be taken not to add too much mass to a model in which inertial effects are significant.

The time step size calculated for the elements is multiplied by a scale factor (usually 0.9). The user's input for the mass scaling affects the time step size before scaling. To control the time step size after mass scaling, use the $TSSFAC$ parameter of the $EDCTS$ command.
Chapter 11: Subcycling

Subcycling, also known as mixed time integration, can be used to speed up an analysis when element sizes within a model vary significantly. Relatively small elements will result in using a small time step size for all of the elements within a model, even the larger ones. If subcycling is turned on, elements are sorted based on their time step size into groups. Each group is assigned a step size that is some even multiple of the smallest element step size (see Figure 11.1: Time Step Sizes Before and After Subcycling (p. 113)). Thus, the minimum time step size is increased for the smallest element and the groups with larger elements are processed every 2nd, 3rd, 4th, etc., step depending on their size. Subcycling is turned on in the program using the EDCSC command.

Figure 11.1: Time Step Sizes Before and After Subcycling

\[ \Delta t_{\text{min}} \] determines time step size for all elements \[ \Delta t_{\text{fmin}} = n \times \Delta t_{\text{min}} \]

Elements are sorted three times by the program:

1. By element number in ascending order.
2. By part number for large vector blocks.
3. By connectivity to ensure disjointedness for right hand side vectorization.

There are two advantages to using subcycling:

- Greatly speeds-up problems with differences in element size.
- Allows local mesh refinement without penalty.

Subcycling supports the following element classes and contact options:

- Solid elements
- Beam elements
- Shell elements
- Penalty based contact algorithms

Discrete elements are intentionally excluded since these elements generally contribute insignificantly to the calculational costs.
The interface stiffnesses used in the contact algorithms are based on the minimum value of the slave node or master segment stiffness and, consequently, the time step size determination for elements on either side of the interface is assumed to be decoupled; thus, scaling penalty values to larger values when subcycling is active can be a dangerous exercise.

Nodes that are included in constraint equations, rigid bodies, or in contact with rigid walls are always assigned the smallest time step sizes.
Chapter 12: Postprocessing

You can review ANSYS LS-DYNA results using the two ANSYS postprocessors, POST1 and POST26. Use POST1 to view results for the entire model at specific time points, or to view animated results. Use POST26 to view the results of a specific component of the model at a larger number of time points over a period of time. With an explicit dynamic analysis, you will typically want to view animated results (POST1) and time-history results (POST26).

Note

Experienced LS-DYNA users may also use the LSTC postprocessor LS-POST. However, this processor is not supported by ANSYS, Inc.

The following postprocessing topics are available:

12.1. Output Controls
12.2. Using POST1 with ANSYS LS-DYNA
12.3. Using POST26 with ANSYS LS-DYNA
12.4. Finding Additional Information

12.1. Output Controls

12.1.1. Results (Jobname.RST) vs. History (Jobname.HIS) Files

Results available for postprocessing depend on the information written to the Jobname.RST and Jobname.HIS files via the EDRST and EDHTIME commands (Main Menu> Solution> Output Controls> File Output Freq).

Note the distinction between the Jobname.RST and the Jobname.HIS files: The Jobname.RST file, used primarily in POST1 processing, contains solutions for the entire model, but captured at a relatively small number of time points. Generally, the Jobname.RST file contains just enough solutions to animate the results. The Jobname.HIS file, used in POST26, contains solutions captured at a relatively large number of time points, but for only a portion of the model. (Capturing a large number of solutions for the entire model would quickly fill up disk space.) As a comparison, the number of time steps in the Jobname.RST file is usually less than 100; in the Jobname.HIS file, it is usually 1000 or more.

12.1.2. Creating Components for POST26

Before reviewing results in POST26, you may create element and/or nodal components within your model. For example, during either the PREP7 or the SOLUTION phases of your analysis, choose a set of elements for which you want to review results. Create a component consisting only of those elements. You can also create a component consisting of a specified set of nodes. These components can be created via the GUI or by issuing the following commands:

ESEL,S,MAT,,1 !Select elements of material 1.
CM,elm1,elem !Create element component elmI.
NSLE !Select nodes from elements.
CM,nod1,node !Create nodal component nodI.
Element and nodal components should be limited to conserve disk space. For details on creating components, see the Basic Analysis Guide.

12.1.3. Writing the Output Files for POST26

Before using POST26 to review results, you must direct ANSYS LS-DYNA to write the appropriate information to the Jobname.HIS and other output files. While in either the PREP7 or SOLUTION phases of your analysis, specify the number of time steps, the element and nodal components to be analyzed, and the ASCII files to be written. You can specify this information via the GUI (Main Menu> Solution> Output Controls) or by issuing the following commands:

```
EDHTIME, NSTEPS !Specify the number of time steps for time-history results.
EDHIST,elm1   !Specify the name of the element component.
EDHIST,nod1   !Specify the name of the nodal component.
EDOUT,GLSTAT  !Write ASCII file GLSTAT (Global time step and energy statistics).
EDOUT,MATSUM  !Write ASCII file MATSUM (Energy information for each PART).
EDOUT,SPCFORC !Write ASCII file SPCFORC (Single point constraint (Reaction) forces).
EDOUT,RCFORC  !Write ASCII file RCFORC (Resultant interface force data).
EDOUT,SLEOUT  !Write ASCII file SLEOUT (Sliding interface energy).
EDOUT,NODOUT  !Write ASCII file NODOUT (Nodal data).
EDOUT,RBDOUT  !Write ASCII file RBDOUT (Rigid body data).
```

Note to LS-DYNA Users--If you are familiar with other ASCII files created by the LS-DYNA program, you may want to issue EDOUT, ALL to write all possible ASCII files.

12.2. Using POST1 with ANSYS LS-DYNA

The first step in POST1 is to read data from the results file into the database. If the database does not already contain model data, issue the RESUME command to read the database file (or in the GUI pick Utility Menu> File> Resume Jobname.db). Then issue the SET command to read in the desired set of results (or in the GUI pick Main Menu> Postproc> Read Results> By Load Step or By Time/Freq).

The typical POST1 operations available in ANSYS are also available in ANSYS LS-DYNA. You can display the deformed shape, show contour and vector displays, and obtain tabular listings just as you would in ANSYS. For details on these and other POST1 postprocessing functions, see the Basic Analysis Guide.

When using POST1, you can rotate stress results into a defined coordinate system using the RSYS command. Stress data will be rotated during printout, display, or element table operations. RSYS supports stress output for all explicit element types except BEAM161, COMBI165, and composite SHELL163 (KEYOPT(3) = 1). To use RSYS in models that contain these element types, you must unselect the unsupported elements before issuing the RSYS command.

Note

Strain data results cannot be rotated for any explicit elements. If you request strain results when RSYS is not set to the global Cartesian coordinate system, the printing or plotting command will be ignored.

When you display results in POST1, failed elements will automatically be removed from the display. Failed elements are elements that exceed the specified failure criteria (for example, failure strain). In some cases, the mesh connectivity may appear to be lost due to the number of failed elements in the model at a given
time step. However, the elements are still in the selected set, and the model will behave as expected physically. (During the LS-DYNA solution, failed elements are removed from the solution after they fail.)

### 12.2.1. Animating Results

You can also display animated shapes using POST1 capabilities; for example, you could animate the element centroidal stresses superimposed on the deformed shape. After you have read in a set of results from the Jobname.RST file, pick **Utility Menu> PlotCtrls> Animate> Over Results**. Then use the “push-button” animation controls in the dialog box to start, stop, resume, and otherwise control the animation.

If you are using command input, use the appropriate plotting command (for example, **PLESOL**) followed by the **ANDATA** command (as shown below).

```
PLESOL, Item, Comp
ANDATA
```

This command method reads in all results data and animates the plot. For faster plotting, use the **INRES** command to select specific types of solution data before issuing the above commands:

```
INRES, Item
```

### 12.2.2. Element Output Data

Remember that the results available for POST1 processing differ from element to element. For a complete description of output data for each explicit dynamics element (**LINK160**, **BEAM161**, **PLANE162**, **SHELL163**, **SOLID164**, **COMBI165**, **MASS166**, **LINK167**, **SOLID168**), see the **Element Reference**.

The stress output for explicit dynamic elements is true stress. The strain is the time integration of the incremental strain. The incremental strain is calculated from the incremental displacement, assuming a linearized relationship. So the strain measure is similar to the logarithmic strain definition. However, the reported shear strain is the engineering shear strain.

For the **PLANE162**, **SHELL163**, **SOLID164**, and **SOLID168** elements, the available strain output depends on the type of material models used. For all material models, total strains (total strain components, total principle strains, total strain intensity, and total equivalent strain) are always available. For the following material models, elastic strains are also available.

- Isotropic Elastic
- Bilinear Isotropic
- Bilinear Kinematic
- Viscoelastic
- Johnson-Cook Plasticity
- Powerlaw Plasticity
- Strain Rate Dependent Plasticity
- Piecewise Linear Plasticity
- 3 Parameter Barlat Plasticity

For the following plastic material models, equivalent plastic strain is available. However, plastic strain components and principal plastic strains are not available.

- Bilinear Isotropic
- Bilinear Kinematic
- Temperature Dependent Bilinear Isotropic
- Elastic-Plastic Hydrodynamic
- Steinberg
Below are some additional notes about the explicit dynamics elements:

- With **SOLID164**, stress and strain solutions are saved only at the element centroid, regardless of whether you use 1-point or 8-point integration (KEYOPT(1)).
- With **BEAM161** and **SHELL163** elements, stress and strain solutions are saved only at the element center (for each layer for shell elements), regardless of whether you use full or reduced integration.
- When using **BEAM161** and **SHELL163** elements, you must specify both the number of integration points for which you want ANSYS LS-DYNA to calculate the results (using real constants) and the number of these points for which you want results stored (using the **EDINT** command). By default, the number of integration points (NIP) is set to 2 for shell elements; you can specify up to 5 points (layers) for the Gauss integration rule. For more than 5 layers, you need to use either the trapezoidal integration rule or the user-defined integration rule. (Note--The trapezoidal rule is not recommended for less than twenty integration points, especially in bending.) Also by default, the number of layers you specify using **EDINT**, SHELLIP is set to 3 for shell elements. If NIP = 2, the results at the bottom correspond to integration point 1, the results at the top correspond to integration point 2, and the results at the middle are an average of the top and bottom results. For beam elements, results are saved for 4 integration points by default (BEAMIP = 4 on the **EDINT** command). For the resultant beam formulation (KEYOPT(1) = 2), there is no stress output, regardless of the BEAMIP setting. If you set BEAMIP = 0, no stress output is written for any of the beam elements. In this case, the beams will not appear in any POST1 plots because the program assumes they are failed elements.
- With **SHELL163** elements use the **LAYER**, NUM command to specify the layer for which you want results. Layers are numbered starting from the bottom layer and moving up for stress data. However, for strain data, layer 1 is the bottom layer and layer 2 is the top layer, no matter how many layers exist. Strain information is saved only at these two layers. Further, data is only available at the middle of the layer, not at the top or bottom surface of a layer (or the element, for that matter). To get results close to the element surface, specify a large number of integration points through the thickness of the shell. However, saving data for all of these layers can become expensive.
- **SHELL163** is not affected by the **SHELL**, LOC command. By default, output data for the top layer is displayed when plotting data (**PLNSOL**, **PLESOL**), but data for the top and bottom layers appears on printed results.

### 12.2.3. Postprocessing after Adaptive Meshing

If you use adaptive meshing in an explicit dynamic analysis, the analysis will typically produce multiple results files. The procedure for postprocessing these files is slightly different than postprocessing a single results file.
Adaptive meshing can be used in ANSYS LS-DYNA to refine the mesh of shell elements during a large deformation analysis (see Adaptive Meshing (p. 29) for details on how to use adaptive meshing). When adaptive meshing is included in an analysis, the number of elements in the model may increase during the solution as the mesh is improved to conform to a user specified element aspect ratio. As the number of elements in the model changes, the extension of the results file is updated to represent a new finite element mesh for the model.

The name of the results file extension is incremented as follows:

- Jobname.RS01 - First mesh refinement
- Jobname.RS02 - Second mesh refinement
- Jobname.RS03 - Third mesh refinement
- etc.

In order to postprocess multiple results files, you must specify the appropriate filename and extension (FILE command) to examine a particular time step. You should not resume the original model database (.DB file), as the mesh in that database will not match what is in the results files. The sample input below demonstrates a typical postprocessing session.

```
/POST1
FILE,Jobname,RS01  ! Specify the results file for the first mesh refinement
SET,LIST          ! List out the time steps that were written
! for the first mesh refinement
SET,1,2           ! Set the results to a particular time step value
PRNSOL,...        ! Print or display the results
```

**Caution**

The element attributes are not saved on the RSnn file. Therefore, care must be taken when using select logic during postprocessing with these files. Also, you should not attempt to perform a solution using only the information contained in these files, nor should you attempt to save a new database from these files.

You should not use the standard method for animating results (ANDATA command) after adaptive meshing because it is only valid for animating a single results file (Jobname.RST). Instead, you must use the ANMRES command to animate results when multiple results files are present (Jobname.RS01, Jobname.RS02, etc.).

Before starting the animation, you must use the /FILNAME command to specify the jobname that the results will be animated across. Because results are animated across multiple files with the same jobname, simply specifying the FILE command will not allow multiple results animations.

The following sample input demonstrates a typical animation session using ANMRES. In this example, the jobname is 'impact' and the animation runs from results file impact.rs01 to impact.rs10. The results file increment is two and the time delay is 0.5 seconds.

```
/FILNAME,impact
/POST1
FILE,impact,rs01
SET,last
PRNSOL,s,eqv
ANMRES,.5,1,10,2
```
In the GUI, the ANMRES command is accessed from the utility menu: Utility Menu> PlotCtrls> Animate> Over Results. Choose the options to animate over multiple results files and then use the push-button animation controls in the dialog box to start, stop, resume, and otherwise control the animation.

### 12.3. Using POST26 with ANSYS LS-DYNA

The typical POST26 operations available in ANSYS are also available in ANSYS LS-DYNA. You can obtain two types of results using POST26:

- Nodal and element solutions
- Miscellaneous output data, such as different energies, work, and reaction forces, stored in ASCII files (GLSTAT, RCFORC, SLEOUT, MATSUM, SPCFORC, NODOUT, and RBDOUT)

Remember when reviewing results in POST26 that you must use the Jobname.HIS file to have adequate results for a useful time-history record. The Jobname.HIS file must be loaded into POST26 using the FILE command (Main Menu> TimeHist Postpro> Settings> File):

```
FILE, Jobname,HIS
```

If you do not load the Jobname.HIS file into POST26, the results will be read from the Jobname.RST file, which may provide inadequate data for time-history postprocessing. (This is because the EDHTIME setting is usually much larger than the EDRST setting.)

#### 12.3.1. Nodal and Element Solutions

POST26 works with tables of result items versus time, known as variables. Each variable is assigned a reference number, with variable number 1 reserved for time. To generate results, define and then graph or list the variables via the GUI (Main Menu> TimeHist Postpro) or by using the following commands:

```
NSOL Specifies nodal data be stored from the results file
ESOL Specifies element data be stored from the results file
STORE Stores data into variables
PLVAR Displays results in graph format
PRVAR Lists variables vs. time
```

Element and nodal solution results can be obtained from both the Jobname.HIS and Jobname.RST. Care must be taken when reading from the .HIS file, however, such that only nodes and elements included in the component specified by the EDHIST,NCOMP are specified.

For storing and displaying solutions at different integration points for the BEAM161 and SHELL163 elements, use the LAYERP26 (Main Menu> TimeHist Postpro> Define Variables) and the STORE command (Main Menu> TimeHist Postpro> Store Data) for each layer, as shown below for SHELL163:

```
LAYERP26,num1 !Specify the first layer number
ESOL !Store element data for first layer
NSOL !Store nodal data for first layer
STORE, MERGE !Store results for first layer
LAYERP26,num2 !Specify the next layer number
ESOL !Store element data for next layer
NSOL !Store nodal data for next layer
STORE, MERGE !Store results for next layer (in addition to results for first layer)
.
.
.
LAYERP26,numn !Specify the nth layer number
```
You must specify `STORE,MERGE` for each layer, or your final plot will contain data from the last layer only.

### 12.3.2. Reading ASCII Files for Miscellaneous Output Data

To review information written to the GLSTAT, MATSUM, SPCFORC, RCFORC, SLEOUT, NODOUT, and RBDOUT files, first issue `FILE,Jobname,HIS` (Main Menu> TimeHist Postpro> Settings> File) to obtain the time information, then issue the `EDREAD` command (Main Menu> TimeHist Postpro> Read LSDYNA Data). After `EDREAD`, you must issue the `STORE` command (Main Menu> TimeHist Postpro> Store Data) to store the data in time history variables. Once stored, the data can be analyzed using standard POST26 procedures.

### 12.3.3. Data Smoothing

If you're working with noisy data (such as an earthquake excitation), you may want to “smooth” that data to a smaller set of data that provides an accurate approximation of the data points.

Four arrays are required for smoothing data. The first two contain the noisy data from the independent and the dependent variables, respectively; the second two will contain the smoothed data (after smoothing takes place) from the independent and dependent variables, respectively. You must always create the first two vectors (`*DIM`) and fill these vectors with the noisy data before smoothing the data. If you are working in interactive mode, ANSYS automatically creates the third and fourth vector, but if you are working in batch mode, you must also create these vectors (`*DIM`) before smoothing the data (ANSYS will fill these with the smoothed data).

Once these arrays have been created, you can smooth the data using the `EDNDTSD` command (Main Menu> TimeHist Postpro> Smooth Data). You can choose to smooth all or some of the data points using the `DATAP` field, and you can choose how high the fitting order for the smoothed curve is to be using the `FITPT` field. `DATAP` defaults to all points, and `FITPT` defaults to one-half of the data points. To plot the results, you can choose to plot unsmoothed, smoothed, or both sets of data.

### 12.4. Finding Additional Information

For more information on postprocessing, including POST1 and POST26, see the Basic Analysis Guide. For more information on any of the commands discussed in this chapter, see the appropriate discussion in the Command Reference.
Chapter 13: Restarting

Restarting means performing an analysis that continues from a previous analysis. The restart can begin from the end of the previous analysis or from partway through it. Reasons for doing a restart include:

- To complete an analysis that was terminated or that requires more than the user-defined CPU limit.
- To perform an analysis in stages and monitor results at the end of each stage.
- To diagnose an analysis that went wrong.
- To change the model and continue calculations.

The ability to perform an analysis in stages provides great flexibility for explicit dynamic applications. After the completion of each stage, a restart "dump" file is written. This file contains all information required to continue the analysis. Results can be checked at each stage by postprocessing the output in the normal way. You may then modify the model and continue the analysis. For example, you may decide to delete excessively distorted elements, materials that are no longer important, or contact surfaces that are no longer needed. You may also choose to change loading or consider other materials that were not present in the previous analysis. You can also change output frequencies of the various results files. By continuously monitoring and fine-tuning the analysis, you can increase efficiency and reduce the chance of wasting computer time on an incorrect analysis.

Restarting can also be used to diagnose an analysis that encountered problems. You can restart at a point before the problem (numerical difficulties or error message) occurred, and request more frequent outputs to the results files. By viewing the development of the error, you may be able to identify where the first symptoms appeared and what caused them.

The following restart topics are available:

13.1. The Restart Dump File
13.2. The EDSTART Command
13.3. Effect on Output Files

13.1. The Restart Dump File

By default, LS-DYNA writes a restart "dump" file (d3dump) at the end of each analysis. The dump file is a binary file that contains a complete LS-DYNA database for use in a restart. You can use the EDDUMP command to request that additional restart dump files be written at a specified interval during the analysis. This will give you more options for choosing a point from which to initiate a restart. The restart files are written sequentially as d3dump01, d3dump02, etc. Care should be taken not to request too many dump files because they can be quite large.

13.2. The EDSTART Command

The EDSTART command specifies the status (new or restart) of an explicit dynamic analysis (in the GUI, pick Main Menu> Solution> Analysis Options> Restart Option). There are four analysis types available: a new analysis (the default), a simple restart, a small restart, or a full restart.
13.2.1. A New Analysis

For a new analysis, you can use the EDSTART command to change the memory to be used (for example, you can increase the value if more memory is required by the LS-DYNA solver) or to change the scale factor for binary file sizes.

13.2.2. A Simple Restart

A simple restart is one for which the database (Jobname.DBB) has not been changed. You would typically run a simple restart when the ANSYS LS-DYNA solution was prematurely interrupted by a user defined CPU limit or by issuing a sense switch control SW1 (after CTRL-C). For a problem which was prematurely interrupted, enter the solution processor and issue EDSTART,1,,d3dumpnn followed by the SOLVE command. The analysis will then be continued and all results will be appended to the results files Jobname.RST and Jobname.HIS.

13.2.3. A Small Restart

Use a small restart when minor changes to the database are necessary. For this type of restart, you must issue the command EDSTART,2,,d3dumpnn followed by any commands appropriate for changing the database, then issue the SOLVE command. The types of changes you can make to the database during a small restart are listed below.

- Reset the termination time (TIME)
- Reset the output file interval (EDRST, EDHTIME)
- Specify additional ASCII files for output (EDOUT)
- Set more displacement constraints (D)
- Change initial velocities (EDVEL, EDPVEL)
- Change loading curves (EDCURVE)
- Change LS-DYNA numerical controls such as:
  - global mass damping (EDDAMP)
  - dynamic relaxation control (EDDRELAX)
  - contact small penetration control (EDSP)
  - time step control (EDCTS)
- Change the termination criteria (EDTERM)
- Delete, deactivate, or reactivate contact entities (EDDC)
- Delete elements (EDELE)
- Clear meshes (LCLEAR, ACLEAR, VCLEAR)
- Change selected set of parts (PARTSEL)
- Switch parts from rigid-to-deformable or deformable-to-rigid (EDRD, EDRC)
- Change the restart dump file frequency (EDDUMP)

Only the commands mentioned above can be used in a small restart analysis (refer to the Command Reference for details on their usage). Because some of these commands are applicable to a restart as well as a new analysis, it is important that you issue EDSTART,2 first so that subsequent commands are processed correctly for the restart.
In a small restart, you should generally extend the time of the calculation (TIME command). If the previous analysis (new or restart) finished at the specified end-time and no new time is input for the subsequent restart, the restart analysis will stop immediately with only one step. You may also need to modify termination criteria that were set in the previous analysis using the EDTERM command. If the previous analysis terminated due to one of these criteria, that specific criterion must be modified so that it will not cause the restart to terminate immediately.

In some cases, the usage of a command in the restart depends on initial settings in the original analysis. For example, in order to perform a rigid-deformable switch in the restart analysis, you must issue the EDRD command in the original analysis, even if no switch is being made. Furthermore, inertia properties for parts that are switched in the restart must be specified in the original analysis (EDRI command). Another example is mass scaling; in order to use mass scaling (EDCTS command) in the restart, mass scaling must be active in the original analysis. Limitations related to restarts are discussed in each restart related command description.

When you issue the SOLVE command to initiate the restart analysis, ANSYS LS-DYNA creates a text file named Jobname.R which is used as input to LS-DYNA. This file contains only the changes that were made to the model for the restart. If you want to run LS-DYNA directly, you can use the EDWRITE command to output this file, then specify it as the input file on the LS-DYNA execution command.

The results for the small restart analysis will be appended to all the results files. The restart solutions are numbered as load step 2, 3, etc. in Jobname.RST. In other results files, results are appended according to the time value. Restart dump files (d3dumpnn) are consecutively numbered beginning from the last number. (All modifications to the database during the restart will be reflected in subsequent restart dump files.)

---

**Note**

When postprocessing the restart results, take care not to select parts that were unselected in the restart analysis. If you do select such parts, there will be a mismatch in the database since the associated element definitions still exist, but no postprocessing data is written for the unselected parts.

After the first small restart analysis, you may choose to perform an additional restart, or even a series of restart analyses. Be sure to issue EDSTART,2 with a different d3dump file at the beginning of each restart. The general procedure for multiple restarts is outlined below.

1. Create the initial model and run a new analysis.
2. Postprocess the results.
3. Issue EDSTART,2 with an appropriate d3dump file.
4. Issue the commands needed to change the model.
5. Issue the SOLVE command
6. Postprocess the restart analysis results.
7. Repeat step 3-6, as needed.

---

**13.2.4. A Full Restart**

A full restart is appropriate when many changes to the database are needed. For example, you may need to consider more materials, to remove portions of the model, or to apply different loading conditions.
To initiate a full restart, you must issue the command **EDSTART**,3 to indicate that subsequent commands apply to the full restart. For example, assume that the previous analysis was run using the input file Jobname.K, and it produced a restart dump file named d3dump01. You would issue the command **EDSTART**,3,,d3dump01, then make any necessary changes to the model using commands available in the ANSYS LS-DYNA product. (A few of the ANSYS LS-DYNA commands are not supported in a new restart; these are discussed below.)

When the **EDSTART**,3 command is issued, the jobname is automatically changed to Jobname_01 to avoid overwriting the previous results and database. In a full restart, LS-DYNA creates entirely new results files instead of appending to the existing results (as is done in other types of restart analyses). The advantage of the full restart is that the changed database and result files match each other.

An important step in performing a full restart is specifying stress initialization via the **EDIS** command. You must carry forward some results (deformed nodal positions and stresses/strains) from the previous analysis. Typically, you would specify stress initialization for some or all of the parts that will be kept in the full restart. To do this, you must issue the command **EDIS**,ADD,PIDN,PIDO for each part that will be initialized. If part IDs have changed in the full restart due to changes in the model, you need to specify the new part ID in the **PIDN** field, and the old part ID from the previous analysis in the **PIDO** field. If you issue **EDIS** with no arguments, stress initialization is performed for all parts that were defined in the previous analysis (that is, parts having the same part ID); this option is only appropriate if part IDs have not changed and you want all parts to be initialized.

When you issue **SOLVE** to initiate the full restart solution (or when you issue **EDWRITE**), the complete database is written as an LS-DYNA input file, Jobname_01.K. When LS-DYNA starts execution, it uses the information contained in files Jobname_01.K and d3dump01 to initialize any portions of the model which were specified by **EDIS** commands. The deformed positions and velocities of the nodes on the elements of each part, and the stresses and strains in the elements (and the rigid body properties, if the material of the part is rigid) are assigned at this time.

**Note**

Portions of the model that are not initialized will have no initial strains and stresses. However, if initialized and non-initialized parts share common nodes, those nodes will be considered by the initialized part. This will cause a sudden strain in the non-initialized part.

During the initialization, it is assumed that each part being initialized has the same characteristics (that is, the same number of elements, in the same order, with the same topology) in the full restart analysis (Jobname_01.DB) as it did in the previous analysis (Jobname=DB). If this is not the case, the part cannot be initialized. (Note that the parts may have different identifying numbers, as stated above.) To avoid a mismatch in parts, follow these recommendations when creating or changing the model:

- If you anticipate the need to delete certain elements from the model in a future full restart analysis, use a different element type number, material number, or real constant number for those elements in the original analysis, even if they share the same attributes with other elements in the model. This will cause a unique part number to be assigned to those elements so that they may be deleted later without altering the makeup of any other parts in the model.

- If you need to add more elements to the model during the full restart, use a different element type number, material number, or real constant number for those elements, even if they share the same attributes with other elements in the full restart analysis. Again, this will cause a unique part number to be assigned to the new elements, and previously existing parts will not be altered.
If you do not follow the above guidelines, you may inadvertently create parts in the restart analysis that do not match parts from the previous analysis. In this case, stress initialization may fail for those parts.

For discrete elements (COMBI165), the initialization is "all or nothing." If you initialize any discrete elements in the full restart, all discrete elements will be initialized.

Although you can change almost any aspect of the analysis in a full restart, a few features are not supported or are supported in a limited way, as described below.

- **Contact specifications**: You cannot add or delete any contact specifications (EDCGEN and EDDC) in the full restart. However, you can list contact specifications (EDCLIST) that were defined in the previous analysis.

- **Initial velocities**: You cannot change initial velocities (EDVEL and EDPVEL) in the full restart. For any portions of the model that are carried over from the previous analysis, the velocities at the beginning of the full restart will be the same as the velocities at the end of the previous analysis. You cannot define initial velocities for new nodes or parts added in the full restart; the initial velocity of new model entities are assumed to be zero. You can list the initial velocities defined in the previous analysis by using the commands EDVEL, LIST and EDPVEL, LIST.

- **Adaptive meshing**: Adaptive meshing (EDADAPT and EDCADAPT) is not supported in a full restart. In addition, a full restart is not possible if adaptive meshing was used in the previous analysis.

- **Mass scaling**: As in a small restart, mass scaling (EDCTS) is only supported in a full restart if it was active in the previous analysis.

You can perform multiple full restart analyses and you can mix full restart analyses with other restart analysis types (simple or small). Just issue EDSTART, 3 with a different d3dump file at the beginning of each full restart. Each time EDSTART, 3 is issued, the jobname will be changed from the current jobname to Jobname_nn (nn = 01, 02, ... ) automatically.

### 13.3. Effect on Output Files

For both the simple restart and small restart, results are appended to the output files of the previous analysis. For a simple restart, all outputs are shown as substeps of load step 1 in the Jobname.RST file (similar to a new analysis). For a small restart, different stages of restarts are shown as different load steps in Jobname.RST. For both the simple restart and small restart, the history file (Jobname.HIS) and ASCII files (glstat, matsum, etc.) are appended with consecutive time markings.

In a full restart, new results files are created using the numbered jobname of the restart (Jobname_nn.RST and Jobname_nn.HIS). ASCII output files, however, are not renamed and will be overwritten. If you need to keep any ASCII output files from the previous analysis, you must save them to another name prior to issuing SOLVE in the full restart. Time is continuous and is not reset to zero in any of the output files (Jobname_nn.RST, Jobname_nn.HIS, glstat, matsum, etc.). The results in Jobname_nn.RST are saved as substeps of load step 1.
Chapter 14: Explicit-to-Implicit Sequential Solution

The simulation of some engineering processes requires the capabilities of both implicit (ANSYS) and explicit ANSYS LS-DYNA analyses. To solve these problems, you need to use both solution methods, e.g., an explicit solution followed by an implicit solution or vice-versa. ANSYS LS-DYNA is an explicit dynamics program intended to solve short duration dynamic problems. If an engineering process contains phases that are essentially static or quasi-static (such as a preload before a dynamic phase or a springback after a metal forming phase), then these phases are best analyzed using the ANSYS implicit code. Procedures combining the ANSYS implicit solver with the ANSYS LS-DYNA explicit solver provide an extremely powerful tool that can be used to simulate many complex physical phenomena. In this chapter, we will describe explicit-to-implicit procedures, while implicit-to-explicit processes are outlined in Chapter 15, Implicit-to-Explicit Sequential Solution (p. 135).

The following explicit-to-implicit analysis topics are available:

14.1. Performing an Explicit-to-Implicit Sequential Solution
14.2. Troubleshooting a Springback Analysis

14.1. Performing an Explicit-to-Implicit Sequential Solution

In sheet metal forming operations, springback deformation is an essential parameter that significantly complicates the design of forming tools. Springback deformation is best defined as the dimensional change of the formed part from that of the die which occurs from elastic deformations during unloading. In most dynamic metal forming operations, the highly nonlinear deformation processes tend to generate a large amount of elastic strain energy in the blank material. This elastic energy, which becomes stored in the blank while it is in dynamic contact with the die components, is subsequently released after the forming pressure is removed. This release of energy, which is the driving force for springback, generally causes the blank to deform towards its original geometry. Hence, the final part shape in a sheet metal forming process not only depends upon the contours of the dies, but also on the amount of elastic energy stored in the part while it is being plastically deformed. Since the amount of elastic energy stored in a part is a function of many process parameters such as material properties and interfacial loads, predicting springback during forming is extremely complicated. This poses significant problems to designers and analysts who must accurately assess the amount of springback that will occur during a forming process so that a desired part shape can be obtained. Herein lies the great advantage of performing an explicit-to-implicit sequential solution. By simulating the dynamic forming process explicitly, and then modeling the springback deformations implicitly, stringent design tolerances can be attained for sheet metal forming operations.

In an explicit-to-implicit sequential solution, you must first run the ANSYS LS-DYNA program to simulate the metal forming process. In the explicit metal forming analysis, the sheet metal blank being deformed must consist of SHELL163 and/or SOLID164 elements. Then the deformed shape, stresses, and thicknesses of these elements only are transferred into the corresponding implicit ANSYS elements (SHELL181 and SOLID185). Once valid boundary conditions are specified on the blank material, implicit simulation of the elastic springback within the workpiece can be performed.
Note

Only elastic properties are considered in the implicit phase. This implies that the analysis is purely elastic unloading from the formed state (configuration). The displacements are interpreted as the changes in these quantities from the formed configuration.

A detailed description of the explicit-to-implicit solution procedure follows.

1. Run the explicit analysis as described earlier, using Jobname1. You must use SHELL163 and/or SOLID164 to model the working piece in order to analyze the springback effect in a subsequent ANSYS implicit analysis. In addition, you must use one of the following element formulations for the SHELL163 elements: KEYOPT(1) = 1, 2, 6, 7, 8, 9, 10, 11, or 12 (10 and 12 are recommended). Solve and finish the analysis.

You should always check your explicit analysis solution from ANSYS LS-DYNA carefully before proceeding with the ANSYS implicit analysis. Specifically, check whether there is any undesirable dynamic effect left in the structure at the end of the explicit run (using POST26).

2. Save the explicit analysis database to file Jobname1.DB.

**Command(s):** SAVE
**GUI:** Utility Menu> File> Save as

---

Note

If you do not save your Jobname1.DB file at this point, then the database for this explicit run will not be saved. Only the database file for the subsequent implicit run will be saved.

3. Change to Jobname2 to prevent the explicit results files from being overwritten.

**Command(s):** /FILNAME,Jobname2
**GUI:** Utility Menu> File> Change Jobname

4. Reenter the preprocessor.

**Command(s):** /PREP7
**GUI:** Main Menu> Preprocessor

5. Convert explicit element types to corresponding companion implicit element types. (Note that the 2-D explicit element, PLANE162, and composite shell element SHELL163 with KEYOPT(3) = 1, cannot be used in a sequential solution.) The companion explicit-implicit element type pairs are:

<table>
<thead>
<tr>
<th>Explicit Element Type</th>
<th>Implicit Element Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINK160</td>
<td>LINK8</td>
</tr>
<tr>
<td>BEAM161</td>
<td>BEAM161</td>
</tr>
<tr>
<td>SHELL163</td>
<td>SHELL181</td>
</tr>
<tr>
<td>SOLID164</td>
<td>SOLID185</td>
</tr>
<tr>
<td>COMBI165</td>
<td>COMBI165</td>
</tr>
<tr>
<td>MASS166</td>
<td>MASS21</td>
</tr>
<tr>
<td>LINK167</td>
<td>LINK10</td>
</tr>
</tbody>
</table>
Although all explicit element types are converted, only SHELL163 and SOLID164 data (stresses and thicknesses for SHELL163, and stresses for SOLID164) are transferred to SHELL181 and SOLID185 (via the RIMPORT command; see step 12).

**Command(s):** ETCHG, ETI  
**GUI:** Main Menu > Preprocessor > Elem Type > Switch Elem Type

6. Redefine the key options, real constants, material properties, boundary conditions, and loading values on any implicit elements that are converted from explicit element types. (For SHELL163 elements that were converted to SHELL181, you do NOT need to redefine the real constants, but you do need to redefine the other values. For SOLID164 elements that were converted to SOLID185 elements, you need to specify the uniform reduced integration option by setting KEYOPT(2) = 1.) The TYPE, REAL, and MAT numbers from the explicit elements are retained, but the actual key option and real constant values are reset to zero or the default settings.

**Command(s):** KEYOPT, R, MP, etc.  
**GUI:** Main Menu > Preprocessor > Element Type/Real Constants/Material Properties/Loads

**Note**

Only linear elastic material properties (as specified with the MP command) can remain active in the ANSYS implicit phase. Delete any inelastic material properties (as specified with the TB command) from the ANSYS LS-DYNA run.

7. Turn off shape checking because elements may have undergone considerable deformation during the explicit analysis.

**Command(s):** SHPP, OFF  
**GUI:** Main Menu > Preprocessor > CheckingCtrls > Shape Checking

8. Redefine the implicit elements to the deformed configuration.

**Command(s):** UPGEOM  
**GUI:** Main Menu > Preprocessor > Modeling > Update Geometry

9. Unselect or delete any unnecessary elements (mainly those making up any rigid bodies from the explicit analysis), or convert them to null elements. Any explicit elements that are not either unselected, deleted, converted to null elements, or converted to implicit will remain active in ANSYS, which will produce an error and terminate the analysis. Also, if the rigid bodies in the explicit analysis were made up of SHELL163 or SOLID164 elements, these elements must be unselected, deleted, or converted to NULL elements before importing stresses and thicknesses (from SHELL163 to SHELL181, or SOLID164 to SOLID185) by the RIMPORT command (See Step 12); otherwise, the implicit analysis will be terminated.

**Command(s):** ESEL, EDELE  
**GUI:** Utility Menu > Select > Entities or Main Menu > Preprocessor > Delete > Elements

10. Reenter the solution processor.

**Command(s):** /SOLU  
**GUI:** Main Menu > Solution

11. Set any necessary constraints on the model by modifying or adding to the boundary conditions defined during the explicit analysis (for example, in a metal forming analysis, you need to constrain the blank).

**Command(s):** D, etc.
12. Import stresses and changed thicknesses (from SHELL163 to SHELL181, and from SOLID164 to SOLID185). For SOLID164, only the stresses are transferred. For SHELL163, the stresses and thicknesses are transferred. The deformed integration point thicknesses are averaged before being transferred to the implicit analysis. These thicknesses override those defined using real constants.

**Command(s):** RIMPORT

**GUI:** Main Menu> Solution> Loads> Apply> Structural-Displacement> On Nodes, etc.

13. Turn large deformation effects on.

**Command(s):** NLGEOM, ON

**GUI:** Main Menu> Solution> Sol’n Controls

14. Solve and finish the analysis.

**Command(s):** SOLVE, FINISH

**GUI:** Main Menu> Solution> Solve> Current LS
Main Menu> Finish

Once you have solved the analysis, you can use any of the standard ANSYS postprocessing functions to review your results.

The following is a sample input stream for performing an explicit-to-implicit sequential solution.

```
/batch,list
resume,stamp1,db        ! resume explicit database (explicit
                        ! problem run previously)
/filename,stamp2        ! change jobname so explicit results
                        ! are not overwritten
/prep7
etchg,eti              ! convert explicit to implicit elements
r,1,.01,.01,.01         ! change real constants (deformed
                        ! thicknesses used by RIMPORT)
ddel,all                ! delete constraint loads from
                        ! explicit analysis
tbdelete,all,all        ! delete material models (retain
                        ! linear material properties)
upgeom,,,,,stamp1,rst   ! redefine the implicit elements
                        ! at the deformed geometry
d,n1,ux,0.0,,,,uy,uz    ! Set necessary constraints on the model
                        ! (at least 3 non-colinear nodes)
d,n2,ux,0.0,,,,uy
                        ! esel,s,ename,,181       ! Select only corresponding SHELL181
                        ! elements (No “former” rigid bodies)
sdle,s,1                ! Select only SHELL181 nodes
shpp,off                ! Turn off shape checking
finish
/solution
nlgeom,on               ! Turn on nonlinear geometry
rimport,dyna,,,,,stamp1,rst ! Import stresses and thicknesses from
                           ! explicit analysis
save
solve
finish
```

### 14.2. Troubleshooting a Springback Analysis

Springback analyses are sometimes numerically difficult to solve. Some typical causes of convergence difficulties and actions you can take to resolve them are listed below.

- Dynamic effects due to a high punch speed used in the explicit (LS-DYNA) run.
Use a lower punch speed in the explicit run. You can do this by redefining the tool speed in the explicit analysis (EDLOAD command).

- Slenderness of the structure; the ratio of part size to its thickness is large.

  A more refined mesh may reduce this slenderness effect.

- The model contains local instabilities: the part, being very thin, tends to wrinkle along edges or sharp curvatures.

  Use a refined mesh to accurately model curvatures of the formed part.

- The model is improperly constrained.

  Verify the boundary conditions in the springback analysis. This requires some engineering judgment on how and where to apply the boundary conditions in the springback phase. When the tools are removed, the part should be free to move except for restraints against rigid body movement. It is important to simulate this carefully in the finite element model. A commonly recommended approach is to use displacement constraints at a minimum of three non-collinear nodes. One node should be restricted from motion in all directions, another node should be restricted from motion in two directions, and the last node should be restricted from motion in only one direction. You must select these three nodes carefully so that the expected springback behavior is not prevented. The magnitude and direction of the displacements in the springback analysis should be viewed as relative quantities to the plane formed by these three nodes.

If you have checked all of the above conditions and the analysis is still not converging, try the following methods (in the order listed):

- Attempt incremental springback by using a higher number of substeps.
- Use a linear springback analysis (if such an approximation is acceptable) with the command NLGEOM,OFF.
- Invoke springback stabilization (which is described below).

### 14.2.1. Springback Stabilization

If convergence difficulties persist, you should activate springback stabilization using the SPSCALE and MSCALE arguments of the RIMPORT command. The stabilization method is analogous to attaching artificial springs to all unconstrained degrees of freedom. The stiffnesses of these springs are directly proportional to the stiffnesses of the elements.

In each substep, ANSYS will assign the artificial springs large default stiffnesses to simulate a total restraint of the motion of the model at the beginning of each load step. The artificial stiffness is scaled down exponentially with the progress of iterations. (The picture below illustrates rapid decline of artificial spring stiffness over an iterative cycle.) The substep is considered to be converged only when the equilibrium convergence tolerances are met (as in a normal nonlinear analysis) and the artificial stiffness has reached an insignificantly small value. This may slow down the conversion of the solution in some cases; however, the benefit is more robust behavior of the model.

The SPSCALE parameter on the RIMPORT command is a scale factor that you input to scale (up or down) the default initial stiffness of the springs. (You must input SPSCALE to activate springback stabilization.) You can also specify an acceptable small stiffness value using the MSCALE parameter on the RIMPORT command.

The springback stabilization method is valid only when the geometric nonlinear option is active (NLGEOM,ON).
Chapter 14: Explicit-to-Implicit Sequential Solution

![Graph showing stiffness scaling factor vs iteration number]

- **Stiffness Scaling Factor**
- **Iteration Number**

The graph illustrates the decrease in stiffness scaling factor as the iteration number increases, indicating convergence in the sequential solution process.
Chapter 15: Implicit-to-Explicit Sequential Solution

In Chapter 14, Explicit-to-Implicit Sequential Solution (p. 129), we explained how to perform explicit-to-implicit sequential solutions for analyzing springback in metal forming processes. This chapter discusses another form of sequential solution, implicit-to-explicit. There are two types of implicit-to-explicit sequential solutions. The first type, which involves performing an ANSYS implicit structural analysis followed by an ANSYS LS-DYNA explicit dynamic analysis, is used primarily for applying a preload prior to the transient phase of the analysis. (If this preload contains thermally induced strains, the implicit structural analysis needs to contain temperatures from an implicit heat transfer analysis.) The second type is a thermal-to-explicit sequential solution in which the implicit phase is an ANSYS thermal analysis. This type of analysis is useful for modeling temperature-dependent phenomena such as forging.

The following implicit-to-explicit analysis topics are available:

15.1. Structural Implicit-to-Explicit Solution for Preload
15.2. Thermal Implicit-to-Explicit Solution

15.1. Structural Implicit-to-Explicit Solution for Preload

Unlike explicit-to-implicit solutions, which are used only in forming applications, structural implicit-to-explicit sequential solutions can be used for a broad range of engineering problems where a structure’s initial stress state affects its dynamic response. The following is a short list of applications in which this type of implicit-to-explicit solution may be useful.

- Droptest simulation of prestressed consumer goods
- Bird-strike of a rotating engine blade (and blade containment)
- Mounting and bearing loads on a turbine
- Pressure vessels with initial internal pressure
- Golf club striking a preloaded (wound) golf ball
- Dynamic response of bolted joints
- Dynamic response of a turbine engine under thermal straining

In an implicit-to-explicit sequential solution, you must first run an ANSYS implicit structural analysis to apply a preload to the structure being analyzed. In this implicit analysis, completely constrain all of the nodes of any elements that will only be used in the explicit analysis (e.g., the bird in a bird-strike problem). The nodal displacements and rotations from the ANSYS implicit solution are written to the ANSYS LS-DYNA dynamic relaxation file drelax.

**Note**

Temperatures from the ANSYS implicit structural solution are also written to the drelax file, but are not used by LS-DYNA. See Special Considerations for Thermal Loading (p. 139) for more information on how temperature loads are handled.
After defining additional loads, initial velocities, different material models (e.g., adding plasticity), etc., the explicit dynamic analysis can be conducted. The first part of this analysis uses the displacement results stored in the drelax file to do a stress initialization to a prescribed geometry. This preload is applied in pseudo time over 101 time steps to damp out any kinetic energy. The transient portion of the analysis then begins at time zero with a stable preloaded structure.

A detailed description of the implicit-to-explicit solution procedure follows.

1. Run the implicit analysis as described earlier, using Jobname1. Keep in mind that this analysis must be small strain with linear material behavior. The only element types that can be used for an implicit-to-explicit sequential solution are:

   | LINK8  |
   | BEAM4  |
   | SHELL181 |
   | SOLID185 |
   | COMBIN14 |
   | MASS21  |
   | LINK10  |

2. Define any additional nodes and elements that are necessary to complete the explicit solution (for example, the bird in a bird-strike simulation, or a rigid surface that a phone would impact in a drop test). These additional nodes and elements may not be part of the implicit analysis, but they need to be defined here nonetheless. These additional nodes must be constrained (using D,ALL,ALL,0).

   Command(s): N, E
   GUI: Main Menu> Preprocessor> Modeling> Create> Nodes or Elements

3. Solve and finish the analysis.

   Command(s): SOLVE, FINISH
   GUI: Main Menu> Solution> Solve
   Main Menu> Finish

4. Save the implicit analysis database to file Jobname1.DB.

   Command(s): SAVE
   GUI: Utility Menu> File> Save as

   Note

   If you do not save your Jobname1.DB file at this point, then the database for this implicit run will not be saved. Only the database file for the subsequent explicit run will be saved.

5. Change to Jobname2 to prevent the implicit results files from being overwritten.

   Command(s): /FILNAME,Jobname2
   GUI: Utility Menu> File> Change Jobname

6. Reenter the preprocessor.

   Command(s): /PREP7
   GUI: Main Menu> Preprocessor

7. Convert implicit element types to corresponding companion explicit element types. Note that the 2-D explicit element, PLANE162, cannot be used in this type of sequential solution. PLANE162 is allowed
in a thermal implicit-to-explicit sequential solution; see *Thermal Implicit-to-Explicit Solution* (p. 139) for details.) The corresponding companion implicit-explicit element type pairs are:

<table>
<thead>
<tr>
<th>Implicit Element Type</th>
<th>Explicit Element Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINK8</td>
<td>LINK160</td>
</tr>
<tr>
<td>BEAM4</td>
<td>BEAM161</td>
</tr>
<tr>
<td>SHELL181</td>
<td>SHELL163</td>
</tr>
<tr>
<td>SOLID185</td>
<td>SOLID164</td>
</tr>
<tr>
<td>COMBIN14</td>
<td>COMBIN165</td>
</tr>
<tr>
<td>MASS21</td>
<td>MASS166</td>
</tr>
<tr>
<td>LINK10</td>
<td>LINK167</td>
</tr>
</tbody>
</table>

**Command(s):** `ETCHG,ITE`

**GUI:** Main Menu > Preprocessor > Element Type > Switch Elem Type

Implicit elements not listed above can also be used, as long as they are defined by the same number of nodes, but they will not automatically be converted to explicit elements when `ETCHG` is issued. These elements must be converted manually using `EMODIF`. Higher-order implicit elements can also be used, but must also be converted manually using `EMODIF with the corner nodes only`. Do NOT delete or unselect the midside nodes - these nodes must be written to the LS-DYNA input file. The drelax file contains solutions for these nodes, but the ANSYS LS-DYNA explicit elements do not use these nodes in their definition.

**Command(s):** `EMODIF`

**GUI:** Main Menu > Preprocessor > Modeling > Move/Modify > Nodes

---

**Note**

Element types `LINK8` and `LINK10` lack a third node; however, their corresponding companion explicit element types, `LINK160` and `LINK167`, require a third (orientation) node. If you are using element types `LINK8` or `LINK10`, you must first convert the element type using `ETCHG,ITE`, and then manually define the third node of `LINK160` or `LINK167` elements using `N` and `EMODIF`.

Also, if you are converting `BEAM4` to `BEAM161`, you may need to manually define the third node of `BEAM161` elements as well. However, `BEAM4` allows you to define a third, optional node. If you have defined this third node on `BEAM4`, then the conversion to `BEAM161` will be completed automatically when you issue `ETCHG,ITE`. If you did not define the third node on `BEAM4`, then you must manually define it on `BEAM161` using `N` and `EMODIF`.

8. Redefine the key options, real constants, boundary conditions, and loading values on the explicit elements. The `TYPE`, `REAL`, and `MAT numbers` from the implicit elements are retained, but the actual key option and real constant `values` are reset to zero or the default settings.

**Command(s):** `KEYOPT, R, MP`, etc.

**GUI:** Main Menu > Preprocessor > Element Type, Real Constants, Material Props, or LS-DYNA Options

9. Remove constraints from the additional nodes or elements defined in Step 2, above.

**Command(s):** `DDELE`
GUI: Main Menu > Preprocessor > LS-DYNA Options > Constraints > Delete

10. Reenter the solution processor.

**Command(s):** /SOLU  
**GUI: Main Menu > Solution**

11. Read nodal displacements, rotations, and temperatures from the implicit results file, and write this information to an ASCII LS-DYNA file, drelax.

**Command(s):** REXPORT  
**GUI: Main Menu > Solution > Constraints > Read Disp**

12. Initialize the structure to the prescribed geometry according to the displacements and rotations contained in the drelax file. In this step, LS-DYNA applies the load information (displacements and rotations) from the drelax file to the original geometry and calculates the deformed geometry, which it then uses as a starting point for the explicit analysis.

**Command(s):** EDDRELAX  
**GUI: Main Menu > Solution > Analysis Options > Dynamic Relax**

13. Apply any necessary loading for the explicit run.

**Command(s):** EDVEL, EDLOAD, EDCURVE, etc.  
**GUI: Main Menu > Solution > Initial Velocity**  
**Main Menu > Solution > Loading Options > Specify Loads**  
**Main Menu > Solution > Loading Options > Curve Options**

14. Solve and finish the explicit dynamics analysis. You can then return to the implicit solution, if necessary.

The following is a sample input stream for performing an implicit-to-explicit sequential solution.

```
/batch,list
resume,drop1,db  ! Resume implicit database (implicit
                ! problem run previously)
/filename,drop2  ! Change jobname so implicit results are
                ! not overwritten
/prep7
etchg,ite       ! Convert implicit to explicit elements
mp,dens,1,.0216  ! Change material properties
ddel,all        ! Delete constraint loads from implicit
                ! analysis
tb,plaw,1,,,8   ! Define explicit dynamics nonlinear
                ! material models
edmp,rigid,2,7,7! Change an existing material to a rigid body
edcgen,assc    ! Specify contact algorithms (if any)
sel,s,loc,x,0   ! Select geometry for new constraints to be
                ! specified
sel,a,loc,y,0
d,all,ux,0      ! Set necessary constraints on the rigid
                ! body, etc.
d,all,uy,0
finish
/solution
rexport,dyna,,,,,drop1,rst  ! Create DRELAX file from implicit
                            ! results
eddrelax,ansys         ! Specify stress initialization by prescribed
                        ! geometry
edpart,create          ! Create parts for loading
edpart,list            ! List parts
edvel,...             ! Apply initial velocities
edload,add,rbvx,2,time,load,0  ! Apply phase = 0 loads
save
solve
finish
```
15.1.1. Special Considerations for Thermal Loading

Most types of temperature loads that are applied in the implicit structural analysis will be carried over to the subsequent explicit analysis; these include temperature loads applied with the BF, BFK, BFL, BFA, BFV, LDREAD, TUNIF, BFUNIF, and TREF commands. If you do not want to include the temperature loads in the explicit analysis, you can delete them in the ANSYS structural analysis phase (before you convert the model to explicit elements via ETCHG), or you can delete them in the explicit analysis.

In the explicit analysis phase, you can use the BFLIST command to list the temperature loads mentioned above. You can then use the BFDELE command to delete any temperatures originally defined with the LDREAD command or one of the “BF” type commands listed above. If you do not want to include a uniform temperature carried over by TUNIF (or BFUNIF,TEMP) you should set TUNIF to the reference temperature (which is the temperature specified on the TREF command). Any temperature loads that are not deleted will be written to the LS-DYNA input file, Jobname.K, automatically when the SOLVE or EDWRITE command is issued.

In order for temperature loads to take effect in the explicit analysis, you must use the Temperature Dependent Bilinear Isotropic material model or the Elastic Viscoplastic Thermal material model for portions of the model that are subjected to temperature loading. Note that only the SHELL163 and SOLID164 elements can accept temperature loading in a structural implicit-to-explicit sequential solution.

Note

The temperature loads in the implicit analysis are also written to the drelax file when REXPRT is issued (Step 12 above). However, the temperatures contained in the drelax file are not used in the LS-DYNA calculation.

Appendix D (p. 189) contains a complete example of a thermal/structural preload followed by a structural transient dynamic analysis.

Currently, ANSYS LS-DYNA does not support applying a thermal transient in a model that includes a thermal preload.

15.2. Thermal Implicit-to-Explicit Solution

The following is a detailed description of how to perform a thermal implicit-to-explicit sequential solution. Since the temperature loads are step-applied at the beginning of the transient, the analysis should be run for a sufficient amount of time to reduce the effects of the thermal shock loading. If only temperature-dependent material behavior is desired without the effects of thermally induced strain, set the coefficient of thermal expansion equal to zero (ALPX = 0) so that the analysis will be stable from the beginning of the transient. This is a convenient way to vary the modulus of elasticity in a model with only one material definition.

1. Run the implicit thermal analysis using Jobname1. The only element types that can be used for a thermal implicit-to-explicit sequential solution are:

   PLANE55
   SOLID70
   SHELL57

   Keep in mind that you cannot combine 2-D and 3-D elements in the subsequent explicit analysis. Therefore, you should use only 2-D or only 3-D elements in the thermal analysis phase.
2. You may define any additional nodes and elements that are necessary to complete the explicit solution. These additional nodes and elements may not be part of the thermal analysis, but they can be defined here if desired. These additional nodes must be constrained to a constant temperature (using $D_{NODE}, TEMP, VALUE$).

   **Command(s):** N, E  
   **GUI:** Main Menu> Preprocessor> Modeling> Create> Nodes or Elements

3. Solve and finish the analysis.

   **Command(s):** SOLVE, FINISH  
   **GUI:** Main Menu> Solution> Solve  
   Main Menu> Finish

4. Save the thermal analysis database to file Jobname1.DB.

   **Command(s):** SAVE  
   **GUI:** Utility Menu> File> Save as

   **Note**

   If you do not save your Jobname1.DB file at this point, then the database for this thermal run will not be saved. Only the database file for the subsequent explicit run will be saved.

5. Change to **Jobname2** to prevent the thermal results files from being overwritten.

   **Command(s):** /FILNAME, Jobname2  
   **GUI:** Utility Menu> File> Change Jobname

6. Reenter the preprocessor.

   **Command(s):** /PREP7  
   **GUI:** Main Menu> Preprocessor

7. Convert thermal element types to corresponding companion explicit element types. The corresponding companion thermal-explicit element type pairs are:

<table>
<thead>
<tr>
<th>Thermal (Implicit) Element Type</th>
<th>Explicit Element Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLANE55</td>
<td>PLANE162</td>
</tr>
<tr>
<td>SHELL57</td>
<td>SHELL163</td>
</tr>
<tr>
<td>SOLID70</td>
<td>SOLID164</td>
</tr>
</tbody>
</table>

   **Command(s):** ETCHG, TTE  
   **GUI:** Main Menu> Preprocessor> Element Type> Switch Elem Type

8. Redefine the key options, real constants, and material property values on the explicit elements. The TYPE, REAL, and MAT numbers from the implicit elements are retained, but the actual key option and real constant values are reset to zero or the default settings. Keep in mind that for temperature loading to take effect, you must use the Temperature Dependent Bilinear Isotropic material model in the explicit analysis.

   **Command(s):** KEYOPT, R, MP, etc.  
   **GUI:** Main Menu> Preprocessor> Element Type> Real Constants> Material Properties
9. Define additional nodes and elements, if needed. You may have already completed this task in the thermal analysis portion of the sequential solution (see Step 2). However, if desired, you could wait until the explicit analysis to complete this step.

   **Command(s): N, E**  
   GUI: Main Menu> Preprocessor> Modeling> Create> Nodes or Elements

10. Reenter the solution processor.

   **Command(s): /SOLU**  
   GUI: Main Menu> Solution

11. Read temperature results from the thermal analysis results file (Jobname1.RTH) and apply them as loads in the explicit analysis. You can only transfer the temperatures from one time point in the thermal analysis; specify this point by inputting the time value or the corresponding load step and substep numbers on the **LDREAD** command. If you issue **LDREAD** more than once, the temperature results specified by the last command will be used.

   **Command(s): LDREAD,TEMP**  
   GUI: Main Menu> Solution> Loading Options> Temp From ANSYS

   The temperatures read in are applied as body loads to the nodes in the explicit model. You can use the **BFLIST** command to list these temperature loads and the **BFDELE** command to delete them.

   **Command(s): BFLIST, BFDELE**  
   GUI: Main Menu> Solution> Loading Options> List Temps/Delete Temps

12. Apply a uniform temperature load, if needed. You can use the **TUNIF** command (or **BFUNIF,TEMP**) to apply a uniform temperature load to all nodes in the model. The temperatures read in by **LDREAD** will overwrite the **TUNIF** temperature for all nodes that were present in the thermal analysis. Therefore, **TUNIF** would typically be used only when additional nodes and elements have been defined in the explicit portion of the sequential solution.

   **Note**

   If you used **TUNIF** (or **BFUNIF,TEMP**) to define a uniform temperature in the thermal analysis, the same uniform temperature will be applied automatically to any new nodes added in the explicit analysis phase. If you do not want to apply a uniform temperature in the explicit analysis, you should set **TUNIF** to zero (or to the **TREF** temperature, if it is nonzero).

   **Command(s): TUNIF**  
   GUI: Main Menu> Solution> Loading Options> Uniform Temp

13. Define a reference temperature. The thermal loading is defined as the difference between the applied temperature and the reference temperature. If the reference temperature is not specifically defined, it defaults to zero.

   **Command(s): TREF**  
   GUI: Main Menu> Solution> Loading Options> Reference Temp

14. Apply any additional loading needed for the explicit run.

   **Command(s): EDVEL, EDLOAD, EDCURVE, etc.**  
   GUI: Main Menu> Solution> Loading Options> Initial Velocity  
   Main Menu> Solution> Loading Options> Specify Loads  
   Main Menu> Solution> Loading Options> Curve Options
15. Solve and finish the explicit dynamic analysis.

The following is a sample input stream for performing a thermal implicit-to-explicit sequential solution.

```
/clear
/title, Thermal Implicit-to-Explicit Sequential Solution
/view,,1,2,3
/plopts,info,1
/filnam,thermal ! define unique name for thermal analysis
/prep7
/et,1,SOLID70 ! thermal brick element
/r,1
/mp,kxx,1,16.3 ! thermal conductivity, W/m-C
/mp,c,1,502.0 ! specific heat, J/kg-C
/block,0,1,0,1,0,11 ! create and mesh model
/esize,0.5
/vmesh,all
/finish
/solu
/antype,static ! steady-state heat transfer analysis
/time,1.0
/nsel,s,loc,z,0
/d,all,temp,300.0 ! 300 Celsius constraint at far end of beam
/nsel,s,loc,z,11
/cm,ntip,node
/cm,etip,elem
/bf,all,hgen,20.0e3 ! 20,000 W/m^3 heat generation at near end ...
/esel,all
/nsel,s,ext ! exterior beam nodes
/sf,all,conv,6.0,0.0 ! hf=6.0 W/m^2-C and Tbulk=0 degrees C
/nsel,all
/save
/solve
/fini
/post1
/set, last
/plnsol,temp
/wait,3
/fini
/filnam,explicit ! define unique name for explicit analysis
/prep7
/bfdele,all,all ! delete thermal analysis heat generation loads
/sfdele,all,all ! delete thermal analysis convection loads
ddele,all,all ! delete thermal analysis temp. constraints
etchg,tte ! convert thermal element types to explicit
/mp,dens,1,8030.0 ! kg/m^3 - density cannot be temp. dependent
/edmp,hgls,1,5 ! add hourglass control

! Define temperature-dependent BISO model
! Specify material temperatures, degrees Celsius
/mptemp,1, 0, 500, 700, 900, 1100, 1200

! Modulus of elasticity, N/m^2 (Pascal)
/mpdata,ex,1,1, 1.93e11, 1.5e11, 1.0e11, 0.9e11, 0.8e11, 0.7e11
```
! Poisson's ratio, unitless
mpdata,nuxy,1,1, 0.3, 0.31, 0.32, 0.33, 0.34, 0.35

! Coefficient of thermal expansion, m/m-C
mpdata,alpx,1,1, 1.78e-5, 2.5e-5, 5e-5, 5.5e-5, 6.0e-5, 6.5e-5

tb,biso,1,6
tbtemp,0
tbdta,1,30000,3e6
tbtemp,500
tbdta,1,20000,2e6
tbtemp,700
tbdta,1,10000,1e6
tbtemp,900
tbdta,1,9000,0.9e6
tbtemp,1100
tbdta,1,8000,0.8e6
tbtemp,1200
tbdta,1,7000,0.7e6
edpart,create
eddamp,all,,0.01
fini

/solu
ldread,temp,last,,,thermal,rth   ! apply temperature body loads
bflist
nsel,s,loc,z,0
d,all,ux,0.0,,,uy,uz  ! structural displacement constraints
esel,all
nsel,all
time,0.15              ! use sufficient time (see below)

! Note: Since the temperatures are step-applied at the beginning of
! the transient, a thermal shock condition is created. You
! should run the analysis for a sufficient amount of time
! before adding any structural loads in order for the structure
! to stabilize (i.e., stop vibrating). Animation plots of both
! stress and strain are helpful in verifying this condition.
!
! Alternately, if only temperature-dependent material properties
! (e.g., modulus of elasticity) are desired, you can avoid the
! thermal shock condition by setting the coefficient of thermal
! expansion equal to zero (ALPX=0). Large plastic strains can
! develop due to a thermal shock scenario, so care must be
! used when running a thermal-to-explicit sequential solution.

edrst,100
edhtime,500
edhist,ntip
edhist,etip
edout,glistat
save
solve
fini

/post1
set,last
/dscale,,10       ! 10 times displacement magnification
plnsol,epxo,x,2   ! X and Y strains are the same ...
Chapter 16: Arbitrary Lagrangian-Eulerian Formulation

The Arbitrary Lagrangian-Eulerian (ALE) element formulation is a standard numerical approach for solving large strain deformation problems encountered in metal forming and high-speed impact applications. The general concept of the ALE formulation is that an arbitrary referential domain is defined for the description of motion that is different from the material (Lagrangian) and spatial (Eulerian) domains.

In a pure Lagrangian system, the mesh deforms with the material being modeled so that there is no material flow between elements. The Lagrangian approach is well suited for moderately large strain problems where mesh distortion and element entanglement are not a significant problem. The advantage of the Lagrangian approach is that the free surface of the material is automatically captured by the mesh. The main disadvantage of the Lagrangian approach is that problems develop in physical situations that involve highly deformed surfaces. Consider the high speed impact problem of a metal bar shown in Figure 16.1: High Speed Impact of a Metal Bar (p. 145). As shown in Figure 16.2: Lagrangian Impact Solution (p. 146), the Lagrangian approach begins to break down along the impact surface between the metal and the rigid wall. The mesh in this region becomes highly distorted and will not yield accurate results. The best solution for increasing the accuracy of the Lagrangian approach in highly deformed regions is to use an adaptive meshing procedure. Adaptive procedures, however, are computationally expensive and are still not fully developed in three-dimensional problems. Other disadvantages of the Lagrangian approach are that only one material can be modeled in each element and that new (damaged) surfaces cannot be created.

Figure 16.1: High Speed Impact of a Metal Bar

![High Speed Impact of a Metal Bar](image-url)
In a Eulerian based formulation, the mesh is stationary and the material flows through the mesh. The Eulerian approach originated in the fluid dynamics field and is best suited for very large deformation flow problems, such as the channel flow depicted in Figure 16.3: Eulerian Channel Flow Solution (p. 146). In the Eulerian method, new (damaged) surfaces are automatically created. The greatest disadvantage of the Eulerian approach is that a fine mesh is required to capture the material response, making the method very computationally expensive. This is particularly true for problems that contain regions where the structural response is desired and the strains are relatively small.

The Arbitrary Lagrangian-Eulerian (ALE) approach is a very effective alternative for simulating large deformation problems. In its most basic sense, the ALE method defines that the mesh motion is independent of the motion of the material being analyzed. Although the mesh motion may be arbitrary, it typically deforms with the material in ‘near Lagrangian’ flow fields. The greatest advantage of the ALE method is that it allows smoothing of a distorted mesh without performing a complete remesh. This smoothing allows the free surface of the material to be followed automatically without encountering the distortional errors of the Lagrangian approach (see Figure 16.4: ALE Impact Solution (p. 147)).

The main difficulty of the ALE method is the path dependent behavior of the plastic flow being modeled. Due to the path dependence, the relative motion between the mesh and the material must be accounted for in the material constitutive equations. In addition, the ALE method does not allow new (damaged) surfaces to be created and is limited to geometries where the material flow is relatively predictable.
16.1. Performing an ALE Analysis

The ALE formulation is available in the ANSYS LS-DYNA program only for PLANE162 and SOLID164 elements. For both of these element types, KEYOPT(5) defines the element continuum treatment. The default KEYOPT(5) setting is zero, which defines the material continuum to be Lagrangian. Setting KEYOPT(5) = 1 defines the ALE formulation. The current implementation of ALE in ANSYS LS-DYNA does not allow mixing of materials within an element.

In general, the 2-D ALE formulation (PLANE162) is not as robust as the 3-D formulation. Therefore, it may be advantageous to model 2-D problems with SOLID164 elements and then apply nodal motion constraints in the depth direction.

As described earlier, the greatest advantage of the ALE method is that it allows smoothing of a distorted mesh without performing a complete remesh. The algorithms for moving the mesh relative to the material control the range of the problems that can be solved by an ALE formulation. In the ANSYS LS-DYNA program, smoothing may be applied via the EDALE command. Using this command, you must define at least one of four different weighting factor options:

- **AFAC**: Simple average smoothing weight factor. For this method, the coordinates of a node are the simple average of the coordinates of its surrounding nodes.
- **BFAC**: Volume-weighted smoothing weight factor. This method uses a volume-weighted average of the coordinates of the centroids of the elements surrounding a node.
- **DFAC**: Equipotential smoothing weight factor. Equipotential zoning is a method of making a structured mesh for finite difference or finite element calculations by using the solutions of Laplace equations as the mesh lines. The same method can be used to smooth selected points in an unstructured 3-D mesh provided that it is at least locally structured.
- **EFAC**: Equilibrium smoothing weight factor (available only for PLANE162). For this method, artificial springs are attached to each ALE element node. The springs are used to adjust the position of each node from the equilibrium solution. This approach can overcome possible calculation instabilities found in the other smoothing methods.

The EDALE command also allows you to define start and end times for ALE smoothing.

You can define two additional ALE options with the EDGCALE command:

- **ADV**: Number of cycles between advection
METH - Advection method (donor cell or Van Leer)

In general, it is not worthwhile to advect an element unless at least twenty percent of its volume will be transported, because the gain in the time step size will not offset the cost of the advection calculations. It is best to begin an ALE analysis with a Van Leer advection technique ($METH = 1$).

If you are working in the GUI, all ALE related options are accessed by picking Main Menu> Solution> Analysis Options> ALE Options.

Note

In order to activate the ALE formulation for a given set of elements, you must specify a smoothing weight factor ($EDALE$ command), and you must specify the cycles between advectons ($NADV$ argument on the $EDGCALE$ command). If you do not supply both of these inputs, the default Lagrangian formulation will be used for all elements, including those for which $KEYOPT(5) = 1$.

The remap step maps the solution from a distorted Lagrangian mesh onto the new mesh. The underlying assumptions of the remap step are 1) the topology of the mesh is fixed (that is, the element nodal connectivity remains unchanged), and 2) the mesh motion during a step is less than the characteristic lengths of the surrounding elements. The algorithms for performing the remap step are taken from the computational fluid dynamics community and are referred to as “advection” algorithms.

The donor cell algorithm is a first order Godunov method applied to the advection equation. Aside from its first order accuracy, it is a good advection algorithm; it is stable, monotonic and simple. The Van Leer algorithm is a higher order Godunov method that improves the estimates of the initial values of left and right states for the Riemann problem at the nodes. The donor cell algorithm assumes that the distribution of the initial value function is constant over an element. Van Leer replaces the piecewise constant distribution with a higher order interpolation function that is subject to an element level conservation constraint.
Chapter 17: Drop Test Module

The Drop Test Module (DTM) greatly simplifies the procedure for simulating a drop test. A drop test involves orienting an object with respect to an assumed gravitational field and allowing it to drop from some specified height under the influence of gravity onto a flat, rigid surface (the target). In a typical drop test, the object is dropped from rest and the target lies in a plane perpendicular to the direction of the acceleration due to gravity (g).

The DTM provides a streamlined GUI that guides you through the steps involved in a drop test analysis. It allows you to quickly orient your model relative to the target, specify solution controls, and perform the explicit dynamic analysis of the test. The analysis predicts deformations and stresses in the object over a user-specified time interval to characterize the implications of the impact of the object with the target.

The Drop Test Module is specially designed so that a user having little experience with explicit dynamics can easily perform a drop test analysis. Thus, you will find many advantages to using the Drop Test Module. For example, the module automatically creates the target. The DTM also offers options to specify initial translational and angular object velocities, contact surface friction coefficients, and a target orientation that is not perpendicular to the direction of g. For most drop tests, the analysis can be started at a time just prior to impact. This reduces computation time. When this option is used, the DTM automatically calculates the object’s translational velocity and applies it at the start of the analysis. For all drop test cases, drag forces acting on the object as it moves through the air are neglected.

The following Drop Test Module topics are available:

17.1. Typical Drop Test Procedure
17.2. Advanced DTM Features
17.3. Drop Test Set-up Dialog Box
17.4. Picking Nodes
17.5. Postprocessing - Animation
17.6. Postprocessing - Graph and List Time-History Variables

17.1. Typical Drop Test Procedure

The step-by-step procedure described in this section can be used as a guide for most types of drop test analyses. A typical test would involve dropping an object from some height in a gravitational field onto a flat, rigid surface (target), neglecting surface friction. The basic procedure outlined here assumes that the object has a zero initial velocity, and the object is being dropped onto a target that lies in a plane which is normal to the direction of the acceleration due to gravity. More advanced features, such as inclusion of surface friction effects, specification of a nonzero initial velocity (translational and/or angular), and modification of the target size, properties, and orientation, are discussed in Advanced DTM Features (p. 153) in this user’s guide.

Before you begin, it is important for you to understand how the screen coordinates are used within the Drop Test Module. In the DTM, the screen coordinates are always defined so that the screen Y axis (vertical) is in the direction opposite to the acceleration due to gravity (g). Thus, screen coordinates are used as a convenient method for keeping track of the relationship between the object to be dropped and the gravitational field. See Screen Coordinates Definition (p. 152) in this user’s guide for more details.
17.1.1. Basic Drop Test Analysis Procedure

17.1.1.1. STEP 1: Create or import the model

Before entering the Drop Test Module, you must build or import the model of the object to be dropped. All elements should be defined and all material properties, including damping, should be specified. Only elements that are compatible with LS-DYNA (that is, explicit dynamic elements) should be included in the model. Make sure that there are not materials defined with either density or modulus of elasticity equal to zero. Otherwise, you will not be able to set up the DTM. See Chapter 2, Elements (p. 7) in this user’s guide for more information.

To avoid time increment degeneration problems, follow these guidelines when building the model:

- Avoid triangular, tetrahedron, and prism elements
- Avoid small elements
- Avoid acute-angled elements
- Use as uniform a mesh as possible

We recommend that a certain degree of damping be applied to the model to reduce the oscillatory response. You can apply both alpha (mass weighted) and beta (stiffness weighted) damping using the EDDAMP command.

Note

After completing the model and before entering the DTM, it is good practice to save the existing database under a unique name (pick Utility Menu> File> Save As).

17.1.1.2. STEP 2: Set up the DTM

You set up your analysis in the DTM by choosing Main Menu> Drop Test> Set Up. The Drop Test Set-up dialog box appears that includes several tabbed pages. The most common drop test analysis parameters are on the Basic tab, while parameters you may occasionally need to modify are included on most of the remaining tabs. A description of each tab is presented below along with the tasks you can perform within the tab:

Basic tab -- Specify magnitude of acceleration due to gravity (g), object drop height, object orientation, analysis start time, run time after impact, and the number of times data is written to the results file and to the time history output file. See Basic Tab of the Drop Test Set-up Dialog Box (p. 157) in this user’s guide for more information.

Velocity tab -- Modify initial translational and/or angular velocity of the object. See Velocity Tab of the Drop Test Set-up Dialog Box (p. 159) in this user’s guide for more information.

Target tab -- Modify dimensions and material properties of the target, rotate the target, and define contact and friction properties for the target. See Target Tab of the Drop Test Set-up Dialog Box (p. 160) in this user’s guide for more information.

Status tab -- Display the values currently set for acceleration due to gravity, initial translational and angular velocity, and the estimated time at the end of solution. See Status Tab of the Drop Test Set-up Dialog Box (p. 162) in this user’s guide for more information.
While the Drop Test Set-up dialog box is displayed, no other ANSYS menus are available until you choose **OK** or **Cancel**.

The remainder of this section (STEP 3 through STEP 9) discusses the options you would use during a typical drop test analysis. Advanced options that are not commonly used are covered in *Advanced DTM Features* (p. 153) in this user’s guide.

**17.1.1.3. STEP 3: Define the magnitude of (g)**

You must specify the magnitude of (g) by choosing the **Basic** tab. Under the **Gravity** heading, you pick a (g) value corresponding to the test specific system of units (386.4, 32.2, etc.), or you can type your own value.

**17.1.1.4. STEP 4: Specify the drop height**

You must specify the drop height on the **Basic** tab. Under the **Drop Height** heading, you type the drop height, and specify a reference point. The drop height is measured along the screen Y axis from the center of the top face of the target to some height reference point within the model. Reference point options include the lowest object point, that is, the point on the object with the minimum screen Y coordinate value, or the object's center of gravity as calculated by ANSYS. The drop height that you input must be in length units that are consistent with the rest of the analysis (gravity for example).

**17.1.1.5. STEP 5: Orient the object**

In the DTM, the screen coordinates are automatically defined with the Y axis directed opposite to the direction of (g). You can orient the object with respect to the gravitational field under the **Set Orientation** heading on the **Basic** tab. There you have the option of orienting the object by using the DTM Rotate tool, or by picking nodes on the object that define a vector parallel to the screen Y direction.

**17.1.1.6. STEP 6: Specify solution controls**

You can access solution control options on the **Basic** tab. Under the **Solution time** heading, you can specify when the analysis starts (near impact time or at drop time), as well as the run duration time after impact. Under the **Number of Results Output** heading, you can specify the number of times results are written to the results file (*file*.RST) and to the time-history file (*file*.HIS).

For more information on the tasks involved in STEP 3 through STEP 6, see *Basic Tab of the Drop Test Set-up Dialog Box* (p. 157) in this user’s guide.

**17.1.1.7. STEP 7: Solve**

Before solving, you should make sure that you are satisfied with the information in the Drop Test Set-up dialog box. Perform any of the following verification steps:

1. Review the information under all of the tabs.
2. Click on the **Status** tab since this displays your most updated input data.
3. View the object and the target in the ANSYS Graphics window.

If you choose the **Status** tab, ANSYS attempts to automatically create the target (unless you have already created it by choosing the Target tab) and displays a statement in black print at the end of the **Status** tab page indicating that all the set-up data is acceptable and you can proceed to solve the analysis, or a statement in yellow print indicating that you should proceed with caution because the data produced error messages, or a statement displayed in red indicating that the set-up data is unacceptable.
If you receive either the second or third statement regarding error messages, check them in the ANSYS Output window. See *Status Tab of the Drop Test Set-up Dialog Box* (p. 162) in this user's guide for more information on using the Status tab. Make changes if necessary, then choose OK. At this time, all of the standard ANSYS menus become usable again. You then initiate the LS-DYNA solution by choosing Main Menu> Drop Test> Solve.

**Note**

A database that you save from the current version of ANSYS LS-DYNA DTM is not compatible with a previous version of ANSYS LS-DYNA DTM.

### 17.1.1.8. STEP 8: Animate results

Once the solution is completed, you can choose Main Menu> Drop Test> Animate Results to animate dynamic results (such as von Mises stresses) for the dropped object over the analysis time duration. For animating results, it is recommended that you choose to start the analysis at drop time. See *Postprocessing - Animation* (p. 163) in this user's guide for information on the Animate Over Results dialog box, and see "Animation" in the Basic Analysis Guide for general information on producing animations in ANSYS.

### 17.1.1.9. STEP 9: Obtain Time-History Results

In addition to animating the results, you can also review the analysis results at specific points in the model as a function of time by choosing Main Menu> Drop Test> Time History. See Chapter 12, Postprocessing (p. 115) and *Postprocessing - Graph and List Time-History Variables* (p. 164) in this user's guide for more information.

### 17.1.2. Screen Coordinates Definition

When you first enter the DTM, screen Cartesian coordinates, which are fixed, are defined automatically, and are represented graphically in the lower left corner of the ANSYS Graphics window. Screen coordinates are used as a convenient way of keeping track of the relationship between the object to be dropped and the gravitational field. There are three important points you should understand about screen coordinates:

- In the DTM, screen coordinates are always defined so that the screen Y axis is in the direction opposite to the acceleration due to gravity, (g).
- Screen coordinates are defined with respect to global Cartesian coordinates.
- Object coordinates can be rotated with respect to the screen coordinates.

The screen coordinates are always defined so that the Y axis is in the screen “up” direction. The object coordinate system is the coordinate system in which the finite element model was defined. You can reorient the object coordinate system using the DTM Rotate tool. You can define a vector to be parallel to the screen Y coordinate system by picking two nodes on the object. See *Basic Tab of the Drop Test Set-up Dialog Box* (p. 157) in this user's guide for more information. By redefining the object Y direction, you are specifying the relationship between the object and the gravitational field, because the screen Y direction is always assumed to be “up” or opposite to the direction of the acceleration due to gravity, (g).

The screen coordinates also determine the default drop test viewing orientation, which is defined as follows:

- The screen X direction is "right" (to the viewer's right).
- The screen Y direction is "up" on the screen.
- The screen Z axis is along the normal from the screen to the viewer, or "out".
17.1.3. Additional Notes on the Use of the DTM

This section contains some notes of interest and practices to follow that will help you to use the DTM more effectively. Some of these points were mentioned briefly in previous sections.

• During initialization, a keyword is created that is related to the filtering of menu picks in the DTM. The keyword is DTFILT. You should avoid using this word as a keyword or a parameter name, as it could interfere with the filtering of menu options.

• Within the DTM, a component containing all the nodes in the object is automatically created and given the name _ddtrpob. This component name is used in issuing commands associated with LS-DYNA. You should avoid using this name.

• Parameters _dtcgnum and _dtlownum are used for time-history postprocessing plots and lists. You should avoid using these names.

• The DTM assumes that there are no constraints imposed on the object being dropped. However, coupling of components within an object is allowed.

• Some parameter names used internally by the Drop Test Module begin with the underscore character (_), so you should avoid using parameter names that begin with an underscore. (This is true for any ANSYS analysis.)

• If you do not choose to orient the object, the default assumption is the current screen Y axis view, which is directed opposite to the acceleration due to gravity (g).

• You can modify any input set up parameter any number of times before entering the solution phase. All parameters in the Drop Test Module are subject to redefinition after being initially defined. You will see that whenever an input parameter that affects the initial target/object orientation is changed, the target is recreated automatically to reflect the change when you click on the Target tab or the Status tab.

• After you specify a drop height, you will notice that, by default, the target is located nearly in contact with the object. This is due to the default option to begin the analysis near impact time. The target is created with just a small separation between the object and target. You may want to adjust the Solution time option to begin the analysis “at drop time”. After doing so, you will observe that the redefined target is at a position consistent with the specified drop height. However, while this option provides full animation of the drop scenario, it will require more computation time.

• To use the option to start the analysis near impact time, the initial angular velocity must be zero.

• The Drop Test Module does not alter the model of the object in any way. The DTM does create a model of a target after you specify a new orientation for the object.

17.2. Advanced DTM Features

This section discusses advanced features of the Drop Test Module that are provided as a convenience, but may not be required in a typical drop test analysis. These features include options to specify initial translational and angular velocities, modify the target's position and properties, and specify contact surface friction coefficients. Details on these features are included in the following sections.

17.2.1. Object Initial Velocity

By default, the translational and angular velocities of the object being dropped are assumed to be zero at the time of the drop. You can impose an initial translational and/or angular velocity on the object being dropped; however, in certain cases when you specify initial angular velocities, as discussed below, the “near impact time” starting option is not valid and the GUI prevents you from making this choice. The “near impact
time” starting option is further discussed in Basic Tab of the Drop Test Set-up Dialog Box (p. 157), under Solution Time in this user’s guide.

You should always input initial velocities corresponding to the time of the "drop", even if the analysis is set to begin near impact. The program will calculate the initial translational velocity at the start of the analysis based on the initial velocity input and the acceleration due to gravity.

The initial velocities that you specify are applied via the EDVEL command. This command is discussed further under Initial Velocity (p. 42) in this user’s guide.

You can specify initial translational or angular velocity by choosing Main Menu> Drop Test> Set Up then choosing the Velocity tab. Under either the Translational Velocity or Angular Velocity headings, you can choose to input the velocity in terms of screen coordinates or object coordinates. After choosing the desired option, simply input the X, Y, and Z components of the velocity. If you enter the velocity relative to the screen coordinates, the components are resolved into the object coordinates internally by the DTM before being input on the EDVEL command. See Velocity Tab of the Drop Test Set-up Dialog Box (p. 159) in this user’s guide for more information.

17.2.2. Modifying the Target

The target is one rigid SOLID164 brick element, which is created automatically in the Drop Test Module and is always assumed to be stationary. Figure 17.1: Two Views of the Target (p. 154)(a) shows the target in the default drop test viewing orientation, and Figure 17.1: Two Views of the Target (p. 154)(b) is an alternate view of the target that more clearly shows its geometry. Figure 17.1: Two Views of the Target (p. 154)(b) shows that the target has two faces that are much larger than the other four. The two large faces lie in planes that are parallel to each other. The large face that has an outward normal vector directed along the screen Y axis (by default) is the “top” face of the target.

**Figure 17.1: Two Views of the Target**

(a) Side View of Target in its Default Orientation:

(b) Alternate View of Target:
It is expected that the default target position, size, orientation, material properties, and contact parameters will be acceptable for most analyses. But if necessary, you can change the characteristics of the target by following the procedures outlined in the following sections.

17.2.2.1. Target Position

You can change the target position by redefining the reference for the target center.

To change the target position, choose **Main Menu> Drop Test> Set Up** then choose the **Target** tab. Under the **Dimensions** heading, there are two options from which you can choose for centering the target in the screen X-Z plane. You can align the target's center of gravity either underneath the object's center of gravity, or underneath the object's lowest Y coordinate point. See **Target Tab of the Drop Test Set-up Dialog Box (p. 160)** in this user's guide for more information.

17.2.2.2. Target Size

The default target size is based on the characteristic target length \( T_L \). The program calculates \( T_L \) based on the overall dimensions of the object being dropped. By default, the sides of the two large square faces of the target are of length \( 2T_L \), and the thickness of the target in the direction normal to these two faces is \( 0.1T_L \).

You can change the size of the target by choosing **Main Menu> Drop Test> Set Up** then choosing the **Target** tab. Under the **Dimensions** heading, you can enter two target scaling factors, one for scaling the target length \( L_s \) and one for scaling the target thickness \( T_s \). Thus, the sides of the two large square faces are of length \( 2L_sT_L \), and the target height is \( 0.1T_sT_L \). As an example, if you input \( L_s=2.0 \) and \( T_s=0.5 \), then the target is recreated with sides that are twice the default length and a thickness that is half its default value. Using this option to change the target size does not alter the location of the target center point. If you change the target's size, the target is re-drawn automatically. See **Target Tab of the Drop Test Set-up Dialog Box (p. 160)** in this user's guide for more information.

17.2.2.3. Target Orientation

By default, the target is oriented so that its two large faces are normal to the screen Y axis, and therefore normal to \((g)\). But you may need to change the target orientation for your particular model. For example, you may want to simulate the object being dropped onto an inclined plane. To modify the target orientation, choose **Main Menu> Drop Test> Set Up**, choose the **Target** tab. Under the **Orientation angle** heading, you type an angle in degrees that is used to rotate the target about the screen Z axis (which is normal to the screen and toward the viewer) using a right-hand rule. This is ideal for creating an inclined plane target. If the target is not in its default orientation when this option is chosen, it is initially returned to its default orientation so that it is always modified beginning from the default. If you change the target's orientation, the target is re-drawn automatically. See **Target Tab of the Drop Test Set-up Dialog Box (p. 160)** in this user's guide for more information.

17.2.2.4. Target Material Properties

Although the target is always defined to be rigid, a material density, modulus of elasticity, and Poisson's ratio are still required for the LS-DYNA solution. Material properties are used for a contact stiffness calculation and a time step calculation. Details on the contact stiffness and time step can be found under **Chapter 6, Contact Surfaces (p. 55)** in this user's guide.

By default, the target is defined to be rigid using the **EDMP** command:

```
EDMP, RIGID, MAT, 7, 7
```
The material number (MAT) used is one greater than the highest material number used by the model of the object being dropped. The default property values used for the target depend on the material properties used for the object. The Drop Test Module searches through all materials belonging to the object, finds the one with the lowest elastic modulus, and stores the corresponding material number. The modulus, density, and Poisson's ratio corresponding to this material number are also specified as the modulus, density, and Poisson's ratio of the target material.

If you know the actual target material properties, it is preferred that you input them directly. To specify the properties, choose Main Menu> Drop Test> Set Up, choose the Target tab, then, under the Material Properties heading, enter the Young's modulus, density, and Poisson's ratio. See Target Tab of the Drop Test Set-up Dialog Box (p. 160) in this user's guide for more information.

17.2.2.5. Specifying Friction Coefficients

In the Drop Test Module, automatic single surface contact (AG) is defined for the entire model. The contact is specified via the EDCGEN command.

You can change the static friction coefficient (FS), the dynamic friction coefficient (FD), and the exponential decay coefficient (DC). These coefficients apply to all contact surfaces, including the target surface. You can input values for these coefficients by choosing Main Menu> Drop Test> Set Up, then choosing the Target tab.

If you do not change parameters FS, FD, or DC, then they have default values of 0.0. See Target Tab of the Drop Test Set-up Dialog Box (p. 160) in this user's guide for more information.

17.3. Drop Test Set-up Dialog Box

17.3.1. Using the Drop Test Set-up Dialog Box

You use the Drop Test Set-up dialog box for inputting parameters for an ANSYS LS-DYNA analysis that uses the Drop Test Module (DTM). You access it by choosing Main Menu> Drop Test> Set Up.

The Drop Test Set-up dialog box consists of four tabbed “pages,” each of which contains a group of related DTM set-up information. The tabs, in order from left to right, are Basic, Velocity, Target, and Status. Each information group is logically arranged on a tab; the most basic and frequently addressed parameters appear on the first tab, with each of the next two tabs providing more advanced parameters. Most DTM analyses involve using the default settings for parameters on these two tabs. You do however have the option of modifying these parameters as needed. The Status tab displays the values currently set for acceleration due to gravity, initial translational and angular velocity, and the estimated time at the end of solution. This tab is located furthest to the right because you are encouraged to examine its contents as a final set up task, before committing the parameters to the ANSYS database (by clicking OK in the dialog box).

Once you are satisfied with the settings on the Basic tab, you do not need to progress through the remaining tabs unless you want to change some of the advanced parameters. As soon as you click OK on any tab of the dialog box, the settings are applied to the ANSYS database and the dialog box closes. The target is then created and loads are applied.

---

**Note**

When you make changes to any tabbed page, your changes are applied to the ANSYS database only when you click OK to close the dialog box.
17.3.2. Basic Tab of the Drop Test Set-up Dialog Box

Figure 17.2: Drop Test Set-up Dialog Box - Basic Tab

Use the Basic tab to specify the options described below.

**Gravity** -- Specify the magnitude of gravity.

**Magnitude of g** drop down list and text field: You have the option to select one of the standard values for gravity or type in a customized value. The values for gravity you can choose from the drop down list are:

- zero (default)
- 386.4
- 32.2
- 9.81
- 9810

Any value that you choose must have units that are consistent with the units used in the model of the object being dropped. You only need to input the magnitude of gravity. The direction is assumed to be in the negative Y direction, referenced to the screen coordinates. The object's coordinate components of the gravity loading are imposed on the object internally by the DTM, using three separate EDLOAD commands just before the problem is solved.

**Drop Height** -- Specify values for the drop height and the height reference point.

- **Height** field: (default = 0) You type the value of the drop height in units consistent with the units of length used in creating the object being dropped. The height is always measured along the Y axis (referenced to the screen coordinates) from the center of the top face of the target to some height reference point in the object.
- **Reference** drop down list: You choose the reference point from the following options:
  - **Lowest Object Point** (default): Reference point is the point on the object with the minimum screen Y coordinate value. If this option is used, then the node on the object with the minimum Y coordinate value is the reference point. If there is more than one node at this Y coordinate, then one of them is chosen arbitrarily to be the reference point.
- **Object CG**: Reference point is the object’s center of gravity as calculated by ANSYS. If the center of gravity has been previously calculated for some other purpose, then the information is stored, and the calculation is not repeated.

If the analysis is set to begin near impact time (default), then you will notice that the separation between the center of the target and the reference point, measured along the screen Y axis, is not equal to the specified drop height. This is because the target is created so that there is only a small separation, which is the case just before impact. However, if the analysis is set to begin at drop time, the separation between the target and the reference point equals the drop height.

By default, the target is oriented such that its top face is normal to the screen Y axis. If the orientation of the target is modified from the default orientation (see *Target Orientation* (p. 155) in this User’s Guide), the location of the center of the top face of the target at the time of the drop is not altered. Therefore, the definition of the drop height outlined above is not changed.

---

**Note**

You should always define the drop height so that all nodes on the object are *above* the plane of the top face of the target prior to the drop.

---

**Set Orientation** -- Specify the orientation of the model with respect to the gravitational field.

- **Rotate** button: You click this button to access the DTM Rotate dialog box, which allows you to dynamically rotate the object on the screen. This dialog box contains a *Help* button that you can click for more information.
- **Pick Nodes** button: You click this button to access a picking dialog box, which allows you to define a vector either by picking two nodes, or by picking one node and having the object’s center of gravity be the second node. The “up” direction is defined as the direction going from the first node that you pick to the second node that you pick. This dialog box contains a *Help* button that you can click for more information.

**Solution Time** -- Specify when the drop test analysis will start and how long it will run.

- **Start analysis near impact time** radio button (default): You click this option to save computation time that would be expended as the object drops through the air. To use this option, the angular velocity must be zero. The velocity at the start of the analysis is calculated automatically assuming the object is a rigid body and there is no resistance due to the air. If an initial translational velocity is applied, it is taken into account in calculating the velocity at the start of the analysis.
- **Start analysis at drop time** radio button: You click this option to include the entire time it takes for the object to drop from the specified height in the analysis. Although extra computation time is expended that is not directly involved with the drop test impact, this option accounts for rigid body rotation and is useful if you want to produce animations that include frames of the entire drop test, starting from when the object drops.
- **Run time after impact** field: You type the time in seconds that you want the analysis to run after the object hits the target.

**Number of Results Output** -- Specify the number of times that the output is written to the results file and to the time-history file.

- **On Results file** field: (default = 100) You type the number of times that the output is written to results file *Jobname*.RST. See the EDRST command for details.
- **On time history file** field: (default = 1000) You type the number of times that the output is written to time-history file *Jobname*.HIS. See the EDHTIME command for details. Note here that a component...
containing the lowest point and the node closest to the object’s center of gravity is automatically created for time-history postprocessing. You may define additional nodes. Refer to the EDHIST command for details.

**Action Buttons**

- **OK**: Applies the settings on the Drop Test Set-up dialog box to the ANSYS database and closes the dialog box.
- **Cancel**: Cancels any changes that you have made on the Drop Test Set-up dialog box since the last time that you clicked the **OK** button.
- **Help**: Displays this help topic.

**Related Information** -- For further information about the options on the **Basic** tab, see the following sections in this user’s guide:

- **Typical Drop Test Procedure** (p. 149)
- **Modifying the Target** (p. 154)
- **Picking Nodes** (p. 163)
- **Object Initial Velocity** (p. 153)
- **Chapter 12, Postprocessing** (p. 115)
- **Using the Drop Test Set-up Dialog Box** (p. 156)

**17.3.3. Velocity Tab of the Drop Test Set-up Dialog Box**

**Figure 17.3: Drop Test Set-up Dialog Box - Velocity Tab**

Use the **Velocity** tab to specify the object velocities, as described below.

**Translational Velocity** -- Specify the initial translational velocity of the object being dropped at the beginning of the drop time, even if the analysis is set to begin near impact. If you apply a translational velocity other than zero, and choose to start the analysis near impact time, the initial angular velocity must be zero. If these conditions are not met, ANSYS will automatically start the analysis at drop time, and it will automatically recreate the target.

- **Velocity relative to** drop down list: You choose either object coordinate system or screen coordinate system (default). If you enter the velocity relative to the screen coordinate system, the components are resolved into the object coordinate system internally by the DTM.
- **X Component** field: (default = 0) The X component of the translational velocity.
• **Y Component** field: (default = 0) The Y component of the translational velocity.
• **Z Component** field: (default = 0) The Z component of the translational velocity.

**Angular Velocity** -- Specify the initial angular velocity of the object being dropped. The angular velocity is about the object centroid in units of radians/(unit time). Applying an angular velocity other than zero is only valid when starting the analysis at drop time. ANSYS automatically sets this option when you choose Solve, and it automatically recreates the target to reflect this change.

• **Velocity relative to** drop down list: You choose either object coordinate system (default) or screen coordinate system. If you enter the velocity relative to the screen coordinate system, the components are resolved into the object coordinate system internally by the DTM before being input on the **EDVEL** command.
• **X Component** field (default = 0): The X component of the angular velocity.
• **Y Component** field (default = 0): The Y component of the angular velocity.
• **Z Component** field (default = 0): The Z component of the angular velocity.

**Action Buttons**

• **OK**: Applies the settings on the Drop Test Set-up dialog box to the ANSYS database and closes the dialog box.
• **Cancel**: Cancels any changes that you have made on the Drop Test Set-up dialog box since the last time that you clicked the **OK** button.
• **Help**: Displays this help topic.

**Related Information** -- For further information about the options on the **Velocity** tab, see the following sections in this user's guide:

• **Object Initial Velocity** (p. 153)
• **Using the Drop Test Set-up Dialog Box** (p. 156)

### 17.3.4. Target Tab of the Drop Test Set-up Dialog Box

**Figure 17.4: Drop Test Set-up Dialog Box - Target Tab**

When you choose the **Target** tab, the target is automatically created.

Also, use the **Target** tab to specify the options described below.

**Dimensions** -- Specify the target's position and size.
• **Center target using** drop down list. You can change the position of the target by changing the location of the point on the center of the top face of the target in the screen X-Z plane. The point's Y coordinate is fixed when the drop height is specified. You can designate this point with one of the following options in the drop down list:
  - **Lowest Object Point**: (default) The lowest object point is the point on the object with the minimum screen Y coordinate value. If only a single node exists on the object at this point, then the screen X coordinate value of this node is used as the point's X coordinate, and the screen Z value is used as the point's Z value. If there is more than one node at the minimum Y point, then the X coordinate of the point is found by first selecting only those nodes located at the minimum Y point. Then, the point's X value is equal to the sum of the maximum and minimum screen X values in the selected set of nodes, divided by 2. Similarly, the point's Z value is equal to the sum of the maximum and minimum screen Z values in the selected set of nodes, divided by 2.
  - **Object CG**: The screen X value at the object center of gravity is used as the point's X value, and the screen Z value of the cg is used as the point's Z value.

• **Length scale factor** field: (default = 1.0) You type a scale factor that changes the length of the target from the default length. The default length is determined by ANSYS, based on the overall dimensions of the object being dropped.

• **Thickness scale factor** field: (default = 1.0) You type a scale factor that changes the thickness of the target from the default thickness. The default thickness is equal to 0.1 times the default length.

**Orientation angle** -- Specify the rotation of the target about the screen Z axis (normal to the screen and pointing toward you).

**Rotation about screen Z** field: (default = 0) You type the rotation angle in degrees about the screen Z axis.

The drop height loses its meaning when the normal to the target's large faces is perpendicular to the screen Y axis. Therefore, if you modify the orientation so that the angle is greater than 89° or less than -89° with the screen Y axis, then the input is considered to be invalid. A warning is printed to the screen, and the target is returned to its previous orientation.

**Material Properties** -- Specify the material properties of the target. Material properties are used for a contact stiffness calculation and a time step calculation.

• **Young's Modulus** field: (default = object's lowest Young's modulus) You type the target's Young's modulus in units consistent with the analysis.

• **Density** field: (default = object's highest density) You type the target's density in units consistent with the analysis.

• **Poisson's Ratio** field: (default = object's Poisson's ratio) You type the target's Poisson's ratio.

**Contact** -- Specify the contact friction coefficients described below.

• **Static friction coeff** field: (default = 0.0) You type the static friction coefficient, which is $FS$ as described in the Notes section of the **EDCGEN** command.

• **Dynamic friction coeff** field: (default = 0.0) You type the dynamic friction coefficient, which is $FD$ as described in the Notes section of the **EDCGEN** command.

• **Exponential decay coeff** field: (default = 0.0) You type the exponential decay coefficient, which is $DC$ as described in the Notes section of the **EDCGEN** command.

**Action Buttons**

• **OK**: Applies the settings on the Drop Test Set-up dialog box to the ANSYS database and closes the dialog box.

• **Cancel**: Cancels any changes that you have made on the Drop Test Set-up dialog box since the last time that you clicked the **OK** button.
17.3.5. Status Tab of the Drop Test Set-up Dialog Box

Use the **Status** tab to check the values currently set for acceleration due to gravity, initial translational and rotational velocity, and the total time at the end of solution. One of the following statements is also displayed:

- **Successful set-up. Ready to drop.** [black print] This statement indicates that all the current set-up data is acceptable and you can click **OK** to commit the settings.

- **Proceed with caution. See error messages.** [yellow print] This statement indicates that some of the current set-up data could be problematic in enabling a solution, or is otherwise “non-standard” (for example, a zero entry detected for the value of gravity). You should examine the error messages in the ANSYS Output window to determine what is causing them to occur. Then, if necessary, change settings as appropriate and recheck the **Status** tab until the first message in black appears for problem situations. Click **OK** to commit the settings.

- **Unsuccessful set-up. See error messages.** [red print] This statement indicates that the current set-up data has produced errors that are serious enough to prevent a solution. The target will not be created in this situation. You should examine the error messages in the ANSYS Output window to determine the causes of these problems. Then change settings as appropriate and recheck the **Status** tab until the first message in black appears. Click **OK** to commit the settings.

It is recommended that you check this information in the **Status** tab along with a view of the object and the target in the ANSYS Graphics window before you click **OK** in the Drop Test Set-up dialog box as this commits all the set up data to the ANSYS database.

**Action Buttons**

- **OK**: Applies the settings on the Drop Test Set-up dialog box to the ANSYS database and closes the dialog box.
17.4. Picking Nodes

The Pick Nodes dialog box appears after you have clicked on the Pick Nodes... button on the Basic tab of the Drop Test Set-up dialog box. You use the Pick Nodes dialog box in specifying the orientation of the model with respect to the gravitational field by picking two nodes to define a vector in the screen y direction, or by picking one node and having the object’s center of gravity (cg) be the second node. The “up” direction is defined as the direction going from the first node that you picked to the second node that you pick.

- **Pick Two Nodes** option: To specify the model’s orientation by picking two nodes:
  1. Pick the first node.
  2. Choose OK in the Pick Nodes dialog box.
  3. Pick the second node.

  The object’s orientation changes according to the vector defined by the two nodes that you picked. The object is aligned such that the vector you created will be parallel to the Y direction of the screen coordinate system.

- **Pick One Node and Use CG** option: To specify the model’s orientation by picking one node and using the object’s center of gravity (calculated by ANSYS):
  1. Pick the first node.
  2. Choose OK in the Pick Nodes dialog box.
  3. Choose OK again in the Pick Nodes dialog box because the second node is the object’s center of gravity by default.

  The object’s orientation changes according to the vector defined by the node that you picked and the object’s center of gravity.

Related Information -- For further information related to the Pick Nodes dialog box, see the following sections:

- Basic Tab of the Drop Test Set-up Dialog Box (p. 157) in this user’s guide
- "Graphical Picking" in the Operations Guide

17.5. Postprocessing - Animation

You access the Animate Over Results dialog box after solution by choosing Main Menu> Drop Test> Animate Results.

The Animate Over Results dialog box is based on the ANSYS ANDATA command and allows you to specify animation options as described below.
Model result data radio buttons -- Specify the type of results data to be used for the animation sequence. Options are:

- Current Load Step (default).
- Load Step Range.
- Result Set Range.

Range Minimum, Maximum text fields -- Specify the first and last data sets over which the animation should span. If the Minimum field is left blank or equals 0, the default is the first data set. If the Maximum field is left blank or equals 0, the default is the last data set.

Increment result set text field -- Specify the increment between result data set (default = 1).

Include last SBST (substep) for each LDST (load step) check box -- Specify whether or not to force the last result data set in a load step to be animated. The default is unchecked, meaning that the last data set is not included in the animation.

Animation time delay (sec) text field -- Specify the time delay between frames during the animation (default = 0.5 seconds).

Display Type two-column selection list -- Specify what parameter is to be animated in the left column (for example, Stress, Stress-total, Stress-plastic), and the specific subset of the chosen parameter in the right column (for example, 1st Principal Stress, von Mises Stress).

Action Buttons

- OK: Applies the settings on the Animate Over Results dialog box to the ANSYS database and closes the dialog box.
- Cancel: Cancels any changes that you have made on the Animate Over Results dialog box since the last time that you clicked the OK button.
- Help: Displays information on the ANDATA command from the Command Reference.

After you choose OK, the animation will be created and will be displayed. An Animation Controller dialog box will also be displayed that includes controls for starting and stopping the animation, as well as changing its direction. This dialog box includes its own Help button where more detailed information is available.

Related Information -- For further information related to the Animate Over Results dialog box, see the following sections:

- STEP 8: Animate results (p. 152) in this user's guide.
- "Animation" in the Basic Analysis Guide.

17.6. Postprocessing - Graph and List Time-History Variables

You access the Graph Time-History Variables dialog box or the List Time-History Variables dialog box after solution by choosing Main Menu> Drop Test> Time History.
Figure 17.6: Graph and Time-History Variables Dialog Box

Note

The contents of the Graph Time-History Variables dialog box and the List Time-History Variables dialog box are the same so this help topic describes both of them. Only the Graph Time-History Variables dialog box is shown above.

Use the applicable dialog box to display the time-history results of a drop test analysis as described below:

Nodes to graph/list drop down list: You select one of the following choices as the node(s) to graph or list:

- **CG and Lowest Pt.**: (default) Both the center of gravity and the lowest point of the object, as initially specified will be graphed or listed.
- **Center of grav.**: Only the center of gravity of the object will be graphed or listed.
- **Initial Low Pt.**: Only the lowest point of the object, as initially specified will be graphed or listed.

Item and Comp to be graphed/listed two-column selection list: You first select Displacement (default), Velocity, or Acceleration as the item to be graphed or listed, then select the x, y, or z component of the chosen item referenced to the object or screen coordinate system. The default item and component is displacement in the Y direction referenced to the object coordinate system.

Action Buttons

- **OK**: Applies the settings on the Graph/List Time-History Variables dialog box to the ANSYS database and closes the dialog box.
- **Cancel**: Cancels any changes that you have made on the Graph/List Time-History Variables dialog box since the last time that you clicked the OK button.
- **Help**: Displays this help topic.

Related Information -- For further information related to the Graph/List Time-History Variables dialog box, see the following sections in this user's guide:

- *Chapter 12, Postprocessing* (p. 115)
- *Screen Coordinates Definition* (p. 152)
Appendix A. Comparison of Implicit and Explicit Methods

This appendix addresses some fundamental differences in the implementation of the implicit solution method versus the explicit solution method. These topics are available:

A.1. Time Integration
A.2. Stability Limit
A.3. Critical Time Step Size of a Rod
A.4. ANSYS LS-DYNA Time Step Size

A.1. Time Integration

A.1.1. Implicit Time Integration

For implicit time integration, ANSYS solves the transient dynamic equilibrium equation using the Newmark approximation (or an improved method known as HHT). For more information, see Transient Analysis in the Theory Reference for the Mechanical APDL and Mechanical Applications.

For linear problems, the implicit time integration is unconditionally stable for certain integration parameters. The time step size will vary to satisfy accuracy requirements.

For nonlinear problems:

- The solution is obtained using a series of linear approximations (Newton-Raphson method), so each time step may have many equilibrium iterations.
- The solution requires inversion of the nonlinear dynamic equivalent stiffness matrix.
- Small, iterative time steps may be required to achieve convergence.
- Convergence tools are provided, but convergence is not guaranteed for highly nonlinear problems.

A.1.2. Explicit Time Integration

For the explicit method, a central difference time integration method is used. Accelerations evaluated at time \( t \) are given by:

\[
\{a_t\} = [M]^{-1} (\{F_{t}^{\text{ext}}\} - \{F_{t}^{\text{int}}\})
\]

where

- \( \{F_{t}^{\text{ext}}\} \) is the applied external and body force vector
- \( \{F_{t}^{\text{int}}\} \) is the internal force vector which is given by:

\[
F^{\text{int}} = \sum_{\Omega} \left( \int_{\Omega} \mathbf{B}^T \sigma \, d\Omega + \mathbf{F}_{\text{hg}} + \mathbf{F}_{\text{contact}} \right)
\]
$F_{\text{hg}}$ is the hourglass resistance force, and $F_{\text{cont}}$ is the contact force.

The velocities and displacements are then evaluated:

$$\{v_{t+\Delta t/2}\} = \{v_{t-\Delta t/2}\} + \{a_t\} \Delta t_t$$

$$\{u_{t+\Delta t}\} = \{u_t\} + \{v_{t+\Delta t/2}\} \Delta t_t + \Delta t / 2$$

where

$$\Delta t_t + \Delta t / 2 = .5(\Delta t_t + \Delta t_t + \Delta t)$$

and

$$\Delta t_t - \Delta t / 2 = .5(\Delta t_t - \Delta t_t + \Delta t)$$

The geometry is updated by adding the displacement increments to the initial geometry ($X_0$):

$$\{x_{t+\Delta t}\} = \{x_0\} + \{u_{t+\Delta t}\}$$

For nonlinear problems:

- A lumped mass matrix is required for simple inversion.
- The equations become uncoupled and can be solved for directly (explicitly).
- No inversion of the stiffness matrix is required. All nonlinearities (including contact) are included in the internal force vector.
- The major computational expense is in calculating the internal forces.
- No convergence checks are needed since the equations are uncoupled.
- Very small time steps are required to maintain the stability limit.

### A.2. Stability Limit

#### A.2.1. Implicit Method

For linear problems, the implicit solution is always stable; that is, the time step can be arbitrarily large.

For nonlinear problems, the time step size may become small due to convergence difficulties.

#### A.2.2. Explicit Method

The explicit solution is only stable if the time step size is smaller than the critical time step size:

$$\Delta t \leq \Delta t_{\text{crit}} = \frac{2}{\omega_{\text{max}}}$$

where $\omega_{\text{max}}$ = largest natural circular frequency.
Due to this very small time step size, the explicit method is useful only for very short transients.

**A.3. Critical Time Step Size of a Rod**

The critical time step is based on the Courant-Friedrichs-Levy criterion. As an example, we will compute the critical time step size of a rod.

The natural frequency is given by:

\[ \omega_{\text{max}} = 2 \frac{c}{l} \]

where

\[ c = \sqrt{\frac{E}{\rho}} \]

(c is the wave propagation velocity).

Substituting \( \omega_{\text{max}} \) into the equation for critical time step yields:

\[ \Delta t = \frac{l}{c} \]

\( \Delta t \) is the time needed for the wave to propagate through the rod of length \( l \). Note that the critical time step size for explicit time integration depends on element length and material properties (sonic speed).

**A.4. ANSYS LS-DYNA Time Step Size**

ANSYS LS-DYNA checks all elements when calculating the required time step. For stability reasons, a scale factor of 0.9 (default) is used to decrease the time step:

\[ \Delta t = 0.9 \frac{l}{c} \]

The characteristic length, \( l \), and the wave propagation velocity, \( c \), are dependent on element type.

For beam elements:

\( l = \text{length of the element, and} \)

\[ c = \sqrt{\frac{E}{\rho}} \]

For shell elements:

\( l = \frac{A}{\max(l_1, l_2, l_3, l_4)} \)
For triangular shell elements:

\[ I = \frac{2A}{\max(I_1, I_2, I_3)} \]

For all shell elements:

\[ c = \sqrt{\frac{E}{\rho (1 - \nu^2)}} \]
Appendix B. Material Model Examples

This appendix provides example input for most of the material models available in ANSYS LS-DYNA.

**Note**

The input values given in this appendix are provided as examples only. The validity of these example materials is not guaranteed by ANSYS, Inc. It is the user’s responsibility to ensure that material data used in an ANSYS LS-DYNA analysis are accurate and appropriate for the structure being modeled.

The following material model topics are available:

B.1. ANSYS LS-DYNA Material Models

B.2. Material Model Examples

**B.1. ANSYS LS-DYNA Material Models**

The following table lists ANSYS LS-DYNA material models and their corresponding LS-DYNA commands. The LS-DYNA material model number is also listed. The “Example” column indicates whether example material input is provided for each model. The example input listings are shown in the section that follows this table.

<table>
<thead>
<tr>
<th>ANSYS Material Model</th>
<th>LS-DYNA Command</th>
<th>LS-DYNA MAT #</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isotropic Elastic</td>
<td>*MAT_ELASTIC</td>
<td>1</td>
<td>Yes</td>
</tr>
<tr>
<td>Orthotropic Elastic</td>
<td>*MATORTHOTROPIC_ELASTIC</td>
<td>2</td>
<td>Yes</td>
</tr>
<tr>
<td>Anisotropic Elastic</td>
<td>*MAT_ANISOTROPIC_ELASTIC</td>
<td>2</td>
<td>Yes</td>
</tr>
<tr>
<td>Elastic Fluid</td>
<td>*MAT_ELASTIC_FLUID</td>
<td>1</td>
<td>No</td>
</tr>
<tr>
<td>Blatz-Ko Rubber</td>
<td>*MAT_BLATZ-KO_RUBBER</td>
<td>7</td>
<td>Yes</td>
</tr>
<tr>
<td>Mooney-Rivlin Rubber</td>
<td>*MAT_MOONEY-RIVLIN_RUBBER</td>
<td>27</td>
<td>Yes</td>
</tr>
<tr>
<td>Viscoelastic</td>
<td>*MAT_VISCOELASTIC</td>
<td>6</td>
<td>Yes</td>
</tr>
<tr>
<td>Bilinear Isotropic Plasticity</td>
<td>*MAT_PLASTIC_KINEMATIC</td>
<td>3</td>
<td>Yes</td>
</tr>
<tr>
<td>Temperature Dependent Bilinear Isotropic Plasticity</td>
<td>*MAT_ELASTIC_PLASTIC_THERMAL</td>
<td>4</td>
<td>No</td>
</tr>
<tr>
<td>Transversely Anisotropic Elastic Plastic</td>
<td>*MAT_TRANSVERSELY_ANISOTROPIC_ELASTIC_PLASTIC</td>
<td>37</td>
<td>Yes</td>
</tr>
</tbody>
</table>
### Appendix B. Material Model Examples

<table>
<thead>
<tr>
<th>ANSYS Material Model</th>
<th>LS-DYNA Command</th>
<th>LS-DYNA MAT #</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transversely Anisotropic FLD</td>
<td>*MAT_FLD_TRANSVERSELY_ANISOTROPIC</td>
<td>39</td>
<td>Yes</td>
</tr>
<tr>
<td>Bilinear Kinematic</td>
<td>*MAT_PLASTIC_KINEMATIC</td>
<td>3</td>
<td>Yes</td>
</tr>
<tr>
<td>Plastic Kinematic</td>
<td>*MAT_PLASTIC_KINEMATIC</td>
<td>3</td>
<td>Yes</td>
</tr>
<tr>
<td>3 Parameter Barlat Plasticity</td>
<td>*MAT_3-PARAMETER_BARLAT</td>
<td>36</td>
<td>Yes</td>
</tr>
<tr>
<td>Barlat Anisotropic Plasticity</td>
<td>*MAT_BARLAT_ANISOTROPIC_PLASTICITY</td>
<td>33</td>
<td>Yes</td>
</tr>
<tr>
<td>Rate Sensitive Powerlaw Plasticity</td>
<td>*MAT_RATE_SENSITIVE_POWERLAW_PLASTICITY</td>
<td>64</td>
<td>Yes</td>
</tr>
<tr>
<td>Strain Rate Dependent Plasticity</td>
<td>*MAT_STRAIN_RATE_DEPENDENT_PLASTICITY</td>
<td>19</td>
<td>Yes</td>
</tr>
<tr>
<td>Piecewise Linear Plasticity</td>
<td>*MAT_PIECEWISE_LINEAR_PLASTICITY</td>
<td>24</td>
<td>Yes</td>
</tr>
<tr>
<td>Modified Piecewise Linear Plasticity</td>
<td>*MAT_MODIFIED_PIECEWISE_LINEAR_PLASTICITY</td>
<td>123</td>
<td>Yes</td>
</tr>
<tr>
<td>Composite Damage</td>
<td>*MAT_COMPOSITE_DAMAGE</td>
<td>22</td>
<td>No</td>
</tr>
<tr>
<td>Concrete Damage</td>
<td>*MAT_CONCRETE_DAMAGE</td>
<td>72</td>
<td>No</td>
</tr>
<tr>
<td>Powerlaw Plasticity</td>
<td>*MAT_POWER_LAW_PLASTICITY</td>
<td>18</td>
<td>Yes</td>
</tr>
<tr>
<td>Elastic Viscoplastic Thermal</td>
<td>*MAT_elastic_viscoelastic _THERMAL</td>
<td>106</td>
<td>Yes</td>
</tr>
<tr>
<td>Elastic-Plastic Hydrodynamic</td>
<td>*MAT_ELASTIC_PLASTIC_HYDRO</td>
<td>10</td>
<td>No</td>
</tr>
<tr>
<td>Geological Cap</td>
<td>*MAT_GEOLOGICAL_CAP_MODEL</td>
<td>25</td>
<td>Yes</td>
</tr>
<tr>
<td>Closed Cell Foam</td>
<td>*MAT_CLOSED_CELL_FOAM</td>
<td>53</td>
<td>No</td>
</tr>
<tr>
<td>Viscous Foam</td>
<td>*MAT_VISCIOUS_FOAM</td>
<td>62</td>
<td>No</td>
</tr>
<tr>
<td>Low Density Foam</td>
<td>*MAT_LOW_DENSITY_FOAM</td>
<td>57</td>
<td>No</td>
</tr>
<tr>
<td>Crushable Foam</td>
<td>*MAT_CRUSHABLE_FOAM</td>
<td>63</td>
<td>No</td>
</tr>
<tr>
<td>Honeycomb Foam</td>
<td>*MAT_HONEYCOMB</td>
<td>26</td>
<td>No</td>
</tr>
<tr>
<td>Tabulated EOS</td>
<td>*EOS_TABULATED</td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>Bamman</td>
<td>*MAT_BAMMAN</td>
<td>51</td>
<td>No</td>
</tr>
<tr>
<td>ANSYS Material Model</td>
<td>LS-DYNA Command</td>
<td>LS-DYNA MAT #</td>
<td>Example</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>--------------------------------------</td>
<td>---------------</td>
<td>---------</td>
</tr>
<tr>
<td>Johnson-Cook</td>
<td>*MAT_JOHNSON_COOK</td>
<td>15</td>
<td>Yes</td>
</tr>
<tr>
<td>Linear Polynomial EOS</td>
<td>*EOS_LINEAR_POLYNOMIAL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Johnson-Cook</td>
<td>*MAT_JOHNSON_COOK</td>
<td>15</td>
<td>Yes</td>
</tr>
<tr>
<td>Gruneisen EOS</td>
<td>*EOS_GRUNEISEN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Null Linear Polynomial EOS</td>
<td>*MAT_NULL</td>
<td>9</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>*EOS_LINEAR_POLYNOMIAL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Null Gruneisen EOS</td>
<td>*MAT_NULL</td>
<td>9</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>*EOS_GRUNEISEN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zerilli-Armstrong</td>
<td>*MAT_ZERILLI_ARMSTRONG</td>
<td>65</td>
<td>No</td>
</tr>
<tr>
<td>Steinberg</td>
<td>*MAT_STEINBERG</td>
<td>11</td>
<td>Yes</td>
</tr>
<tr>
<td>Gruneisen EOS</td>
<td>*EOS_GRUNEISEN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linear Elastic Spring</td>
<td>*MAT_SPRING_ELASTIC</td>
<td>N/A</td>
<td>No</td>
</tr>
<tr>
<td>General Nonlinear Spring</td>
<td>*MAT_SPRING_GENERAL_NONLINEAR</td>
<td>N/A</td>
<td>No</td>
</tr>
<tr>
<td>Nonlinear Elastic Spring</td>
<td>*MAT_SPRING_NONLINEAR_ELASTIC</td>
<td>N/A</td>
<td>No</td>
</tr>
<tr>
<td>Elastoplastic Spring</td>
<td>*MAT_SPRING_ELASTOPLASTIC</td>
<td>N/A</td>
<td>No</td>
</tr>
<tr>
<td>Inelastic Tension or Compression-only Spring</td>
<td>*MAT_SPRING_INELASTIC</td>
<td>N/A</td>
<td>No</td>
</tr>
<tr>
<td>Maxwell Viscosity Spring</td>
<td>*MAT_SPRING_MAXWELL</td>
<td>N/A</td>
<td>No</td>
</tr>
<tr>
<td>Linear Viscosity Damper</td>
<td>*MAT_DAMPER_VISCOUS</td>
<td>N/A</td>
<td>No</td>
</tr>
<tr>
<td>Nonlinear Viscosity Damper</td>
<td>*MAT_DAMPER_NONLINEAR_VISCOUS</td>
<td>N/A</td>
<td>No</td>
</tr>
<tr>
<td>Cable</td>
<td>*MAT_CABLE_DISCRETE_BEAM</td>
<td>71</td>
<td>Yes</td>
</tr>
<tr>
<td>Rigid</td>
<td>*MAT_RIGID</td>
<td>20</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**B.2. Material Model Examples**

**B.2.1. Isotropic Elastic Example: High Carbon Steel**

MP,ex,1,210e9 ! Pa
MP,nuxy,1,.29 ! No units
MP,dens,1,7850 ! kg/m³
Appendix B. Material Model Examples

### B.2.2. Orthotropic Elastic Example: Aluminum Oxide

\[
\begin{align*}
\text{MP}_{ex}, 1,307e9 & \quad \text{(Pa)} \\
\text{MP}_{ey}, 1,358.1e9 & \quad \text{(Pa)} \\
\text{MP}_{ez}, 1,358.1e9 & \quad \text{(Pa)} \\
\text{MP}_{gxy}, 126.9e9 & \quad \text{(Pa)} \\
\text{MP}_{gxz}, 126.9e9 & \quad \text{(Pa)} \\
\text{MP}_{gyz}, 126.9e9 & \quad \text{(Pa)} \\
\text{MP}_{nuxy}, 1.20 & \quad \text{(No units)} \\
\text{MP}_{nuxz}, 1.20 & \quad \text{(No units)} \\
\text{MP}_{nuyz}, 1.20 & \quad \text{(No units)} \\
\text{MP}_{dens}, 1,3750 & \quad \text{(kg/m}^3\text{)}
\end{align*}
\]

### B.2.3. Anisotropic Elastic Example: Cadmium

\[
\begin{align*}
\text{MP}_{dens}, 1,3400 & \quad \text{(kg/m}^3\text{)} \\
\text{TB}, \text{ANEL}, 1 \\
\text{TBDATA}, 1,121.0e9 & \quad \text{C11 (Pa)} \\
\text{TBDATA}, 2,48.1e9 & \quad \text{C12 (Pa)} \\
\text{TBDATA}, 3,121.0e9 & \quad \text{C22 (Pa)} \\
\text{TBDATA}, 4,44.2e9 & \quad \text{C13 (Pa)} \\
\text{TBDATA}, 5,44.2e9 & \quad \text{C23 (Pa)} \\
\text{TBDATA}, 6,51.3e9 & \quad \text{C33 (Pa)} \\
\text{TBDATA}, 10,18.5 & \quad \text{C44 (Pa)} \\
\text{TBDATA}, 15,18.5 & \quad \text{C55 (Pa)} \\
\text{TBDATA}, 21,24.2 & \quad \text{C66 (Pa)}
\end{align*}
\]

### B.2.4. Blatz-Ko Example: Rubber

\[
\begin{align*}
\text{MP}_{dens}, 1,1150 & \quad \text{(kg/m}^3\text{)} \\
\text{MP}_{gxy}, 1,104e7 & \quad \text{(Pa)}
\end{align*}
\]

### B.2.5. Mooney-Rivlin Example: Rubber

\[
\begin{align*}
\text{MP}_{dens}, 1,0018 & \quad \text{(lb/in}^3\text{)} \\
\text{MP}_{nuxy}, 1,499 & \quad \text{(No units)} \\
\text{TB}, \text{MOONEY}, 1 \\
\text{TBDATA}, 1,180 & \quad \text{C}_{10} \quad \text{(psi)} \\
\text{TBDATA}, 2,20 & \quad \text{C}_{01} \quad \text{(psi)}
\end{align*}
\]

### B.2.6. Viscoelastic Example: Glass

\[
\begin{align*}
\text{MP}_{dens}, 1,2390 & \quad \text{(kg/m}^3\text{)}
\end{align*}
\]
TB, EVISC, 1
TB DATA, 46, 27.4e9  ! G₀ (Pa)
TB DATA, 47, 0.0  ! G∞ (Pa)
TB DATA, 48, 60.5e9  ! Bulk modulus (Pa)
TB DATA, 61, 53  ! 1/β

B.2.7. Bilinear Isotropic Plasticity Example: Nickel Alloy

MP, ex, 1, 180e9  ! Pa
MP, nuxy, 1, .31  ! No units
MP, dens, 1, 8490  ! kg/m³
TB, BISO, 1
TB DATA, 1, 1900e6  ! Yield stress (Pa)
TB DATA, 2, 445e6  ! Tangent modulus (Pa)

B.2.8. Transversely Anisotropic Elastic Plastic Example: 1010 Steel

MP, ex, 1, 207e9  ! Pa
MP, nuxy, 1, .29  ! No units
MP, dens, 1, 7845  ! kg/m³
TB, PLAW, 7
TB DATA, 1, 128.5e6  ! Yield stress (Pa)
TB DATA, 2, 202e5  ! Initial strain at failure
TB DATA, 3, 1.41  ! r-value
TB DATA, 4, 1  ! Yield stress vs. plastic strain curve (see EDCURVE below)

*DIM, STRAIN,, 5
*DIM, YLDSTRES,, 5

Strain(1) = 0, .05, 1, .15, .2

YldStres(1) = 207e6, 210e6, 214e6, 218e6, 220e6  ! yield stress

EDCURVE, ADD, 1, Strain (1), YldStres(1)

B.2.9. Transversely Anisotropic FLD Example: Stainless Steel

MP, ex, 1, 30e6  ! Pa
MP, nuxy, 1, .29  ! No units
MP, dens, 1, 00285  ! kg/m³
TB, PLAW, 10
TB DATA, 1, 20e3  ! Initial yield stress (Pa)
TB DATA, 2, 5000  ! Tangent modulus (Pa)
Appendix B. Material Model Examples

TBDATA, 3, 2 ! Hardening parameter
TBDATA, 5, 1 ! Maximum yield stress curve (see EDCURVE below)
*DIM, mnstrn,, 6
*DIM, mjstrn,, 6

mnstrn(1) = -30, -10, 0, 20, 40, 50
mjstrn(1) = 80, 40, 29, 39, 45, 44

EDCURVE, ADD, 1, mnstrn(1), mjstrn(1)

B.2.10. Bilinear Kinematic Plasticity Example: Titanium Alloy

MP, ex, 1, 100e9 ! Pa
MP, nuxy, 1, .36 ! No units
MP, dens, 1, 4650 ! kg/m³
TB, BKIN, 1
TBDATA, 1, 70e6 ! Yield stress (Pa)
TBDATA, 2, 112e6 ! Tangent modulus (Pa)

B.2.11. Plastic Kinematic Example: 1018 Steel

MP, ex, 1, 200e9 ! Pa
MP, nuxy, 1, .27 ! No units
MP, dens, 1, 7865 ! kg/m³
TB, PLAW,,, 1
TBDATA, 1, 310e6 ! Yield stress (Pa)
TBDATA, 2, 763e6 ! Tangent modulus (Pa)
TBDATA, 4, 40.0 ! C (s⁻¹)
TBDATA, 5, 5.0 ! P
TBDATA, 6, .75 ! Failure strain

B.2.12. 3 Parameter Barlat Example: Aluminum 5182

MP, ex, 1, 76e9 ! Pa
MP, nuxy, 1, .34 ! No units
MP, dens, 1, 2720 ! kg/m³
TB, PLAW,,, 3
TBDATA, 1, 1 ! Hardening rule of 1 (yield stress)
TBDATA, 2, 25e6 ! Tangent modulus (Pa)
TBDATA, 3, 145e6 ! Yield stress (Pa)
TBDATA, 4, 0.170 ! Barlat exponent, \( m \)
TBDATA, 5, .73 ! \( R_{00} \)
TBDATA, 6, .68 ! \( R_{45} \)

MP\text{ex,1}, 1.76e9 \quad ! \text{Pa}
MP\text{nuxy,1}, .34 \quad ! \text{No units}
MP\text{dens,1}, 1.2720 \quad ! \text{kg/m}^3
TB, \text{PLAW},.,6
TB\text{DATA,1}, 1.04 \quad ! k (\text{MPa})
TB\text{DATA,2}, .65 \quad ! \text{Initial strain at failure}
TB\text{DATA,3}, .254 \quad ! n
TB\text{DATA,4}, 1.11 \quad ! \text{Barlat exponent, m}
TB\text{DATA,5}, 1.017 \quad ! a
TB\text{DATA,6}, 1.023 \quad ! b
TB\text{DATA,7}, .9761 \quad ! c
TB\text{DATA,8}, .9861 \quad ! f
TB\text{DATA,9}, .8875 \quad ! h


MP\text{ex,1}, 1.75e9 \quad ! \text{Pa}
MP\text{nuxy,1}, .33 \quad ! \text{No units}
MP\text{dens,1}, 1.2750 \quad ! \text{kg/m}^3
TB, \text{PLAW},.,4
TB\text{DATA,1}, 1.002 \quad ! k (\text{MPa})
TB\text{DATA,2}, .7 \quad ! m
TB\text{DATA,3}, .32 \quad ! n
TB\text{DATA,4}, 5.0 \quad ! \text{Initial strain rate (s}^{-1}\text{)}

B.2.15. Strain Rate Dependent Plasticity Example: 4140 Steel

MP\text{ex,1}, 1.209e9 \quad ! \text{Pa}
MP\text{nuxy,1}, .29 \quad ! \text{No units}
MP\text{dens,1}, 1.7850 \quad ! \text{kg/m}^3
TB, \text{PLAW},.,5
TB\text{DATA,1}, 1.1 \quad ! \text{LCID yield stress vs. strain rate (see first EDCURVE command below)}
TB\text{DATA,2}, 2.22e5 \quad ! \text{Tangent modulus (Pa)}
TB\text{DATA,3}, 2 \quad ! \text{LCID Elastic modulus vs. strain rate (see second EDCURVE command below)}
*DIM,StrnRate,.,5
B.2.16. Piecewise Linear Plasticity Example: High Carbon Steel

MP,ex,1,207e9 ! Pa
MP,nuxy,1,.30 ! No units
MP,dens,1,7830 ! kg/m³
TB,PLAW,1,8
TBDATA,1,207e6 ! Yield stress (Pa)
TBDATA,3,.75 ! Failure strain
TBDATA,4,40.0 ! C (strain rate parameter)
TBDATA,5,5.0 ! P (strain rate parameter)
TBDATA,6,1 ! LCID for true stress vs. true strain (see EDCURVE below)

B.2.17. Modified Piecewise Linear Plasticity Example: PVC

MP,EX,1,443478 ! psi
MP,DENS,1,.03 ! lb/in³
MP,NUXY,1,.45 ! No units
TB,PLAW,1,11 ! Modified PLP
TBDATA,1,1000 ! Yield strength
TBDATA,2,50000 ! Tangent modulus
TBDATA,8,.25 ! Thinning strain at failure - 25%
TBDATA,11,1 ! Rate (1/s)
TBDATA,12,1 LCID3 (stress-strain curve for above Rate)
TBDATA,13,4 ! Rate (4/s)
! Define stress-strain curves for each rate
*DIM,strn1,,10
*DIM,strs1,,10
*DIM,strn4,,10
*DIM,strs4,,10
*DIM,strn45,,9
*DIM,strs45,,9

strn1(1)=0,.004,.006,.008,.01,.014,.018,.022,.026,.03
strn4(1)=0,.004,.006,.008,.01,.014,.018,.022,.026,.03
strn45(1)=0,.004,.006,.008,.01,.014,.018,.022
strs1(1)=0,375,520,675,745,900,1050,1200,1325,1400
strs4(1)=0,475,675,825,950,1200,1450,1675,1850,1925
strs45(1)=0,600,950,1175,1475,1775,2100,2400,2650

EDCURVE,ADD,1,strn1(1),strs1(1)
EDCURVE,ADD,2,strn4(1),strs4(1)
EDCURVE,ADD,3,strn45(1),strs45(1)

**B.2.18. Powerlaw Plasticity Example: Aluminum 1100**

MP,ex,1,69e9 ! Pa
MP,nuxy,1,.33 ! No units
MP,dens,1,2710 ! kg/m$^3$
TB,PLAW,,,2
TBDATA,1,0.598 ! k
TBDATA,2,0.216 ! n
TBDATA,3,6500.0 ! C (s$^{-1}$)
TBDATA,4,4.0 ! P

**B.2.19. Elastic Viscoplastic Thermal Example**

MP,dens,1,830 ! kg/m$^3$
MP,alpx,1,10e-6 ! m/m-C
MP,ex,1,3.06e9 ! Pa
TB,PLAW,1,,,12 ! Elastic viscoplastic thermal
TBDATA,1,3.447e7 ! Initial yield stress
TBDATA,12,1 ! LCID1 (effective stress vs. effective plastic strain)
TBDATA,13,2 ! LCID2 (elastic modulus vs. temp.)
TBDATA,14,3 ! LCID3 (Poisson's ratio vs. temp.)
Appendix B. Material Model Examples

TBDATA,15,4 ! LCID4 (yield stress vs. temp.)
TBDATA,18,5 ! LCID7 (coefficient of thermal expansion vs. temp.)

! Stress-strain curve
*DIM,strn1,,10
*DIM,strs1,,10
strn1(1)=0,.004,.006,.008,.01,.014,.018,.022,.026,.03
strs1(1)=0,1.224e7,1.836e7,2.455e7,3.06e7,3.54e7,3.75e7,3.90e7,4.05e7,4.12e7
EDCURVE,ADD,1,strn1(1),strs1(1)

! Temperature dependent curves
*DIM,temp1,,4
*DIM,nuxy1,,4
*DIM,ex1,,4
*DIM,ystrs1,,4
*DIM,alpha1,,4
temp1(1)=70,120,250,500
ex1(1)=3.06e9,1.85e9,1.12e9,6.79e8
nuxy1(1)=.445,.451,.454,.456
ystrs1(1)=3.447e7,1.85e7,1.042e7,5.627e6
alpha1(1)=1e-5,8.4e-6,7.6e-6,7.2e-6
EDCURVE,ADD,2,temp1(1),ex1(1)
EDCURVE,ADD,3,temp1(1),nuxy1(1)
EDCURVE,ADD,4,temp1(1),ystrs1(1)
EDCURVE,ADD,5,temp1(1),alpha1(1)

! Note: Be sure to define a temperature range that exceeds the actual resulting temperatures. In this particular case, LS-DYNA will expect all temperatures to be between 0 and 500 degrees C. This requirement is on the LS-DYNA side.

B.2.20. Geological Cap Example: SRI Dynamic Concrete

MP,gxy,1,1700 ! ksi
MP,dens,1,2.226e-7 ! kip·sec²/in⁴
TB,GCAP,1
TBDATA,1,2100 ! K (ksi)
TBDATA,2,.7 ! α (ksi)
TBDATA,3,.1 ! β
TBDATA,4,.2 ! γ (ksi)
TBDATA,5,1.473 ! β (ksi⁻¹)
TBDATA,6,10.8 ! R
TBDATA,7,.00154 ! D (ksi⁻¹)
B.2.21. Johnson-Cook Linear Polynomial EOS Example: 1006 Steel

\[\begin{align*}
&\text{MP},\text{ex},1,1.207e9 \quad \text{! Pa} \\
&\text{MP},\text{nuxy},1,.30 \quad \text{! No units} \\
&\text{MP},\text{dens},1,7850 \quad \text{! kg/m}^3 \\
&T\text{B},\text{EOS},1,,1,1 \\
&T\text{DATA},1,1350.25e6 \quad \text{! A (Pa)} \\
&T\text{DATA},2,275e6 \quad \text{! B (Pa)} \\
&T\text{DATA},3,.36 \quad \text{! n} \\
&T\text{DATA},4,.022 \quad \text{! c} \\
&T\text{DATA},5,1.0 \quad \text{! m} \\
&T\text{DATA},6,1400 \quad \text{! Melt temperature (°C)} \\
&T\text{DATA},7,30 \quad \text{! Room temperature (°C)} \\
&T\text{DATA},8,10 \quad \text{! Initial strain rate} \\
&T\text{DATA},9,4500 \quad \text{! Specific heat} \\
&T\text{DATA},10,240e6 \quad \text{! Failure stress} \\
&T\text{DATA},11,-.8 \quad \text{! Failure value D1} \\
&T\text{DATA},12,2.1 \quad \text{! Failure value D2} \\
&T\text{DATA},13,-.5 \quad \text{! Failure value D3} \\
&T\text{DATA},14,.0002 \quad \text{! Failure value D4} \\
&T\text{DATA},15,.61 \quad \text{! Failure value D5} \\
&T\text{DATA},17,140e9 \quad \text{! EOS linear polynomial term}
\end{align*}\]

B.2.22. Johnson-Cook Gruneisen EOS Example: OFHC Copper

\[\begin{align*}
&\text{MP},\text{ex},1,1.138e9 \quad \text{! Pa} \\
&\text{MP},\text{nuxy},1,.35 \quad \text{! No units} \\
&\text{MP},\text{dens},1,8330 \quad \text{! kg/m}^3 \\
&T\text{B},\text{EOS},1,,1,2 \\
&T\text{DATA},1,189.63e6 \quad \text{! A (Pa)} \\
&T\text{DATA},2,291.64e6 \quad \text{! B (Pa)} \\
&T\text{DATA},3,.31 \quad \text{! n} \\
&T\text{DATA},4,.025 \quad \text{! c} \\
&T\text{DATA},5,1.09 \quad \text{! m}
\end{align*}\]
B.2.23. Null Material Linear Polynomial EOS Example: Brass

\[ P_{\text{ex}}(1,200,000) \text{ Pa} \]
\[ \rho_{\text{x}}(1,7500) \text{ kg/m}^3 \]
\[ \rho(7500) \text{ kg/m}^3 \]

B.2.24. Null Material Gruneisen EOS Example: Aluminum

\[ P_{\text{ex}}(1,100,000) \text{ Pa} \]
\[ \rho_{\text{x}}(1,2500) \text{ kg/m}^3 \]
\[ \rho(2500) \text{ kg/m}^3 \]
B.2.25. Steinberg Gruneisen EOS Example: Stainless Steel

MP,ex,1,207e9 ! Pa
MP,nuxy,1,.3 ! No units
EDM,cable,1,1 ! See EDCURVE below
*DIM,EngStran,,4
*DIM,EngStres,,4

EngStran(1) = .02,.04,.06,.08
EngStres(1) = 207e6,210e6,215e6,220e6
EDCURVE,ADD,1,EngStran (1),EngStres(1)

B.2.27. Rigid Material Example: Steel

MP,ex,1,207e9 ! Pa
MP,nuxy,1,.3 ! No units
MP,dens,1,7580 ! kg/m³
Appendix B. Material Model Examples

EDMP,rigid,1,7,7
Appendix C. ANSYS LS-DYNA to LS-DYNA Command Mapping

When you initiate an ANSYS LS-DYNA solution via the **SOLVE** command (and also when you issue the **EDWRITE** command), an LS-DYNA input file called **Jobname.k** is created. This file contains all of the model information that you have entered into ANSYS. However, the information in this file is in terms of LS-DYNA keyword commands, not ANSYS commands.

The table below contains information that will help you interpret the **Jobname.k** file. The left column lists ANSYS commands that are unique to ANSYS LS-DYNA, and the right column lists the corresponding LS-DYNA commands. ANSYS commands that perform an action (such as **EDCLLIST**) have no corresponding LS-DYNA command and are not included in this table. ANSYS and LS-DYNA commands that are used to define material properties are listed in **Material Model Examples**.

<table>
<thead>
<tr>
<th>ANSYS Command</th>
<th>Corresponding LS-DYNA Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDADAPT</td>
<td>*PART</td>
</tr>
<tr>
<td>EDALE</td>
<td>*CONTROL_ALE</td>
</tr>
<tr>
<td>EDASMP</td>
<td>*SET_PART_LIST</td>
</tr>
<tr>
<td>EDBOUND</td>
<td>*BOUNDARY_SLIDING_PLANE *BOUNDARY_CYCLIC</td>
</tr>
<tr>
<td>EDBVIS</td>
<td>*CONTROL_BULK_VISCOSITY</td>
</tr>
<tr>
<td>EDBX</td>
<td>*DEFINE_BOX</td>
</tr>
<tr>
<td>EDCADAPT</td>
<td>*CONTROL_ADAPTIVE</td>
</tr>
<tr>
<td>EDCGEN</td>
<td>*CONTACT</td>
</tr>
<tr>
<td>EDCMORE</td>
<td>*CONTACT</td>
</tr>
<tr>
<td>EDCNSTR</td>
<td>*CONSTRAINED_EXTRA_NODES_SET</td>
</tr>
<tr>
<td></td>
<td>*CONSTRAINED_NODAL_RIGID_BODY</td>
</tr>
<tr>
<td></td>
<td>*CONSTRAINED_SHELL_TO_SOLID</td>
</tr>
<tr>
<td></td>
<td>*CONSTRAINED_RIVET</td>
</tr>
<tr>
<td>EDCONTACT</td>
<td>*CONTROL_CONTACT</td>
</tr>
<tr>
<td>EDCPU</td>
<td>*CONTROL_CPU</td>
</tr>
<tr>
<td>EDCRB</td>
<td>*CONSTRAINED_RIGID_BODIES</td>
</tr>
<tr>
<td>EDCSC</td>
<td>*CONTROL_SUBCYCLE</td>
</tr>
<tr>
<td>EDCTS</td>
<td>*CONTROL_TIMESTEP</td>
</tr>
<tr>
<td>EDCURVE</td>
<td>*DEFINE_CURVE</td>
</tr>
<tr>
<td>EDDAMP</td>
<td>*DAMPING_PART_MASS *DAMPING_PART_STIFFNESS</td>
</tr>
<tr>
<td>EDDC</td>
<td>*DELETE_CONTACT</td>
</tr>
<tr>
<td>EDDRELAX</td>
<td>*CONTROL_DYNAMIC_RELAXATION</td>
</tr>
<tr>
<td>EDDUMP</td>
<td>*DATABASE_BINARY_D3DUMP</td>
</tr>
<tr>
<td>ANSYS Command</td>
<td>Corresponding LS-DYNA Command</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------------------------------</td>
</tr>
<tr>
<td>EDENERGY</td>
<td>*CONTROL_ENERGY</td>
</tr>
<tr>
<td>EDGCALE</td>
<td>*CONTROL_ALE</td>
</tr>
<tr>
<td>EDHGLS</td>
<td>*CONTROL_HOURGLASS</td>
</tr>
<tr>
<td>EDHIST</td>
<td>*DATABASE_HISTORY_NODE *DATABASE_HISTORY</td>
</tr>
<tr>
<td>EDHTIME</td>
<td>*DATABASE</td>
</tr>
<tr>
<td></td>
<td>*DATABASE_BINARY_D3THDT</td>
</tr>
<tr>
<td>EDINT</td>
<td>*INTEGRATION_BEAM</td>
</tr>
<tr>
<td></td>
<td>*INTEGRATION_SHELL</td>
</tr>
<tr>
<td>EDIPART</td>
<td>*PART_INERTIA</td>
</tr>
<tr>
<td>EDIS</td>
<td>*STRESS_INITIALIZATION</td>
</tr>
<tr>
<td></td>
<td>*STRESS_INITIALIZATION_DISCRETE</td>
</tr>
<tr>
<td>EDLCS</td>
<td>*DEFINE_COORDINATE_VECTOR</td>
</tr>
<tr>
<td></td>
<td>*DEFINE_COORDINATE_SYSTEM</td>
</tr>
<tr>
<td></td>
<td>*DEFINE_VECTOR</td>
</tr>
<tr>
<td>EDLOAD, (FX, FY, FZ, MX, MY, MZ)</td>
<td>*LOAD_NODE_SET</td>
</tr>
<tr>
<td>EDLOAD, (UX, UY, UZ, ROTX, ROTY, ROTZ, VX, VY, VZ, AX, AY, AZ)</td>
<td>*BOUNDARY_PRESCRIBED_MOTION_SET</td>
</tr>
<tr>
<td>EDLOAD, (ACLX, ACLY, ACLZ, OMGX, OMGY, OMGZ)</td>
<td>*LOAD_BODY_GENERALIZED</td>
</tr>
<tr>
<td>EDLOAD, TEMP</td>
<td>*LOAD_THERMAL_VARIABLE</td>
</tr>
<tr>
<td>EDLOAD, PRESS</td>
<td>*LOAD_SEGMENT</td>
</tr>
<tr>
<td></td>
<td>*LOAD_SEGMENT_SET</td>
</tr>
<tr>
<td></td>
<td>*LOAD_SHELL_SET</td>
</tr>
<tr>
<td>EDLOAD, (RBUX, RBUY, RBUZ, +RBRX, RBRY, RBRZ, RBVX, RBVY, RBVZ, RBOX, RBOY, RBOZ)</td>
<td>*BOUNDARY_PRESCRIBED_MOTION_RIGID</td>
</tr>
<tr>
<td>EDLOAD, (RBFX, RBFY, RBFZ, RBMX, RBMY, RBMZ)</td>
<td>*LOAD_RIGID_BODY</td>
</tr>
<tr>
<td>EDMP, HGLS</td>
<td>*HOURGLASS</td>
</tr>
<tr>
<td>EDNB</td>
<td>*BOUNDARY_NON_REFLECTING</td>
</tr>
<tr>
<td>EDNROT</td>
<td>*BOUNDARY_SPC_SET</td>
</tr>
<tr>
<td>EDOPT</td>
<td>*DATABASE_FORMAT</td>
</tr>
<tr>
<td>ANSYS Command</td>
<td>Corresponding LS-DYNA Command</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>EDOUT</td>
<td>*DATABASE_OPTION</td>
</tr>
<tr>
<td>EDPART</td>
<td>*PART</td>
</tr>
<tr>
<td>EDPVEL</td>
<td>*SET_NODE</td>
</tr>
<tr>
<td></td>
<td>*INITIAL VELOCITY</td>
</tr>
<tr>
<td></td>
<td>*INITIAL VELOCITY_GENERATION</td>
</tr>
<tr>
<td></td>
<td>*CHANGE VELOCITY</td>
</tr>
<tr>
<td></td>
<td>*CHANGE VELOCITY_ZERO</td>
</tr>
<tr>
<td>EDRC</td>
<td>*RIGID DEFORMABLE_CONTROL</td>
</tr>
<tr>
<td>EDRD</td>
<td>*DEFORMABLE TO RIGID</td>
</tr>
<tr>
<td></td>
<td>*RIGID DEFORMABLE D2R</td>
</tr>
<tr>
<td></td>
<td>*RIGID DEFORMABLE R2D</td>
</tr>
<tr>
<td>EDRI</td>
<td>*DEFORMABLE TO RIGID_INERTIA</td>
</tr>
<tr>
<td>EDRST</td>
<td>*DATABASE_BINARY_D3PLOT</td>
</tr>
<tr>
<td>EDSHELL</td>
<td>*CONTROL SHELL</td>
</tr>
<tr>
<td>EDSP</td>
<td>*CHANGE SMALL PENETRATION</td>
</tr>
<tr>
<td>EDSTART</td>
<td>r = d3dumpnn lsdyna command line option</td>
</tr>
<tr>
<td>EDTERM</td>
<td>*TERMINATION_NODE</td>
</tr>
<tr>
<td></td>
<td>*TERMINATION BODY</td>
</tr>
<tr>
<td>EDVEL</td>
<td>*SET_NODE</td>
</tr>
<tr>
<td></td>
<td>*INITIAL VELOCITY</td>
</tr>
<tr>
<td></td>
<td>*INITIAL VELOCITY GENERATION</td>
</tr>
<tr>
<td></td>
<td>*CHANGE VELOCITY</td>
</tr>
<tr>
<td></td>
<td>*CHANGE VELOCITY_ZERO</td>
</tr>
<tr>
<td>EDWELD</td>
<td>*CONSTRAINED SPOTWELD</td>
</tr>
<tr>
<td></td>
<td>*CONSTRAINED GENERALIZED WELD_SPOT</td>
</tr>
<tr>
<td>BFUNIF or TUNIF</td>
<td>*LOAD THERMAL CONSTANT</td>
</tr>
<tr>
<td>LDREAD</td>
<td>*LOAD THERMAL CONSTANT_NODE</td>
</tr>
<tr>
<td>REXPORT</td>
<td>m = drelax lsdyna command line option</td>
</tr>
</tbody>
</table>
Appendix D. Thermal/Structural Preload Example

This appendix contains an example of a combined thermal/structural preload analysis followed by a structural transient. The example demonstrates the implicit-to-explicit procedure discussed in Chapter 15, Implicit-to-Explicit Sequential Solution (p. 135), with emphasis on how to handle temperature loads in this type of sequential solution (see Special Considerations for Thermal Loading (p. 139)).

The input file creates a cantilever beam structure modeled with thermal brick elements. The analysis is performed in four separate phases. In the first phase, a steady-state heat transfer analysis is conducted using ANSYS thermal elements. A thermal gradient is produced in which the left side of the beam is 20 degrees hotter than the right side.

In the second phase, the thermal elements are converted into ANSYS implicit structural elements. Next, the nonuniform temperatures from the end of the heat transfer analysis are read into the structural model with the **LDREAD** command. A pressure load is also applied to the top of the beam, and one end of the beam is restrained. A static thermal/structural preload analysis is then conducted.

The results of the second phase show that the temperature gradient causes the beam to bend to the side, and the pressure load causes the beam to bend downward. This is the desired preload condition, which should match the results at the beginning of the subsequent transient dynamic analysis.

In the third phase, the implicit elements are converted into explicit elements and a stress initialization to a prescribed geometry is conducted. This pseudo-dynamic relaxation analysis uses the displacements from the **drelax** file (but not the temperatures) to establish the preload. The temperatures are applied directly as static thermal loads in the **Jobname.K** file itself. The results of this analysis (at time = 0) match those of the ANSYS implicit analysis very well.

In phase four, a transient dynamic analysis is conducted. The pressure load is ramped up, while the temperature load is held constant. This is because only one type of thermal load is permitted in any given analysis.

The complete input file is shown below.

```
fini
/clear
/title, ANSYS LS-DYNA Thermal/Structural Preload Example
! units: kilogram, meter, second, Newton, Pascal, Joule, Watts
! - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
/filnam,thermal
/plopts,info,1
/psf,conv,hcoe,1
/view,,1,2,3
/prep7
et,1,SOLID70 ! thermal brick element
r,1 ! dummy real constant set
! Temperature-independent thermal material properties:
mp,dens,1,8.03e3 ! kg/m^3
```
Appendix D. Thermal/Structural Preload Example

```
mp, kxx, 1, 16.3 ! W/m-C
mp, c, 1, 502.0 ! J/kg-C

block, 0, 1, 0, 1, 0, 11
esize, 0.25
vmesh, all
cm, nbeam, node
cm, ebeam, elem
fini

/solu
antype, static ! steady state heat transfer analysis
time, 1.0

nsel, s, loc, x, 0 ! left side of beam
esln
bf, all, hgen, 3.0e3 ! 3,000 W/m^3 heat generation on left side
esel, all
nsel, s, ext ! exterior beam nodes
nsel, r, loc, x, 1.0 ! right side of beam
sf, all, conv, 30.0, 0.0 ! hf=30.0 W/m^2-C and Tbulk=0 degrees C
nsel, all

/title, ANSYS Heat Transfer Analysis
save
solve
fini

/title, ANSYS Heat Transfer Analysis

/post1
set, last
plnsol, temp
/wait, 3
save
fini

! = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =

/filnam, static

/psf, pres, norm, 2
/dscale, 1.0
/prep7
bfdele, all, all ! remove thermal body force (heat generation) loads
sfdele, all, all ! remove thermal surface (convection) loads

!!!etchg, tts ! convert SOLID70 elements into SOLID45 elements
!!!keyopt, 1, 2, 1 ! similar formulation to default SOLID164

et, 2, SOLID185 ! desired structural brick element
emodif, all, type, 2 ! ETCHG, TTS creates SOLID45 elements
etdele, 1 ! remove thermal bricks
numcmp, type ! compress element type numbers
keyopt, 1, 2, 1 ! uniform reduced integration to match SOLID164

! Temperature-dependent structural properties

mp, dens, 1, 8.03e3 ! kg/m^3 (can NOT be temp-dependent)
mptemp, 1, 0.0, 500.0 ! degrees C, two temperatures
mpdata, ex, 1, 1, 193.0e9, 93.0e9 ! N/m^2 (temperature-dependent)
mpdata, nxxy, 1, 1, 0.29, 0.28 ! unitless (temperature-dependent)
mpdata, alpx, 1, 1, 18.0e-6, 16.0e-6 ! m/m-C (temperature-dependent)

! Note: Do not define plasticity in this ANSYS implicit preload
! analysis, since implicit-to-explicit sequential solutions
! assume linear elastic material properties and small strain.
! Plasticity can be added in the subsequent transient run

mplist, all, all
fini
```
/solu
antype,static              ! thermal strain (preload) structural analysis
time,1.0
nsubst,2,1000,2
outres,all,last

ldread,temp,last,,,thermal,rth    ! temperature body loads (overwrites TUNIF)
tref,0.0          ! reference temperature for "instantaneous" alpha
bflist,all,temp
nsel,s,loc,y,1
cm,nbeamtop,node
esln
cm,ebeamtop,elem
sf,all,press,1.1e5    ! apply pressure load (N/m^2) to top of beam
nsel,s,loc,z,0
d,all,uz,0.0       ! restrain aft end of beam
nsel,s,loc,z,0
nsel,r,loc,y,0.5
d,all,wy,0
nsel,s,loc,z,0
nsel,r,loc,x,0.5
d,all,ux,0
nsel,all
esel,all
/title, ANSYS Implicit Static Preload Analysis
save
solve
fini

/post1
set,last
/dscale,,10          ! exaggerate displacement
plnsol,s,eqv,2
/wait,3

! Note: Make sure that the yield stress is not exceeded, since
! the preload needs to have a linear response for the
! subsequent dynamic relaxation analysis

save
fini

! = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =

/filnam,dynamic

/psf,pres,norm,2
/dscale,,1.0

/prep7
sfdele,all,all       ! remove structural surface (pressure) loads
etchg,ite            ! convert SOLID185 elements into SOLID164 elements

! Add plasticity to temperature-dependent properties
tb,biso,1,2            ! temperature-dependent BISO (2 temperatures)
tbtemp,0.0              ! first temperature
tbdata,1,66.7e6        ! N/m^2, yield stress at 0 degrees
            ! N/m^2, tangent modulus at 0 degrees
tbdata,2,1.93e9
            ! N/m^2, tangent modulus at 0 degrees
            ! second temperature
tbdata,1,60.0e6        ! N/m^2, yield stress at 500 degrees
tbdata,2,0.93e9        ! N/m^2, tangent modulus at 500 degrees
Appendix D. Thermal/Structural Preload Example

! Note: Be sure to define a temperature range that exceeds the
! actual resulting temperatures. In this particular case,
! LS-DYNA will expect all temperatures to be between 0 and
! 500 degrees C. This requirement is on the LS-DYNA side.

edmp,hgl,1,5           ! add hourglass control
edpart,create          ! create part list
eddamp,all,,0.10       ! mass (alpha) damping
eddamp,1,,1.0e-6       ! stiffness (beta) damping
mplist,all,all
blist,all,all
fini
/solu
time,0.2               ! termination time (seconds)
/rexport,dyna,,,,,static,rst  ! create drelax file

! Note: The temperatures written to the drelax file are NOT used by
! LS-DYNA to establish the thermal preload condition. Instead,
! the temperatures written via the *LOAD_THERMAL_CONSTANT_NODE
! command are used. The latter are stored in the ANSYS database
! as body force BF loads from the LDREAD command. These thermal
! loads remain active unless deleted (BFDELE,ALL,TEMP).

bflist,all,temp        ! no need to re-issue LDREAD command
eddrelax,ansys         ! stress initialization to a prescribed geometry

! Increase the pressure load after a bit to introduce plasticity in
! the beam. The beam should remain bent in the XZ plane (due to the
! thermal load) while bending further downward in the YZ plane.

*dim,etime,,5,1,1
*dim,epress,,5,1,1
etime(1)=0.0,0.025,0.050,0.100,0.201       ! extend curve out
epress(1)=1.1e5,1.1e5,2.2e5,2.2e5,4.4e5       ! force into plasticity
edload,add,press,4,ebeamtop,etime,epress    ! pressure (N/m^2) on beam top
dlist,all,all          ! keep beam restraints from previous implicit run

edrst,100
edhtime,200
edhist,rbeamtop
edhist,ebeamtop
edenergy,1,1,1,1
edout,glstat
edopt,add,,both
edwrite,both,,k

nsel,all
esel,all
/title, ANSYS LS-DYNA Dynamic Relaxation and Transient Analyses
save
solve
fini

/post1
set,first
/dscale,,10           ! exaggerate displacement
/title, ANSYS LS-DYNA Dynamic Relaxation Analysis
plnsol,s,eqv,2
/wait,3
set,last
/title, ANSYS LS-DYNA Transient Dynamic Analysis (Plastic State)
pinsol,s,eqv,2
/wait,3

save
fini
/exit,nosav
Bibliography


**Index**

**A**
Action buttons
- Drop Test Animate Over Results dialog box, 164
- Drop Test Graph Time-History dialog box, 165
- Drop Test List Time-History dialog box, 165
- Drop Test Set-up dialog box, 159
  - Status tab, 162
  - Target tab, 161
  - Velocity tab, 160
Adaptive meshing
  in explicit dynamic analysis, 29
Animating Results, Postprocessing, 117
ANSYS LS-DYNA overview, 1
Applying loads
  explicit dynamic analysis, 22
Array parameters, 35

**B**
Beam Elements, 15
BFUNIF command, 44

**C**
Commands used in explicit dynamic analyses, 2
Components, 34
Constraints, 40
  explicit dynamic analysis, 24
Constraints and initial conditions, 40
Contact Surface
  Definitions, 55
Contact Surfaces, 55
  Contact Options, 59
  Definition of Contact Types, 60
Coupling and constraint equations, 43
Creating Components, POST26, 115

**D**
Data curves, 38
Data Smoothing, 121
defining material models, 68
Discrete Elements, 16
Drop Test Graph Time-History Variables dialog box, 165
Drop Test List Time-History Variables dialog box, 165
Drop Test Pick Nodes dialog box, 163
Drop Test Set-up dialog box
  - Basic tab, 157
    - Drop Height, 157
    - Gravity, 157
    - Number of Results Output, 158
  - Set Orientation, 158
  - Solution Time, 158
general information, 156
Status tab, 162
Target tab, 160
  - Contact, 161
  - Dimensions, 160
  - Material Properties, 161
  - Orientation angle, 161
  - Velocity tab, 159
  - Angular Velocity, 160
  - Translational Velocity, 159
Dynamic relaxation, 45

**E**
EDTIME command, 25
EDLOAD command, 44
EDNDS command, 24
EDRST command, 25
EDSTART Command, 123
elastic material models, 67
Element Output Data, 117
Element Solution, Postprocessing, 120
Elmements
  - Beam and Link Elements, 15
  - BEAM161, 15
  - COMB165 Spring-Damper, 16
  - Discrete Elements, 16
  - explicit dynamics, 7
  - explicit solid and shell elements, 8
  - General Element Capabilities, 17
  - LINK160, 16
  - LINK167, 16
  - MASS166, 17
  - PLANE162, 13
  - SHELL163, 9
  - SOLID164, 8
  - SOLID168, 14
  - Triangular Shell Formulations, 10
equation of state material models, 68
Example problems, explicit dynamics, 5
Explicit dynamic analysis
  - adaptive meshing, 29
  - applying loads, 22
  - commands, 2
  - constraints, 24
data smoothing, 24
defining contact, 24
defining parts, 26
displaying load symbols, 23
DOF coupling, 24
element problems, 5
initial velocities, 23
modeling, 19
overview, 1
viewing load curves, 23
explicit dynamic analysis
output control options, 25
DOF coupling
drelax file, 135
dynamic relaxation, 135
explicit dynamic analysis, 24
explicit-implicit element type pairs, 129
implicit-explicit element type pairs, 135
importing stresses, 129
thermal-explicit element type pairs, 139
explicit-to-implicit sequential solution, 129

F
foam material models, 68

H
Hourglassing, 109

I
implicit-to-explicit sequential solution, 135
Initial velocity, 42
  explicit dynamic analysis, 23

L
LDREAD command, 44
Link Elements, 15
Load curves
  displaying in explicit dynamics, 23
Load symbols
  displaying in explicit dynamics, 23
Load types
  explicit dynamic analysis, 23
Loading, 33
  array parameters, 35
  components, 34
  constraints, 40
  constraints and initial conditions, 40
  coupling and constraint equations, 43
data curves, 38
dynamic relaxation, 45
general loading, 36
general loading options, 33
initial velocity, 42
nonreflecting boundary, 44
temperature, 44
Loads
  explicit dynamic analysis, 22
LS-DYNA input file, 52
LS-DYNA solver, 52

M
mass scaling, 111
material models, 67
  3-parameter Barlat, 77
  anisotropic elastic, 70
  Bamman plasticity model, 95
  Barlat anisotropic plasticity, 79
  bilinear isotropic, 74
  bilinear kinematic (strain rate independent), 76
  Blatz-Ko rubber, 72
cable, 102
closed cell foam, 89
composite damage, 83
crystal plasticity, 84
crushable foam, 91
defining, 68
descriptions, 69
elastic, 67, 70
elastic fluid, 71
elastic viscoplastic thermal, 85
elastic-plastic hydrodynamic, 86
elastoplastic spring, 101
equation of state, 68
foam, 68
general nonlinear spring, 101
Gruneisen equation of state, 93
honeycomb, 92
inertial tension- or compression-only spring, 101
Johnson-Cook plasticity, 95
linear elastic spring, 101
linear polynomial equation of state, 93
linear viscous damper, 102
low density foam, 91
maxwell viscoelastic spring, 102
modified piecewise linear plasticity, 82
Mooney-Rivlin rubber, 72
nonlinear elastic, 67
nonlinear elastic spring, 101
nonlinear inelastic, 67
nonlinear viscous damper, 102
null, 96
orthotropic elastic, 70
others, 68
piecewise linear plasticity, 81
plastic kinematic, 76
Power Law plasticity, 84
pressure dependent plasticity, 68
rate sensitive Power Law plasticity, 80
rigid, 103
spring-damper (discrete), 68
Steinberg model, 98
strain rate dependent plasticity, 80
tabulated equation of state, 94
temperature dependent bilinear isotropic, 74
transversely anisotropic FLD, 75
transversely anisotropic hardening, 75
viscoelastic, 73
viscous foam, 90
Zerilli-Armstrong model, 97

Modeling
in explicit dynamic analysis, 19
Monitoring an LS-DYNA solution, 50
MPP, 48

N
Nodal Solution, Postprocessing, 120
nonlinear elastic material models, 67
nonlinear inelastic material models, 67
Nonreflecting boundary, 44

O
other material models, 68
output control options
explicit dynamic analysis, 25
Output Controls, Postprocessing, 115
Output Data, 117
Output Files, POST26, 116

P
Parallel processing in ANSYS LS-DYNA, 48
Parameters, array, 35
Parts
in explicit dynamic analysis, 26
POST26, Creating Components, 115
POST26, Output Files, 116
POST26, use, 120
Postprocessing, 115
Animating Results, 117
Creating Components for POST26, 115
Data Smoothing, 121
Element Output Data, 117
Finding Additional Information, 121
in explicit dynamic analysis, 116
Nodal and Element Solutions, 120
Output Controls, 115
Reading ASCII Files for Miscellaneous Output Data, 121
Results vs. History, 115
Using POST26 with ANSYS LS-DYNA, 120
Writing the Output Files for POST26, 116

pressure dependent plasticity material models, 68

R
restarting, 123
EDSTART Command, 123
effect on Output Files, 127
Restart Dump File, 123
Rigid bodies, 105
inertia properties, 105
loading, 106
switching (D2R and R2D), 106

S
scaling, 111
Shell elements
explicit dynamics, 8
formulations, 9
Shell Formulations, Triangular, 10
SMP, 48
Solid elements
explicit dynamics, 8
Solution
editing the LS-DYNA input file, 52
solution control and monitoring, 50
Solution process, 47
Spring-Damper, 16
spring-damper (discrete) material models, 68
Subcycling, 113

T
Temperature loading, 44
Termination controls, 47
thermal-to-explicit sequential solution, 139
TIME command, 25
Triangular Shell Formulations, 10
TUNIF command, 44